

Alibaba Cloud Object Storage Service

SDK Reference

Issue: 20190917

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK.
Courier font	It is used for commands.	Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>switch {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Introduction.....	1
2 Java.....	2
2.1 Preface.....	2
2.2 Installation.....	3
2.3 Quick start.....	5
2.4 Initialization.....	9
2.5 Manage a bucket.....	15
2.6 Single-connection bandwidth throttling.....	21
2.7 Bucket tagging.....	23
2.8 Object tagging.....	26
2.8.1 Configure object tagging.....	26
2.8.2 Obtain the tags added to an object.....	37
2.8.3 Delete the tags added to an object.....	38
2.8.4 Object tagging and lifecycle rules.....	38
2.9 Server-side encryption.....	41
2.10 Upload objects.....	44
2.10.1 Overview.....	44
2.10.2 Simple upload.....	45
2.10.3 Form upload.....	48
2.10.4 Append upload.....	52
2.10.5 Resumable upload.....	53
2.10.6 Multipart upload.....	55
2.10.7 Upload progress bars.....	65
2.10.8 Upload callback.....	67
2.11 Download objects.....	68
2.11.1 Overview.....	68
2.11.2 Streaming download.....	68
2.11.3 Download to your local file.....	70
2.11.4 Range download.....	70
2.11.5 Resumable download.....	71
2.11.6 Conditional download.....	74
2.11.7 Download progress bars.....	76
2.12 Manage objects.....	77
2.12.1 Overview.....	77
2.12.2 Determine whether a specified object exists.....	78
2.12.3 Manage ACL for an object.....	78
2.12.4 Manage Object Meta.....	80
2.12.5 List objects.....	85

2.12.6 Delete objects.....	96
2.12.7 Copy objects.....	99
2.12.8 Restore an archive object.....	104
2.12.9 Manage a symbolic link.....	105
2.13 Data security.....	107
2.14 Authorized access.....	110
2.15 Manage lifecycle rules.....	115
2.16 CORS.....	119
2.17 Set logging.....	122
2.18 Static website hosting.....	124
2.19 Anti-leech.....	125
2.20 Cross-region replication.....	127
2.21 Image processing.....	131
2.22 Exception handling.....	140
2.23 FAQ.....	143
3 Python.....	154
3.1 Preface.....	154
3.2 Installation.....	155
3.3 Quick start.....	158
3.4 Initialization.....	160
3.5 Bucket.....	163
3.6 Single-connection bandwidth throttling.....	167
3.7 Bucket tagging.....	169
3.8 Object tagging.....	171
3.8.1 Configure object tagging.....	171
3.8.2 Obtain the tags added to an object.....	180
3.8.3 Delete the tags added to an object.....	180
3.8.4 Object tagging and lifecycle rules.....	181
3.9 Upload objects.....	184
3.9.1 Overview.....	184
3.9.2 Simple upload.....	184
3.9.3 Append upload.....	188
3.9.4 Resumable upload.....	188
3.9.5 Multipart upload.....	191
3.9.6 Upload progress bars.....	192
3.9.7 Upload callback.....	193
3.10 Download objects.....	194
3.10.1 Overview.....	194
3.10.2 Streaming download.....	194
3.10.3 Download to your local file.....	195
3.10.4 Range download.....	196
3.10.5 Resumable download.....	196
3.10.6 Download progress bars.....	199
3.11 Manage objects.....	200
3.11.1 Overview.....	200

3.11.2 Determine whether a specified object exists.....	200
3.11.3 Manage ACL for an object.....	201
3.11.4 Manage Object Meta.....	203
3.11.5 List objects.....	205
3.11.6 Delete objects.....	207
3.11.7 Copy objects.....	208
3.11.8 Restore an archive object.....	210
3.11.9 Manage a symbolic link.....	210
3.12 Authorized access.....	212
3.13 Enable logging in Python SDK.....	213
3.14 Data encryption.....	215
3.14.1 Client-side encryption.....	215
3.14.2 Server-side encryption.....	219
3.15 Static website hosting.....	222
3.16 Manage lifecycle rules.....	223
3.17 CORS.....	226
3.18 Set logging.....	228
3.19 Anti-leech.....	230
3.20 Image processing.....	232
3.21 Chinese character and time.....	241
3.22 Exception handling.....	243
4 Go.....	247
4.1 Preface.....	247
4.2 Installation.....	248
4.3 Quick start.....	249
4.4 Initialization.....	253
4.5 Bucket.....	256
4.6 Single-connection bandwidth throttling.....	263
4.7 Bucket tagging.....	267
4.8 Object tagging.....	270
4.8.1 Configure object tagging.....	270
4.8.2 Obtain the tags added to an object.....	279
4.8.3 Delete the tags added to an object.....	280
4.8.4 Object tagging and lifecycle rules.....	281
4.9 Server-side encryption.....	283
4.10 Upload objects.....	285
4.10.1 Overview.....	285
4.10.2 Simple upload.....	286
4.10.3 Append upload.....	289
4.10.4 Resumable upload.....	290
4.10.5 Multipart upload.....	292
4.10.6 Progress bars.....	298
4.11 Download objects.....	300
4.11.1 Overview.....	300
4.11.2 Streaming download.....	300

4.11.3 Range download.....	302
4.11.4 Download to your local file.....	303
4.11.5 Resumable download.....	304
4.11.6 Conditional download.....	306
4.11.7 Download a compressed object.....	308
4.11.8 Progress bars.....	309
4.12 Manage objects.....	310
4.12.1 Overview.....	310
4.12.2 Determine whether a specified object exists.....	310
4.12.3 Manage ACL for an object.....	311
4.12.4 Manage Object Meta.....	312
4.12.5 List objects.....	316
4.12.6 Delete objects.....	322
4.12.7 Copy objects.....	324
4.12.8 Restore an archive object.....	329
4.12.9 Manage a symbolic link.....	331
4.13 Authorized access.....	333
4.14 Manage lifecycle rules.....	336
4.15 Set logging.....	339
4.16 Static website hosting.....	341
4.17 Anti-leech.....	343
4.18 CORS.....	345
4.19 Image processing.....	347
4.20 Error handling.....	356
5 C++.....	361
5.1 Overview.....	361
5.2 Installation.....	362
5.3 Quick start.....	365
5.4 Initialization.....	369
5.5 Buckets.....	374
5.6 Single-connection bandwidth throttling.....	380
5.7 Object tagging.....	382
5.7.1 Configure object tagging.....	382
5.7.2 Obtain the tags added to an object.....	393
5.7.3 Delete the tags added to an object.....	394
5.7.4 Object tagging and lifecycle rules.....	396
5.8 Upload objects.....	398
5.8.1 Overview.....	398
5.8.2 Simple upload.....	399
5.8.3 Append upload.....	400
5.8.4 Resumable upload.....	402
5.8.5 Multipart upload.....	404
5.8.6 Interrupt an upload task.....	412
5.8.7 Use a progress bar.....	413
5.8.8 Upload callback.....	414

5.9 Download objects.....	415
5.9.1 Overview.....	416
5.9.2 Download to a local file.....	416
5.9.3 Download to local memory.....	417
5.9.4 Range download.....	418
5.9.5 Resumable download.....	419
5.9.6 Use a progress bar.....	421
5.10 Manage objects.....	422
5.10.1 Overview.....	422
5.10.2 Manage the ACL of an object.....	422
5.10.3 Manage Object Meta.....	425
5.10.4 List objects.....	427
5.10.5 Delete objects.....	434
5.10.6 Copy an object.....	435
5.10.7 Determine whether the specified object exists.....	439
5.10.8 Restore an archive object.....	440
5.10.9 Manage a symbolic link.....	442
5.11 Authorized access.....	444
5.12 Lifecycle rules.....	447
5.13 CORS.....	450
5.14 Access logging.....	453
5.15 Static website hosting.....	456
5.16 Anti-leech.....	458
5.17 IMG.....	461
6 PHP.....	469
6.1 Preface.....	469
6.2 Installation.....	470
6.3 Quick start.....	472
6.4 Initialization.....	476
6.5 Bucket.....	479
6.6 Upload objects.....	486
6.6.1 Overview.....	486
6.6.2 Simple upload.....	486
6.6.3 Append upload.....	488
6.6.4 Multipart upload.....	490
6.6.5 Upload callback.....	498
6.7 Download objects.....	501
6.7.1 Overview.....	501
6.7.2 Download to your local file.....	502
6.7.3 Download to your local memory.....	503
6.7.4 Range download.....	503
6.7.5 Conditional download.....	504
6.8 Manage objects.....	506
6.8.1 Overview.....	506
6.8.2 Determine whether a specified object exists.....	507

6.8.3 Manage ACL for an object.....	507
6.8.4 Manage Object Meta.....	510
6.8.5 List objects.....	512
6.8.6 Delete objects.....	515
6.8.7 Copy objects.....	517
6.8.8 Restore an archive object.....	519
6.8.9 Manage a symbolic link.....	520
6.8.10 Enable MD5 verification.....	521
6.9 Authorized access.....	522
6.10 Static website hosting.....	526
6.11 Manage lifecycle rules.....	528
6.12 Set logging.....	531
6.13 CORS.....	534
6.14 Anti-leech.....	536
6.15 Bind a custom domain name.....	539
6.16 Image processing.....	541
6.17 Exception handling.....	548
7 C.....	551
7.1 Preface.....	551
7.2 Installation.....	552
7.3 Quick start.....	556
7.4 Initialization.....	564
7.5 Bucket.....	568
7.6 Upload objects.....	578
7.6.1 Overview.....	579
7.6.2 Simple upload.....	579
7.6.3 Append upload.....	584
7.6.4 Resumable upload.....	588
7.6.5 Multipart upload.....	590
7.6.6 Upload progress bars.....	595
7.6.7 Upload callback.....	597
7.7 Download objects.....	599
7.7.1 Overview.....	599
7.7.2 Download an object to a local file.....	599
7.7.3 Download to your local memory.....	601
7.7.4 Range download.....	602
7.7.5 Resumable download.....	604
7.7.6 Download progress bars.....	606
7.8 Manage objects.....	608
7.8.1 Overview.....	608
7.8.2 Manage ACL for an object.....	608
7.8.3 Manage Object Meta.....	611
7.8.4 List objects.....	613
7.8.5 Delete objects.....	620
7.8.6 Copy objects.....	625

7.8.7 Restore an archive object.....	630
7.8.8 Manage a symbolic link.....	632
7.9 Authorized access.....	635
7.10 Manage lifecycle rules.....	640
7.11 Image processing.....	646
7.12 CORS.....	656
7.13 Set logging.....	661
7.14 Anti-leech.....	665
7.15 Static website hosting.....	670
7.16 Error handling.....	674
7.17 FAQ.....	676
8 . NET.....	677
8.1 Preface.....	677
8.2 Installation.....	678
8.3 Quick start.....	680
8.4 Initialization.....	683
8.5 Manage a bucket.....	687
8.6 Upload objects.....	691
8.6.1 Overview.....	691
8.6.2 Simple upload.....	691
8.6.3 Append upload.....	695
8.6.4 Resumable upload.....	696
8.6.5 Multipart upload.....	697
8.6.6 Upload progress bars.....	705
8.6.7 Upload callback.....	706
8.7 Download objects.....	708
8.7.1 Overview.....	708
8.7.2 Streaming download.....	708
8.7.3 Range download.....	709
8.7.4 Resumable download.....	711
8.7.5 Download progress bars.....	712
8.8 Manage objects.....	713
8.8.1 Overview.....	713
8.8.2 Determine whether a specified object exists.....	714
8.8.3 Manage ACL for an object.....	714
8.8.4 Manage Object Meta.....	716
8.8.5 List objects.....	718
8.8.6 Delete objects.....	723
8.8.7 Copy objects.....	725
8.8.8 Restore an archive object.....	728
8.8.9 Manage a symbolic link.....	730
8.9 Authorized access.....	731
8.10 Manage lifecycle rules.....	734
8.11 Set logging.....	737
8.12 CORS.....	739

8.13 Static website hosting.....	741
8.14 Anti-leech.....	743
8.15 Image processing.....	746
8.16 Exception handling.....	754
9 Android.....	757
9.1 Preface.....	757
9.2 Installation.....	757
9.3 Quick start.....	759
9.4 Initialization.....	763
9.5 Manage buckets.....	768
9.6 Upload objects.....	771
9.6.1 Overview.....	771
9.6.2 Simple upload.....	771
9.6.3 Append upload.....	774
9.6.4 Resumable upload.....	775
9.6.5 Progress bar.....	779
9.6.6 Upload callback.....	780
9.7 Download objects.....	781
9.7.1 Overview.....	781
9.7.2 Streaming download.....	781
9.7.3 Range download.....	783
9.7.4 Download an object to a local file.....	784
9.7.5 Conditional download.....	785
9.7.6 Resumable download.....	786
9.8 Manage objects.....	788
9.8.1 Overview.....	788
9.8.2 Determine whether a specified object exists.....	789
9.8.3 Copy an object.....	789
9.8.4 List objects.....	790
9.8.5 Delete an object.....	793
9.8.6 Set the Content-Type of an object.....	794
9.8.7 Obtain the metadata of an object.....	794
9.9 Data security.....	795
9.10 Sign a URL.....	797
9.11 Authorized access.....	797
9.12 Image processing.....	803
9.13 Exception handling.....	805
10 iOS.....	808
10.1 Preface.....	808
10.2 Installation.....	808
10.3 Quick start.....	810
10.4 Initialization.....	815
10.5 Manage a bucket.....	819
10.6 Upload objects.....	821

10.6.1 Overview.....	821
10.6.2 Simple upload.....	822
10.6.3 Multipart upload.....	824
10.6.4 Append upload.....	828
10.6.5 Resumable upload.....	829
10.6.6 Upload callback.....	831
10.7 Download objects.....	832
10.7.1 Overview.....	832
10.7.2 Simple download.....	832
10.7.3 Streaming download.....	833
10.7.4 Range download.....	834
10.7.5 Resumable download.....	835
10.8 Manage objects.....	844
10.8.1 Overview.....	844
10.8.2 Determine whether the specified object exists.....	844
10.8.3 Obtain Object Meta.....	845
10.8.4 List objects.....	845
10.8.5 Copy an object.....	846
10.8.6 Delete an object.....	847
10.9 Data security.....	848
10.10 Signed URLs.....	850
10.11 Authorized access.....	851
10.12 IMG.....	856
10.13 Exception handling.....	858
11 Node. js.....	860
11.1 Installation.....	860
11.2 Options.....	862
11.3 Quick start.....	863
11.4 Initialization.....	865
11.5 Manage a bucket.....	866
11.6 Upload objects.....	868
11.6.1 Overview.....	868
11.6.2 Upload a local file.....	869
11.6.3 Upload from local memory.....	870
11.6.4 Streaming upload.....	870
11.6.5 Multipart upload.....	871
11.6.6 Resumable upload.....	874
11.7 Download objects.....	875
11.7.1 Overview.....	875
11.7.2 Download to a local file.....	875
11.7.3 Download to local memory.....	876
11.7.4 Streaming download.....	876
11.7.5 HTTP-based download.....	877
11.8 Manage objects.....	877
11.8.1 Overview.....	877

11.8.2 Determine whether an object exists.....	878
11.8.3 Manage Object Meta.....	878
11.8.4 List all objects in a bucket.....	879
11.8.5 List the objects and subfolders of a folder.....	880
11.8.6 Copy an object.....	881
11.8.7 Delete objects.....	882
11.9 CNAME.....	883
11.10 Authorized access.....	884
11.11 Set ACLs.....	887
11.12 Manage lifecycle rules.....	889
11.13 Set logging.....	891
11.14 Static website hosting.....	893
11.15 Anti-leech.....	894
11.16 Image processing.....	896
11.17 Exception handling.....	902
11.18 FAQ.....	903
12 Browser.js.....	905
12.1 Installation.....	905
12.2 Quick start.....	909
12.3 Options.....	915
12.4 Upload objects.....	916
12.5 Download objects.....	922
12.6 Manage objects.....	922
12.7 CNAME.....	926
12.8 Authorized access.....	927
12.9 Set ACLs.....	930
12.10 Image processing.....	931
12.11 Browser.....	937
12.12 Exception handling.....	942
12.13 FAQ.....	942
13 Ruby.....	946
13.1 Installation.....	946
13.2 Quick start.....	949
13.3 Manage buckets.....	952
13.4 Upload objects.....	954
13.5 Download objects.....	959
13.6 Manage objects.....	962
13.7 CNAME.....	966
13.8 Use STS to access OSS.....	967
13.9 Set ACLs.....	969
13.10 Manage lifecycle rules.....	970
13.11 Set logging.....	972
13.12 Static website hosting.....	973
13.13 Anti-leech.....	974

13.14 CORS.....	976
13.15 Exception handling.....	977
13.16 Application of Rails.....	978
14 Media-C.....	987
14.1 Preface.....	987
14.2 Installation.....	988
14.3 Initialization.....	991
14.4 Client operations.....	992
14.5 Server operations.....	999
14.6 HLS basic interfaces.....	1010
14.7 HLS encapsulated interfaces.....	1015
14.8 Scenarios.....	1019

1 Introduction

Before using OSS SDK, you must:

- Learn about and activate the [Alibaba Cloud OSS service](#).
- [Create an AccessKey](#).

This topic lists the topics that describe how to install, develop, and troubleshoot OSS SDKs in the following mainstream languages and provide parameters and examples.

Language	Reference
Java	Java SDK references
Python	Python SDK references
Android	Android SDK references
iOS	iOS SDK references
.NET	.NET SDK references
Node.js	Node.js SDK references
Browser.js	Browser.js SDK references
PHP	PHP SDK references
C	C SDK references
Ruby	Ruby SDK references
Go	Go SDK references
Media-C	Media-C SDK references

2 Java

2.1 Preface

This document is written based on OSS Java SDK 2.8.3.

Compatibility

- For SDK versions 2.x.x:
 - API: compatible
 - Namespace: compatible
- For SDK versions 1.0.x:
 - API: compatible
 - Namespace: incompatible. The Table Store code in version 1.0.x is removed from version 2.0.0. The package names `com . aliyun . openservic es .` * and `com . aliyun . openservic es . oss . *` are replaced with `com . aliyun . oss . *`.

SDK source code and API documents

For the source code of OSS Java SDK, see [GitHub](#). For more information, see [OSS Java SDK API documents](#).

Code samples

OSS SDK for Java provides a variety of code samples for your reference. The following table describes the content of code samples.

Sample file	Content
GetStartedSample.java	Quick start
CreateFolderSample.java	Create a folder in Simple upload .
AppendObjectSample.java	Append upload
UploadSample.java	Resumable upload
MultipartUploadSample.java	Multipart upload
GetProgressSample.java	Upload progress bars
CallbackSample.java	Upload callback

Sample file	Content
CRCSample.java	CRC for file upload
SimpleGetObjectSample.java	Download objects
DownloadSample.java	Resumable download
ObjectMetaSample.java	Manage Object Meta
ListObjectsSample.java	List objects
UploadPartCopySample.java	Copy objects
DeleteObjectsSample.java	Delete objects
BucketOperationsSample.java	Bucket configurations such as authorized access , lifecycle management , access log files, anti-leech , and CORS .
ImageSample.java	Image processing
PostObjectSample.java	PostObject is independent of Java SDK.
ConcurrentGetObjectSample.java	Download multiple objects at the same time. Resumable downloads are recommended.

2.2 Installation

This topic describes how to install OSS Java SDK.

Preparation

- [Environment requirements](#)

You must use Java 1.8 or later versions.

- [Check the Java version](#)

Run `java -version` to check Java version.

Download SDK

- [Direct download](#)
- [Download from GitHub](#)
- [Download previous versions](#)

Install SDK

- Method One: Add dependencies to Maven (recommended)

To use OSS Java SDK in Maven, you only need to add the corresponding dependencies to the pom.xml file. This example uses OSS Java SDK 2.8.3. Add the following content to <dependencies>:

```
< dependency >
  < groupId > com . aliyun . oss </ groupId >
  < artifactId > aliyun - sdk - oss </ artifactId >
  < version > 2 . 8 . 3 </ version >
</ dependency >
```

- Method Two: Import jar files into Eclipse

Take OSS Java SDK 2.8.3 as an example. To import jar files into Eclipse, follow these steps:

1. Download the [Java SDK](#).
2. Decompress the Java SDK.
3. Copy the *aliyun - sdk - oss - 2 . 8 . 3 . jar* file and all files from the lib folder to your project.
4. Right-click your project name in Eclipse. Select Properties > Java Build Path > Add JARs.
5. Select all jar files you have copied in step 3 and import them to Libraries.

- Method Three: Import jar files into IntelliJ IDEA

Take OSS Java SDK 2.8.3 as an example. To import jar files into IntelliJ IDEA, follow these steps:

1. Download the [Java SDK](#).
2. Decompress the Java SDK.
3. Copy the *aliyun - sdk - oss - 2 . 8 . 3 . jar* file and all jar files from the lib folder to your project.
4. Right-click your project name in IntelliJ IDEA. Select File > Project Structure > Modules > Dependencies > + > JARs or directories.
5. Select all jar files you have copied in step 3 and import them to External Libraries.

2.3 Quick start

This topic describes how to use OSS Java SDK to perform routine operations such as bucket creation, object uploads, and object downloads.

Sample project

OSS Java SDK provides Maven-based and Ant-based sample projects. You can compile and run the sample projects on local devices or develop your own applications based on the sample projects. For the compilation and running method of the projects, see the README.md in the project directory.

- Maven sample project: [aliyun-oss-java-sdk-demo-mvn.zip](#).
- Ant sample project: [aliyun-oss-java-sdk-demo-ant.zip](#).

Create a bucket

A bucket is a global namespace in OSS. It is similar to a data container that stores files.

. Run the following code to create a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Create a bucket .
ossClient . createBuck et ( bucketName );

// Close your OSSClient instance .
ossClient . shutdown ();
```

For more information about bucket naming rules, see naming conventions in [Basic concepts](#). For more information about how to create a bucket, see [Manage a bucket](#).

For more information about endpoints, see [Regions and endpoints](#).

Upload objects

Run the following code to upload a file to OSS:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Upload content to the specified bucket and save it
as the specified object name .
String content = " Hello OSS ";
ossClient . putObject ( bucketName , objectName , new ByteArrayI
nputStream ( content . getBytes ( ) ) );

// Close your OSSClient instance .
ossClient . shutdown ( );
```

For more information, see [Upload objects](#).

Download objects

Run the following code to obtain object content:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );
```

```
// Call ossClient . getObject to return an OSSObject
instance . The OSSObject instance contains the object
content and Object Meta .
OSSObject ossObject = ossClient . getObject ( bucketName ,
objectName );
// Call ossObject . getObjectContent to obtain object
InputStream . Read the InputStream to obtain object
content .
InputStream content = ossObject . getObjectContent ();
if ( content != null ) {
    BufferedRe ader reader = new BufferedRe ader ( new
InputStreamReader ( content ));
    while ( true ) {
        String line = reader . readLine ();
        if ( line == null ) break ;
        System . out . println ( "\ n " + line );
    }
    // If you do not close the reader after the
data is read , connection leaks may occur . Consequent
ly , no available connection s are left and an
exception occurs .
    content . close ();
}

// Close your OSSClient instance .
ossClient . shutdown ();
```

For more information, see [Download objects](#).

List objects

Use the following code to list objects in a bucket. You can list a maximum of 100 objects by default.

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Use ossClient . listObject s to return an ObjectList
ing instance . The ObjectList ing instance contains the
response from listObject .
ObjectList ing objectList ing = ossClient . listObject s (
bucketName );
// Use objectList ing . getObjectS ummaries to obtain the
descriptio ns of all objects .
```

```

    for ( OSSObjectSummary objectSummary : objectListing .
    getObjectSummaries () ) {
        System . out . println ( " - " + objectSummary . getKey () + "
        " +
            "( size = " + objectSummary . getSize () + " ) " );
    }

    // Close your OSSClient instance .
    ossClient . shutdown ();

```

For more information, see [List objects](#) in [Manage objects](#).

Delete objects

For the complete code of deleting objects, see [GitHub](#).

Use the following code to delete a specified object:

```

// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Delete objects .
ossClient . deleteObject ( bucketName , objectName );

// Close your OSSClient instance .
ossClient . shutdown ();

```

For more information, see [Delete objects](#) in [Manage objects](#).

2.4 Initialization

OSSClient serves as the OSS Java client to manage OSS resources such as buckets and objects. To initiate an OSS request with Java SDK, you need to initiate an OSSClient instance first and then modify the default configuration items of ClientConfiguration.

Create an OSSClient instance

To create an OSSClient instance, you need to specify an endpoint. For more information about endpoints, see [Regions and endpoints](#) and [Bind a custom domain name](#).

- Use an OSS domain to create an OSSClient instance

Use the following code to create an OSSClient instance with a domain assigned by OSS:

```
// This example uses endpoint China (Hangzhou). Specify
// the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS. We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance. To
// create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// Close your OSSClient instance.
ossClient.shutdown();
```

You can use one or more OSSClients for your project. In other words, you can use multiple OSSClients simultaneously.

- Use a custom domain (CNAME) to create an OSSClient instance

Run the following code to create an OSSClient instance with CNAME:

```
String endpoint = "< yourEndpoint >";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS. We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance. To
// create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "< yourAccessKeyId >";
```

```
String accessKeySecret = "< yourAccess KeySecret >";

// Create a ClientConfiguration instance . Modify
// parameters as required .
ClientConfiguration conf = new ClientConfiguration ();
// Enable CNAME . CNAME indicates a custom domain
// bound to a bucket .
conf . setSupportCname ( true );

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret , conf );

// Close your OSSClient instance .
ossClient . shutdown ();
```

**Note:**

ossClient.listBuckets is not available when CNAME is used.

- Create an OSSClient instance for Apsara Stack

Run the following code to create an OSSClient instance for Apsara Stack:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http:// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https:// ram .
// console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create a ClientConfiguration instance . Modify
// parameters as required .
ClientConfiguration conf = new ClientConfiguration ();
// Disable CNAME .
conf . setSupportCname ( false );

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret , conf );

// Close your OSSClient instance .
ossClient . shutdown ();
```

- Use an IP address to create an OSSClient instance

Run the following code to create an OSSClient instance with an IP address:

```
// In some special cases , use the IP address as
// an endpoint . Specify the actual IP address based on
// your requirements .
String endpoint = " http:// 10 . 10 . 10 . 10 ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
```



```

permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create ClientConfiguration .
ClientConfiguration conf = new ClientConfiguration ();
// Enable OSS access with the level - 2 domain .
Access with the level - 2 domain is disabled by
default . You need to configure this function
for OSS Java SDK versions 2 . 1 . 2 or earlier
because OSS Java SDK versions later than 2 . 1 . 2
automatically detect IP addresses .
conf . setSLDEnabled ( true );

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret , conf );

// Close your OSSClient instance .
ossClient . shutdown ();

```

- Use STS to create an OSSClient instance

Run the following code to create an OSSClient instance with Security Token Service (STS):

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String securityToken = "< yourSecurityToken >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret , securityToken );

// Close your OSSClient instance .
ossClient . shutdown ();

```

For more information, see [What is RAM and STS](#) and [Authorized access](#).

Configure an OSSClient instance

ClientConfiguration is a configuration class of OSSClient. ClientConfiguration is used to configure parameters such as user agents, host proxies, connection timeout, and the maximum number of connections. You can configure the following parameters.

Parameter	Description	Configuration method
MaxConnections	Specifies the maximum number of HTTP connections that can be enabled. The default value is 1024.	ClientConfiguration. setMaxConnections
SocketTimeout	Specifies the timeout time in milliseconds for data transmission at the sockets layer. The default value is 50,000.	ClientConfiguration. setSocketTimeout
ConnectionTimeout	Specifies the timeout time in milliseconds when connections are established. The default value is 50,000 milliseconds.	ClientConfiguration. setConnectionTimeout
ConnectionRequestTimeout	Specifies the timeout time in milliseconds for obtaining connections from the connection pool. No timeout is configured by default. No timeout is configured by default.	ClientConfiguration. setConnectionRequestTimeout
IdleConnectionTime	Specifies the idle connection timeout time. If the idle connection time in milliseconds exceeds the specified value, the connection is closed. The default value is 60,000.	ClientConfiguration. setIdleConnectionTime
MaxErrorRetry	Specifies the maximum number of retry attempts in the case of a request error. The default value is 3.	ClientConfiguration. setMaxErrorRetry
SupportCname	Specifies whether CNAME can be used as an endpoint. You can use CNAME as an endpoint by default.	ClientConfiguration. setSupportCname

Parameter	Description	Configuration method
SLDEnabled	Specifies whether access with the level-2 domain is enabled. Access with the level-2 domain is disabled by default.	ClientConfiguration.setSLDEnabled
Protocol	Specifies the protocol (HTTP or HTTPS) used to connect to OSS. The default value is HTTP.	ClientConfiguration.setProtocol
UserAgent	Specifies the user agent (the user agent in the HTTP header). The default value is “aliyun-sdk-java.”	ClientConfiguration.setUserAgent
ProxyHost	Specifies the IP address to access the proxy host.	ClientConfiguration.setProxyHost
ProxyPort	Specifies the port for the proxy host.	ClientConfiguration.setProxyPort
ProxyUsername	Specifies the username verified by the proxy host.	ClientConfiguration.setProxyUsername
ProxyPassword	Specifies the password verified by the proxy host.	ClientConfiguration.setProxyPassword

Run the following code to configure parameters for an OSSClient instance with ClientConfiguration:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create ClientConfiguration . ClientConfiguration is
// a configurat ion class of OSSClient . You can use
// ClientConfiguration to configure parameters such as
// user agents , host proxies , connection timeout , and the
// maximum number of connection s .
ClientConfiguration conf = new ClientConfiguration ();
```

```

// Configure the maximum HTTP connections allowed by
// an OSSClient instance. The default value is 1024.
conf.setMaxConnections(200);
// Configure the timeout time in milliseconds for
// data transmission at SSL. The default value is 50,
// 000. conf.setSocketTimeout(10000);
conf.setSocketTimeout(10000);
// Configure the timeout time in milliseconds when
// connections are established. The default value is
// 50,000.
conf.setConnectTimeout(10000);
// Configure the timeout time in milliseconds for
// retrieving connections from the connection pool. This
// function is disabled by default.
conf.setConnectRequestTimeout(1000);
// Configure idle connection timeout. If the idle
// connection time in milliseconds exceeds the specified
// value, the connection is closed. The default value is
// 60,000.
conf.setIdleConnectionTime(10000);
// Configure the maximum number of retry attempts in
// the case of a request error. The default value is
// 3.
conf.setMaxErrorRetry(5);
// Check whether CNAME can be used as an endpoint.
// You can use CNAME as an endpoint by default.
conf.setSupportCname(true);
// Check whether access with the level-2 domain is
// enabled. Access with the level-2 domain is disabled
// by default.
conf.setSLDEnabled(true);
// Configure the protocol (HTTP or HTTPS) used to
// connect to OSS. HTTP is used by default.
conf.setProtocol(Protocol.HTTP);
// Configure the user agent (the user agent in the
// HTTP header). The default value is "aliyun-sdk-java
// ."
conf.setUserAgent("aliyun-sdk-java");
// Configure the port for the proxy host.
conf.setProxyHost("< yourProxyHost >");
// Configure the username verified by the proxy host.
conf.setProxyUsername("< yourProxyUserName >");
// Configure the password verified by the proxy host.
conf.setProxyPassword("< yourProxyPassword >");

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId,
    accessKeySecret, conf);

// Close your OSSClient instance.

```

```
ossClient . shutdown ();
```

2.5 Manage a bucket

A bucket serves as a container that stores objects. Objects belong to a bucket.

Create a bucket

Run the following code to create a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Create a bucket .
String bucketName = "< yourBucket Name >";
// The default storage class and ACL of the created
// bucket are set to standard and private read
// respective ly .
ossClient . createBuck et ( bucketName );

// Close your OSSClient .
ossClient . shutdown ();
```

For more information about bucket naming rules, see [naming conventions](#) in [Basic concepts](#).

You can specify the ACL and the storage class when you create a bucket, as shown in [Bucket-level permissions](#) and [Storage class](#). To create Infrequent Access or archive buckets, use Java SDK versions 2.6.0 or later. The sample code is as follows:

```
CreateBuck etRequest createBuck etRequest = new CreateBuck
etRequest ( bucketName );
// Set the ACL to public read for a bucket . The
// default value is Private .
createBuck etRequest . setCannedA CL ( CannedAcce ssControll ist
. PublicRead );
// Set the storage class to Infrequent Access for a
// bucket . The default value is Standard .
createBuck etRequest . setStorage Cla ss ( StorageCla ss . IA );
```

```
ossClient.createBucket(createBucketRequest);
```

List buckets

Buckets are listed alphabetically. You can list all buckets or buckets that meet the specified conditions. To list Infrequent Access or archive buckets, use Java SDK versions 2.6.0 or later.

- List all buckets

Run the following code to list all buckets:

```
// This example uses endpoint China (Hangzhou). Specify
// the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS. We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance. To
// create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// List buckets.
List<Bucket> buckets = ossClient.listBuckets();
for (Bucket bucket : buckets) {
    System.out.println("- " + bucket.getName());
}

// Close your OSSClient.
ossClient.shutdown();
```

- List buckets with a specified prefix

Run the following code to list buckets with a specified prefix:

```
ListBucketsRequest listBucketsRequest = new ListBucketsRequest();
// List buckets by a specified prefix. If no prefix
// is specified, list all buckets.
listBucketsRequest.setPrefix("<yourBucketPrefix>");
BucketList bucketList = ossClient.listBuckets(listBucketsRequest);
for (Bucket bucket : bucketList.getBucketList()) {
    System.out.println("- " + bucket.getName());
}
```

```
}
```

- **List buckets by marker**

The value of the marker parameter indicates the name of a bucket. Run the following code to list buckets after the specified bucket (the value of marker):

```
ListBucket sRequest listBucket sRequest = new ListBucket
sRequest ();
// List buckets after the specified marker . If the
marker is not specified , list all buckets .
listBucket sRequest . setMarker ("< yourBucket Marker >");
BucketList bucketList = ossClient . listBucket s ( listBucket
sRequest );
for ( Bucket bucket : bucketList . getBucketL ist ()) {
    System . out . println (" - " + bucket . getName ());
}
```

- **List a specified number of buckets**

Run the following code to list a specified number of buckets:

```
ListBucket sRequest listBucket sRequest = new ListBucket
sRequest ();
// Set the number of buckets that can be listed
to 500 . The default value is 100 . The maximum
value is 1 , 000 .
listBucket sRequest . setMaxKeys ( 500 );
BucketList bucketList = ossClient . listBucket s ( listBucket
sRequest );
for ( Bucket bucket : bucketList . getBucketL ist ()) {
    System . out . println (" - " + bucket . getName ());
}
```

Determine whether a bucket exists

Run the following code to determine whether a specified bucket exists:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

boolean exists = ossClient . doesBucket Exist ("< yourBucket
Name >");

// Close your OSSClient .
```

```
ossClient . shutdown ();
```

Configure an ACL for a bucket

The ACL of a bucket includes the following permissions.

Permission	Description	Value in Java SDK
Private	The bucket owner and the authorized users can read and write objects in the bucket. Other users cannot perform any operation on the objects.	CannedAccessControlList. Private
Public read	The bucket owner and the authorized users can read and write objects in the bucket. Other users can only read the objects in the bucket. Authorize this permission with caution.	CannedAccessControlList. PublicRead
Public read-write	All users can read and write objects in the bucket . Authorize this permission with caution.	CannedAccessControlList. PublicReadWrite

Run the following code to configure the ACL for a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Set the ACL for a bucket to private read and
// write .
ossClient . setBucketA cl ("< yourBucket Name >", CannedAcce
ssControll ist . Private );

// Close your OSSClient .
```



```
ossClient . shutdown ();
```

Obtain the ACL for a bucket

Run the following code to obtain the ACL for a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );
// Obtain the access ACL for the bucket
AccessControlList acl = ossClient . getBucketAcl ( "<
yourBucket Name >");
System . out . println ( acl . toString ());

// Close your OSSClient .
ossClient . shutdown ();
```

Obtain the region of a bucket

Run the following code to obtain the region for a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

String location = ossClient . getBucketL ocation ( "< yourBucket
Name >");
System . out . println ( location );

// Close your OSSClient .
```

```
ossClient . shutdown ();
```

For more information about regions, see [Region](#) in [Basic concepts](#).

Obtain bucket information

Run the following code to obtain bucket information:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We strongly
// recommend that you create a RAM account and use it
// for API access and daily O & M . Log on to https
// :// ram . console . aliyun . com to create a RAM account .
String accessKeyI d = "< accessKeyI d >";
String accessKeyS ecret = "< accessKeyS ecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

Bucket informatio n includes regions ( region or location
), creation dates ( CreationDa te ), owners ( Owner ), and
permission s ( Grants ).
BucketInfo info = ossClient . getBucketI nfo ("< yourBucket
Name >");
// Obtain region informatio n .
info . getBucket (). getLocatio n ();
// Obtain the creation date .
info . getBucket (). getCreatio nDate ();
// Obtain owner informatio n .
info . getBucket (). getOwner ();
// Obtain permission informatio n .
info . getGrants ();

// Close your OSSClient .
ossClient . shutdown ();
```

Delete a bucket

Before a bucket is deleted, ensure that all objects in the bucket and fragments that are generated from multipart upload are deleted.



Note:

To delete the fragments that are generated from multipart upload, use [Bucket.ListMultipartUploads](#) to list all fragments, and then use [Bucket.AbortMultipartUpload](#) to delete the fragments.

Run the following code to delete a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
```

```
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https://ram.console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

/// Delete the bucket .
ossClient . deleteBucket ( "< yourBucket Name >");

// Close your OSSClient .
ossClient . shutdown ();
```

2.6 Single-connection bandwidth throttling

This topic describes how to ensure other applications have sufficient bandwidth when you perform upload and download operations on OSS. You can include the `x-oss-traffic-limit` parameter in your requests and set bandwidth limits.



Note:

For more information about the scenarios and precautions of the single-connection bandwidth throttling feature, see [#unique_47](#) in the OSS Developer Guide.

Configure bandwidth throttling for common uploads and downloads

Use the following code to configure bandwidth throttling when you upload and download objects to and from OSS:

```
// This example uses China ( Hangzhou ) as the
// endpoint . Specify the actual endpoint based on your
// requiremen ts .
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Security risks may arise if you log on with the
// AccessKey pair of an Alibaba Cloud account because
// the account has permission s on all API operations
// in OSS . We recommend that you log on with a
// RAM user account to call API operations or perform
// routine operations and maintenanc e . To create a RAM
// user account , log on to https://ram.console . aliyun .
// com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

String bucketName = "< yourBucket Name >"
String objectName = "< yourObject Name >"
```

```

String localFileName = "< yourLocalFileName >"
String downloadFileName = "< yourDownloadFileName >"

// Set the bandwidth limit to 100 KB / s ( 819200 bit / s ).
int limitSpeed = 100 * 1024 * 8 ;

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint ,
accessKeyId , accessKeySecret );

// Configure bandwidth throttling for the object to be
uploaded .
InputStream inputStream = new FileInputStream (
localFileName );
PutObjectRequest putObjectRequest = new PutObjectRequest (
bucketName , key , inputStream );
PutObjectRequest.setTrafficLimit ( limitSpeed );
ossClient.putObject ( putObjectRequest );

// Configure bandwidth throttling for the object to be
downloaded .
GetObjectRequest getObjectRequest = new GetObjectRequest (
bucketName , key );
getObjectRequest.setTrafficLimit ( limitSpeed );
File localFile = new File ( downloadFileName );
ossClient.getObject ( getObjectRequest , localFile );

ossClient.shutdown ();

```

Configure bandwidth throttling for uploads and downloads that use signed URLs

Use the following code to configure bandwidth throttling when you upload and download objects from and to OSS by using signed URLs.

```

// This example uses China ( Hangzhou ) as the
endpoint . Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// Security risks may arise if you log on with the
AccessKey pair of an Alibaba Cloud account because
the account has permissions on all API operations
in OSS . We recommend that you log on with a
RAM user account to call API operations or perform
routine operations and maintenance . To create a RAM
user account , log on to https :// ram . console . aliyun .
com .
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";

String bucketName = "< yourBucketName >";
String objectName = "< yourObjectName >";
String localFileName = "< yourLocalFileName >";
String downloadFileName = "< yourDownloadFileName >";

// Set the bandwidth limit to 100 KB / s ( 819200 bit / s ).
int limitSpeed = 100 * 1024 * 8 ;

// Create an OSSClient instance .

```

```

OSS ossClient = new OSSClientBuilder().build(endpoint,
accessKeyId, accessKeySecret);

// Create a signed URL for uploading an object and
set the valid period of the URL to 60s.
Date date = new Date();
date.setTime(date.getTime() + 60 * 1000);
GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(
bucketName, objectName, HttpMethod.PUT);
request.setExpiration(date);
request.setTrafficLimit(limitSpeed);
URL signedUrl = ossClient.generatePresignedUrl(request);
System.out.println("put object url " + signedUrl);

// Configure bandwidth throttling for the object to be
uploaded.
InputStream inputStream = new FileInputStream(
localFileName);
ossClient.putObject(signedUrl, inputStream, -1, null, true);

// Create a signed URL for downloading an object
and set the valid period of the URL to 60s.
date = new Date();
date.setTime(date.getTime() + 60 * 1000);
request = new GeneratePresignedUrlRequest(bucketName, key,
HttpMethod.GET);
request.setExpiration(date);
request.setTrafficLimit(limitSpeed);
signedUrl = ossClient.generatePresignedUrl(request);
System.out.println("get object url " + signedUrl);

// Configure bandwidth throttling for the object to be
downloaded.
GetObjectRequest getObjectRequest = new GetObjectRequest(
signedUrl, null);
ossClient.getObject(getObjectRequest, downloadFileName);

ossClient.shutdown();

```

2.7 Bucket tagging

By using the bucket tagging function, you can classify your buckets to manage them. For example, you can specify that only buckets with specified tags are returned in ListBucket operations.

- Only the bucket owner and authorized RAM users can add tags to a bucket. Otherwise, the 403 Forbidden error is returned with the error code: AccessDenied.
- You can add a maximum of 20 tags (key-value pairs) to a bucket.
- The maximum size of a key is 64 bytes. The key of a tag cannot be null and cannot be prefixed with `http://`, `https://`, or `Aliyun`.
- The maximum size of a tag value is 128 bytes. The value of a tag can be null.
- The key and value of a tag must be UTF-8 encoded.

- If you use PutBucketTagging to add tags to a bucket, the original tags added to the bucket are completely overwritten.

Add tags to a bucket

You can run the following code to add tags to a bucket:

```
// This example uses the China East 1 ( Hangzhou )
// endpoint . Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
// of an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM user account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Adds tags to the bucket .
SetBucketT aggingRequ est request = new SetBucketT
aggingRequ est ("< yourBucket Name >");
request . setTag ("< yourTagkey 1 >", "< yourTagVal ue1 >");
request . setTag ("< yourTagkey 2 >", "< yourTagVal ue2 >");
ossClient . setBucketT agging ( request );

// Closes the OSSClient .
ossClient . shutdown ();
```

Obtain the tags added to a bucket

You can run the following code to obtain the tags to a bucket:

```
// This example uses the China East 1 ( Hangzhou )
// endpoint . Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
// of an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM user account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Obtains the tags added to the bucket .
```

```

TagSet tagSet = ossClient . getBucketTags ( new
GenericRequest ("< yourBucket Name >"));
Map < String , String > tags = tagSet . getAllTags ();

// Closes the OSSClient instance .
ossClient . shutdown ();

```

List buckets with a specified tag

You can run the following code to list buckets with a specified tag:

```

// This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM user account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint ,
accessKeyId , accessKeySecret );

// Lists buckets with the specified tag .
ListBucketRequest listBucketRequest = new ListBucket
Request ();
listBucketRequest . setTag ("< yourTagKey >", "< yourTagValue
>");
BucketList bucketList = ossClient . listBuckets ( listBucket
Request );
for ( Bucket o : bucketList . getBucketList () ) {
    System . out . println (" list result bucket : " + o . getName
());
}

// Closes the OSSClient instance .
ossClient . shutdown ();

```

Delete the tags added to a bucket

You can run the following code to delete the tags added to a bucket:

```

// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To

```

```

create a RAM user account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint ,
accessKeyId , accessKeySecret );

// Deletes the tags added to the bucket .
ossClient . deleteBucketTagging ( new GenericRequest ("<
yourBucket Name >"));

// Closes the OSSClient instance .
ossClient . shutdown ();

```

2.8 Object tagging

2.8.1 Configure object tagging

This topic describes how to configure object tagging.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).



Note:

- You can add tags to an object when uploading it or add tags for an uploaded object. If add tags to an object that already has tags, the original tags are overwritten. For more information about adding tags, see [#unique_52](#).
- To add tags to an object, you must have the permission to call PutObjectTagging.
- The Last-Modified value of an object is not updated when its tags are changed.
- A tag can contain letters, numbers, spaces, and the following symbols: + - = . _ : /

Add tags to an object when uploading it

- Add tags to an object when uploading it by calling putObject.

Run the following code to add tags to an object when uploading it by calling putObject:

```

// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with the
AccessKey of an Alibaba Cloud account because the
account has permissions on all the APIs in OSS
. We recommend that you log on as a RAM user
to access APIs or perform routine operations and

```



```

maintenanc e . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint
, accessKeyI d , accessKeyS ecret );

Map < String , String > tags = new HashMap < String , String
>();
tags . put ( " key1 " , " value1 " );
tags . put ( " key2 " , " value2 " );

// Configures the tags in the HTTP headers .
ObjectMeta data metadata = new ObjectMeta data ();
metadata . setObjectT agging ( tags );

// Uploads the object and add tags to it .
String content = "< yourtConte nt >";
ossClient . putObject ( bucketName , objectName , new
ByteArrayI nputStream ( content . getBytes () ) , metadata );

// Closes the OSSClient .
ossClient . shutdown ();

```

- Add tags to an object when uploading it in the multipart upload method.

Run the following code to add tags to an object when uploading it in the multipart upload method.

```

// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with the
AccessKey of an Alibaba Cloud account because the
account has permission s on all the APIs in OSS
. We recommend that you log on as a RAM user
to access APIs or perform routine operations and
maintenanc e . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";
String localFile = "< yourLocalF ile >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint
, accessKeyI d , accessKeyS ecret );

/* Step 1 : Initialize s a multipart upload task .
*/
// Configures the tags in the HTTP headers .
Map < String , String > tags = new HashMap < String , String
>();
tags . put ( " key1 " , " value1 " );
tags . put ( " key2 " , " value2 " );

```

```

ObjectMeta data metadata = new ObjectMeta data ();
metadata . setObjectT agging ( tags );

// Sends an InitiateMultipartUpload request and
// adds tags to the object .
InitiateMultipartUpload request = new InitiateMultipartUpload
( bucketName , objectName );
InitiateMultipartUpload result = ossClient .
initiateMultipartUpload ( request );
// Obtains the uploadId , which uniquely identifies the
// multipart upload task . You can use it to cancel
// or query the multipart upload task .
String uploadId = result . getUploadId ();

/* Step 2 : Uploads the parts .
*/
// The partETags parameter is a set of PartTags . A
// PartTag is composed of the ETag and number of
// the part .
List < PartTag > partETags = new ArrayList < PartTag > ();
// Calculates the number of parts that the object
// can be divided into .
final long partSize = 1 * 1024 * 1024L ; // 1MB
final File sampleFile = new File ("< localFile >");
long fileLength = sampleFile . length ();
int partCount = ( int ) ( fileLength / partSize );
if ( fileLength % partSize != 0 ) {
    partCount ++;
}
// Uploads the parts one by one .
for ( int i = 0 ; i < partCount ; i ++ ) {
    long startPos = i * partSize ;
    long curPartSize = ( i + 1 == partCount ) ? (
fileLength - startPos ) : partSize ;
    InputStream instream = new FileInputStream (
sampleFile );
    // Skips the uploaded parts .
    instream . skip ( startPos );
    UploadPartRequest uploadPartRequest = new UploadPart
Request ();
    uploadPartRequest . setBucketName ( bucketName );
    uploadPartRequest . setKey ( objectName );
    uploadPartRequest . setUploadId ( uploadId );
    uploadPartRequest . setInputStream ( instream );
    // Sets the part size . The size of a part
// must be equal to or larger than 100 KB . However
// , the size of the last part is not limited .
    uploadPartRequest . setPartSize ( curPartSize );
    // Sets the serial number of each part . Each
// part has a serial number ranging from 1 to 10 ,
// 000 . If the serial number of a part exceeds the
// range , the InvalidArgument error code is returned .
    uploadPartRequest . setPartNumber ( i + 1 );
    // The parts can be uploaded in different orders
// and even from different clients . OSS combines the
// parts following the serial numbers into the complete
// object .
    UploadPartResult uploadPartResult = ossClient .
uploadPart ( uploadPartRequest );
    // A PartTag is returned each time when a part
// is uploaded and is stored in the partETags
// parameter .
    partETags . add ( uploadPartResult . getPartETag ());
}

```

```

}

/* Step 3 : Completes the multipart upload task .
*/
// Sorts the partETags in an ascending order .
Collection s . sort ( partETags , new Comparator < PartETag >()
{
    public int compare ( PartETag p1 , PartETag p2 ) {
        return p1 . getPartNum ber () - p2 . getPartNum ber
    };
});
// All valid partETags must be provided to perform
this step . OSS verifies the availabili ty of each
part according to the partETags and combines the
parts into the complete object .
CompleteMu ltipartUpl oadRequest completeMu ltipartUpl
oadRequest =
    new CompleteMu ltipartUpl oadRequest ( bucketName ,
objectName , uploadId , partETags );
ossClient . completeMu ltipartUpl oad ( completeMu ltipartUpl
oadRequest );

// Views the tags added to the object .
TagSet tagSet = ossClient . getObjectT agging ( bucketName ,
objectName );
Map < String , String > getTags = tagSet . getAllTags ();
System . out . println ( " object tagging : " + getTags . toString
());

// Closes the OSSClient .
ossClient . shutdown ();

```

- Add tags to an object when uploading it in the append upload method.

Run the following code to add tags to an object when uploading it in the append upload method:

```

// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with the
AccessKey of an Alibaba Cloud account because the
account has permission s on all the APIs in OSS
. We recommend that you log on as a RAM user
to access APIs or perform routine operations and
maintenanc e . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint
, accessKeyI d , accessKeyS ecret );

String content1 = " Hello OSS A \ n ";

```

```

String content2 = " Hello OSS B \ n ";
String content3 = " Hello OSS C \ n ";

Map < String , String > tags = new HashMap < String , String
>();
tags . put ( " key1 " , " value1 " );
tags . put ( " key2 " , " value2 " );

ObjectMeta data meta = new ObjectMeta data ();
// Configures the tags to be added to the object
that needs to be uploaded .
meta . setObjectT agging ( tags );
// Specifies the content type of the object to be
uploaded .
meta . setContent Type ( " text / plain " );

// Sets parameters by calling AppendObje ctRequest .
AppendObje ctRequest appendObje ctRequest = new AppendObje
ctRequest ( bucketName , objectName , new ByteArrayI
nputStream ( content1 . getBytes () ) , meta );

// Sets each parameter by calling AppendObje ctRequest .
// Sets the name of the bucket that the object is
uploaded to .
// appendObje ctRequest . setBucketN ame ( "< yourBucket Name >" );
// Sets the object name .
// appendObje ctRequest . setKey ( "< yourObject Name >" );
// Sets the content to be appended to the object ,
which has two types : InputStrea m and File . This
line sets the type to InputStrea m .
// appendObje ctRequest . setInputSt ream ( new ByteArrayI
nputStream ( content1 . getBytes () ) );
// Sets the content to be appended to the object ,
which has two types : InputStrea m and File . This
line sets the type to File .
// appendObje ctRequest . setFile ( new File ( "< yourLocalF ile
>" ) );
// Sets the metadata of the object . The metadata
settings take effect only when the object is
appended for the first time .
// appendObje ctRequest . setMetadat a ( meta );

// Appends the object for the first time . Only the
tags set when the object is appended for the
first time are added to the object .
// Sets the position from where the object is
appended .
appendObje ctRequest . setPositio n ( 0L );
AppendObje ctResult appendObje ctResult = ossClient .
appendObje ct ( appendObje ctRequest );
// Outputs the CRC64 value of the object , which is
calculated according to the ECMA - 182 standard .
System . out . println ( appendObje ctResult . getObjectC RC ( ) );

// Appends the object for the second time .
// The nextPositi on parameter indicates the position (
object size ) required for the next upload request .
nextPositi on .
appendObje ctRequest . setPositio n ( appendObje ctResult .
getNextPos ition ( ) );
appendObje ctRequest . setInputSt ream ( new ByteArrayI
nputStream ( content2 . getBytes () ) );
appendObje ctResult = ossClient . appendObje ct ( appendObje
ctRequest );

```

```
// Appends the object for the third time .
appendObjectRequest.setPosition(appendObjectResult.
getNextPosition());
appendObjectRequest.setInputStream(new ByteArrayI
nputStream(content3.getBytes()));
appendObjectResult = ossClient.appendObject(appendObject
Request);

// Views the tags added to the uploaded object .
TagSet tagSet = ossClient.getObjectTagging(bucketName ,
objectName);
Map<String , String> getTags = tagSet.getAllTags();
System.out.println(" object tagging : "+ getTags.toString
());

// Closes the OSSClient .
ossClient.shutdown();
```

- Add tags to an object when uploading it in the resumable upload method.

Run the following code to add tags to an object when uploading it in the resumable upload method:

```
// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with the
AccessKey of an Alibaba Cloud account because the
account has permission s on all the APIs in OSS
. We recommend that you log on as a RAM user
to access APIs or perform routine operations and
maintenance . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientBuilder().build( endpoint
, accessKeyId , accessKeySecret );

// Configures the tags to be added to the object .
Map<String , String> tags = new HashMap<String , String
>();
tags.put(" key1 ", " value1 ");
tags.put(" key2 ", " value2 ");

ObjectMeta data meta = new ObjectMeta data ();
// Specifies the content type of the object to be
uploaded .
meta.setContentType(" text / plain ");
// Sets the tags to be added to the object .
meta.setObjectTagging( tags );

// Sets parameters by calling UploadFile Request .
UploadFile Request uploadFile Request = new UploadFile
Request("< yourBucket Name >","< yourObject Name >");

// Sets each parameter by calling UploadFile Request .
```

```
// Sets the name of the bucket that the object is
// uploaded to .
// uploadFile Request . setBucketName ("< yourBucket Name >");
// Sets the object name .
// uploadFile Request . setKey ("< yourObject Name >");
// Specifies the local file to be uploaded .
// uploadFile Request . setUploadFile ("< yourLocalFile >");
// Sets the number of upload tasks that can be
// simultaneously performed . The default value is 1 .
// uploadFile Request . setTaskNum ( 5 );
// Specifies the size of the parts to be uploaded
// . The part size ranges from 100 KB to 5 GB .
// The default value is : object size / 10000 .
// uploadFile Request . setPartSize ( 1 * 1024 * 1024 );
// Enables the resumable upload , which is disabled by
// default .
// uploadFile Request . setEnableCheckpoint ( true );
// Sets the local file used to store the results
// of resumable upload tasks . This parameter must be
// set if the resumable upload function is enabled
// . The upload progress is stored in the file . If
// a part fails to be uploaded , the next upload
// task continues from the position recorded in the
// file . This file is deleted after the object is
// uploaded and is stored in the same directory as
// the local file that needs to be uploaded , that is
// , uploadFile . ucp .
// uploadFile Request . setCheckpointFile ("< yourCheckpointFile
// >");
// Sets the metadata of the object .
// uploadFile Request . setObjectMetadata ( meta );
// Sets the upload callback function . The parameter
// type is Callback .
// uploadFile Request . setCallback ("< yourCallbackEvent >");

// Starts the resumable upload task and adds the
// tags to the object .
// ossClient . uploadFile ( uploadFile Request );

// Views the tags added to the uploaded object .
// TagSet tagSet = ossClient . getObjectTagging ( bucketName ,
// objectName );
// Map < String , String > getTags = tagSet . getAllTags ();
// System . out . println (" object tagging : "+ getTags . toString
// ());

// Closes the OSSClient .
// ossClient . shutdown ();
```

Add tags to an uploaded object or modify the tags added to an object

Run the following code to add tags to an uploaded object or modify the tags added to an object:

```
// This example uses the China East 1 ( Hangzhou )
// endpoint . Specify an actual endpoint based on your
// requirements .
// String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
// ";
// It is highly risky to log on with the AccessKey
// of an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
```

```

    that you log on as a RAM user to access APIs
    or perform routine operations and maintenance. To
    create a RAM user account, log on to https://ram.
    console.aliyun.com.
    String accessKeyId = "< yourAccess KeyId >";
    String accessKeySecret = "< yourAccess KeySecret >";
    String bucketName = "< yourBucket Name >";
    String objectName = "< yourObject Name >";

    // Creates an OSSClient instance.
    OSS ossClient = new OSSClientBuilder().build(endpoint,
    accessKeyId, accessKeySecret);

    Map<String, String> tags = new HashMap<String, String>
    >();
    tags.put("key1", "value1");
    tags.put("key2", "value2");

    // Adds tags to the object.
    ossClient.setObjectTagging(bucketName, objectName, tags);

    // Closes the OSSClient.
    ossClient.shutdown();

```

Add tags to an object when copying it

You can specify one of the following tagging rules when copying an object:

- **Copy (default):** Copy the tags of the source object to the destination object.
- **Replace:** Add the tags specified in the request to the destination object.

Examples of adding tags to objects of different sizes when copying them are described as follows:

- Run the following code to add tags to an object smaller than 1 GB when directly copying it:

```

// This example uses the China East 1 (Hangzhou)
// endpoint. Specify an actual endpoint based on your
// requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.
com";
// It is highly risky to log on with the
// AccessKey of an Alibaba Cloud account because the
// account has permissions on all the APIs in OSS.
// We recommend that you log on as a RAM user
// to access APIs or perform routine operations and
// maintenance. To create a RAM user account, log on
// to https://ram.console.aliyun.com.
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

String sourceBucketName = "< yourSource BucketName >";
String sourceObjectName = "< yourSource ObjectName >";
String destinationBucketName = "< yourDestinationBucket
Name >";
String destinationObjectName = "< yourDestinationObjec
tName >";

```

```
// Creates an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint
, accessKeyId , accessKeySecret );

// Creates a CopyObject Request object .
CopyObject Request copyObject Request = new CopyObject
Request ( sourceBucketName , sourceObjectName , destinatio
nBucketName , destinatio nObjectName );

// Configures the tags to be added to the object
. If the headers are not configured , the tags of
the source object are added to the destinatio n
object by default .
Map < String , String > headers = new HashMap < String ,
String > ();
headers . put ( OSSHeaders . COPY_OBJECT_TAGGING_ DIRECTIVE , "
REPLACE " );
headers . put ( OSSHeaders . OSS_TAGGING , " key1 = value1 & key2
= value2 " );
copyObject Request . setHeaders ( headers );

// Copies the source object to the destinatio n
object .
CopyObject Result result = ossClient . copyObject (
copyObject Request );
System . out . println ( " ETag : " + result . getETag () + "
LastModified : " + result . getLastModified () );

// Views the tags added to the destinatio n object .
TagSet tagSet = ossClient . getObjectTagging ( bucketName ,
destinationObjectName );
Map < String , String > getTags = tagSet . getAllTags ();
System . out . println ( " dest object tagging : "+ getTags .
toString () );

// Closes the OSSClient .
ossClient . shutdown ();
```

- Run the following code to add tags to an object larger than 1 GB when copying it in the multipart copy method:

```
// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with the
AccessKey of an Alibaba Cloud account because the
account has permissions on all the APIs in OSS
. We recommend that you log on as a RAM user
to access APIs or perform routine operations and
maintenance . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

String sourceBucketName = "< yourSource BucketName >";
String sourceObjectName = "< yourSource ObjectName >";
String destinationBucketName = "< yourDestinationBucke
tName >";
String destinationObjectName = "< yourDestin ationObjec
tName >";
```



```

ObjectMeta data objectMeta data = ossClient . getObjectM
etadata ( sourceBuck etName , sourceObje ctName );
// Obtains the size of the source object .
long contentLen gth = objectMeta data . getContent Length
();
// Sets the part size to 10 MB .
long partSize = 1024 * 1024 * 10 ;
// Calculates the number of parts .
int partCount = ( int ) ( contentLen gth / partSize );
if ( contentLen gth % partSize != 0 ) {
    partCount ++;
}

System . out . println ( " total part count : " + partCount );

// Configures the tags in the HTTP headers .
Map < String , String > tags2 = new HashMap < String ,
String > ();
tags2 . put ( " key3 " , " value3 " );
tags2 . put ( " key4 " , " value4 " );

ObjectMeta data metadata = new ObjectMeta data ();
metadata . setObjectT agging ( tags2 );

// Initialize s a copy task and sets the tags to
be added to the destinatio n object .
InitiateMu ltipartUpl oadRequest initiateMu ltipartUpl
oadRequest = new InitiateMu ltipartUpl oadRequest (
destinatio nBucketNam e , destinatio nObjectNam e , metadata
);
InitiateMu ltipartUpl oadResult initiateMu ltipartUpl
oadResult = ossClient . initiateMu ltipartUpl oad ( initiateMu
ltipartUpl oadRequest );
String uploadId = initiateMu ltipartUpl oadResult .
getUploadI d ();

// Copy the object part by part .
List < PartETag > partETags = new ArrayList < PartETag > ();
for ( int i = 0 ; i < partCount ; i ++ ) {
    // Calculates the part size .
    long skipBytes = partSize * i ;
    long size = partSize < contentLen gth - skipBytes ?
partSize : contentLen gth - skipBytes ;

    // Creates an UploadPart CopyReques t . You can
specify conditions by setting UploadPart CopyReques t .
    UploadPart CopyReques t uploadPart CopyReques t =
new UploadPart CopyReques t ( sourceBuck etName ,
sourceObje ctName , destinatio nBucketNam e , destinatio
nObjectNam e );
    uploadPart CopyReques t . setUploadI d ( uploadId );
    uploadPart CopyReques t . setPartSiz e ( size );
    uploadPart CopyReques t . setBeginIn dex ( skipBytes );
    uploadPart CopyReques t . setPartNum ber ( i + 1 );

    UploadPart CopyResult uploadPart CopyResult = ossClient .
uploadPart Copy ( uploadPart CopyReques t );
    // Stores the returned ETags of parts in
partETags .
    partETags . add ( uploadPart CopyResult . getPartETa g ());
}

// Completes the multipart copy task .

```

```

CompleteMu ltipartUpl oadRequest completeMu ltipartUpl
oadRequest = new CompleteMu ltipartUpl oadRequest (
    destinatio nBucketNam e , destinatio
nObjectNam e , uploadId , partETags );
CompleteMu ltipartUpl oadResult completeMu ltipartUpl
oadResult = ossClient . completeMu ltipartUpl oad ( completeMu
ltipartUpl oadRequest );

// Views the tags added to the source object .
TagSet tagSet = ossClient . getObjectT agging ( bucketName ,
sourceObje ctName );
Map < String , String > getTags = tagSet . getAllTags ();
System . out . println ( " src object tagging : " + getTags .
toString () );

// Views the tags added to the destinatio n object .
tagSet = ossClient . getObjectT agging ( bucketName ,
destinatio nObjectNam e );
getTags = tagSet . getAllTags ();
System . out . println ( " dest object tagging : " + getTags .
toString () );

// Closes the OSSClient .
ossClient . shutdown ();

```

Add tags to a symbolic link object

Run the following code to add tags to a symbolic link object:

```

// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM user account , log on to https :// ram .
console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String symLink = "< yourSymLin k >";
String destinatio nObjectNam e = "< yourDestin ationObjec
tName >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Configures the tags to be added to the symbolic
link object .
Map < String , String > tags = new HashMap < String , String
> ();
tags . put ( " key1 " , " value1 " );
tags . put ( " key2 " , " value2 " );

// Generates the metadata of the object .
ObjectMeta data metadata = new ObjectMeta data ();
metadata . setObjectT agging ( tags );

```

```
// Creates a CreateSyml inkRequest .
CreateSyml inkRequest createSyml inkRequest = new CreateSyml
inkRequest ( bucketName , symlink , destinatio nObjectNam e );

// Sets the metadata of the symbolic link object .
createSyml inkRequest . setMetadat a ( metadata );

// Creates a symbolic link object .
ossClient . createSyml ink ( createSyml inkRequest );

// Views the tags added to the symbolic link object .
TagSet tagSet = ossClient . getObjectT agging ( bucketName ,
symlink );
Map < String , String > getTags = tagSet . getAllTags ();
System . out . println ( " symlink tagging : " + getTags . toString
());

// Closes the OSSClient .
ossClient . shutdown ();
```

2.8.2 Obtain the tags added to an object

This topic describes how to obtain the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to obtain the tags added to an object:

```
// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM user account , log on to https :// ram .
console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Obtains the tags added to the object .
TagSet tagSet = ossClient . getObjectT agging ( bucketName ,
objectName );
Map < String , String > tags = tagSet . getAllTags ();

// Views the tags added to the object .
System . out . println ( " object tagging : " + tags . toString ());

// Closes the OSSClient .
```

```
ossClient . shutdown ();
```

For more information about how to obtain the tags added to an object, see [#unique_54](#).

2.8.3 Delete the tags added to an object

This topic describes how to delete the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to delete the tags added to an object:

```
// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM user account , log on to https :// ram .
console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Deletes the tags added to the object .
ossClient . deleteObje ctTagging ( bucketName , objectName );
```

For more information about how to delete the tags added to an object, see [#unique_56](#).

2.8.4 Object tagging and lifecycle rules

Lifecycle rules can take effect on objects with specified prefixes or tags. You can also specify prefixes and tags as the condition of a lifecycle rule at the same time.



Note:

If you set a specified tag as the condition of a lifecycle rule, the rule applies to an object only when the key and value in the tag of the object both match the specified tag. If you specify a prefix and multiple tags as the condition of a lifecycle rule, the

rule applies to an object only when the prefix and tags of the object match the prefix and all tags specified in the rule.

Set tags as the matching condition of a lifecycle rule

Run the following code to set tags as the matching condition of a lifecycle rule:

```
// This example uses the China East 1 ( Hangzhou )
// endpoint . Specify an actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
// of an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM user account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Creates a SetBucketL ifecycleRe quest .
SetBucketL ifecycleRe quest request = new SetBucketL
ifecycleRe quest ( bucketName );

// Sets the rule ID and the prefixes and tags used
// as the matching condition .
String ruleId0 = " rule0 ";
String matchPrefi x0 = " A0 /";
Map < String , String > matchTags0 = new HashMap < String ,
String >();
matchTags0 . put ( " key0 " , " value0 " );

String ruleId1 = " rule1 ";
String matchPrefi x1 = " A1 /";
Map < String , String > matchTags1 = new HashMap < String ,
String >();
matchTags1 . put ( " key1 " , " value1 " );

String ruleId2 = " rule2 ";
String matchPrefi x2 = " A2 /";

String ruleId3 = " rule3 ";
String matchPrefi x3 = " A3 /";

// Sets an expiration rule for objects so that
// objects are expired three days after they are
// modified for the last time .
LifecycleR ule rule = new LifecycleR ule ( ruleId0 ,
matchPrefi x0 , RuleStatus . Enabled , 3 );
rule . setTags ( matchTags0 )
request . AddLifecyc leRule ( rule );
```

```
// Sets an expiration rule for objects so that
objects created before the specified date are expired .
rule = new LifecycleRule ( ruleId1 , matchPrefix x1 ,
RuleStatus . Enabled );
rule . setCreated BeforeDate ( DateUtil . parseIso86 01Date ( " 2022
- 10 - 12T00 : 00 : 00 . 000Z " ));
rule . setTags ( matchTags1 )
request . AddLifecycleRule ( rule );

// Sets an expiration rules for parts so that parts
are expired 3 days after they are generated .
rule = new LifecycleRule ( ruleId2 , matchPrefix x2 ,
RuleStatus . Enabled );
LifecycleRule . AbortMulti partUpload abortMulti partUpload =
new LifecycleRule . AbortMulti partUpload ();
abortMulti partUpload . setExpirationDays ( 3 );
rule . setAbortMulti partUpload ( abortMulti partUpload );
request . AddLifecycleRule ( rule );

// Sets an expiration rules for parts so that parts
generated before the specified date are expired .
rule = new LifecycleRule ( ruleId3 , matchPrefix x3 ,
RuleStatus . Enabled );
abortMulti partUpload = new LifecycleRule . AbortMulti
partUpload ();
abortMulti partUpload . setCreated BeforeDate ( DateUtil .
parseIso86 01Date ( " 2022 - 10 - 12T00 : 00 : 00 . 000Z " ));
rule . setAbortMulti partUpload ( abortMulti partUpload );
request . AddLifecycleRule ( rule );

ossClient . setBucketLifecycle ( request );

// Closes the OSSClient .
ossClient . shutdown ();
```

View the tags set as the matching condition of a lifecycle rule

Run the following code to view the tags set as the matching condition of a lifecycle rule:

```
// This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM user account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Creates an OSSClient .
OSS ossClient = new OSSClientBuilder (). build ( endpoint ,
accessKeyId , accessKeySecret );

// Obtains the lifecycle rule set for the object .
```

```

List < LifecycleRule > rules = ossClient . getBucketLifecycle
( bucketName );

// Views the lifecycle rule set for the object .
for ( LifecycleRule rule1 : rules ) {
    // Views the rule ID .
    System . out . println ( " rule id : " + rule1 . getId () );

    // Views the prefixes set as the matching condition .
    System . out . println ( " rule prefix : " + rule1 . getPrefix
());

    // Views the tags set as the matching condition .
    if ( rule1 . hasTags () ) {
        System . out . println ( " rule tagging : " + rule1 . getTags
(). toString () );
    }

    // Views the expiration rules in which objects are
expired some days after they are modified for the
last time .
    if ( rule1 . hasExpirationDays () ) {
        System . out . println ( " rule expiration days : " + rule1
. getExpirationDays () );
    }

    // Views the expiration rules in which objects
created before a specified date are expired .
    if ( rule1 . hasCreatedBeforeDate () ) {
        System . out . println ( " rule expiration create before
days : " + rule1 . getCreatedBeforeDate () );
    }

    // Views the expiration rule for parts .
    if ( rule1 . hasAbortMultipartUpload () ) {
        if ( rule1 . getAbortMultipartUpload () . hasExpiration
Days () ) {
            System . out . println ( " rule abort multipart days : "
+ rule1 . getAbortMultipartUpload () . getExpirationDays () );
        }

        if ( rule1 . getAbortMultipartUpload () . hasCreated
BeforeDate () ) {
            System . out . println ( " rule abort multipart create
before date : " + rule1 . getAbortMultipartUpload () .
getCreatedBeforeDate () );
        }
    }
}

// Closes the OSSClient .
ossClient . shutdown ();

```

2.9 Server-side encryption

OSS supports server-side encryption for uploaded data. This means that when user data is uploaded, OSS encrypts the data and permanently stores the data. Then, when the data is downloaded by a user, OSS automatically decrypts the data, returns the

original data to the user, and declares in the header of the response that the data has been encrypted on the server.

The following server-side encryption methods are available for different application scenarios:

- Server-side encryption that uses CMKs managed by KMS for encryption and decryption (SSE-KMS)

When uploading an object, you can use a specified CMK ID or the default CMK managed by KMS to encrypt and decrypt a large amount of data. This method is cost-effective because you do not need to send user data to the KMS server through networks for encryption and decryption.



Notice:

API calling fees are incurred if you use CMKs to encrypt or decrypt data.

- Server-side encryption fully managed by OSS (SSE-OSS)

When you upload an object, OSS encrypts the object on the server side by using AES256 fully managed by OSS. In this method, OSS uses AES256 to encrypt each object with an individual key. Furthermore, the individual keys are encrypted by a CMK that is updated periodically for higher security. This method applies to encrypt or decrypt bulk data.



Notice:

- Only one server-side encryption method can be used for an object at one time.
- If you configure server-side encryption for a bucket, you can still configure the encryption method for a single object when uploading or copying it. In this case, the encryption method configured for the object takes precedence. For more information, see [PutObject](#).
- For more information about server-side encryption, see [Server-side encryption](#).

Configure server-side encryption for a bucket

You can run the following code to configure the default encryption method for a bucket. After the method is configured, all objects that are uploaded to the bucket

without their encryption methods configured are encrypted in the configured default encryption method.

```
// This example uses the China East 1 ( Hangzhou )
// endpoint . Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
// of an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM user account , log on to https :// ram .
// console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint ,
accessKeyId , accessKeySecret );

// Configures server - side encryption for the bucket .
ServerSide Encryption ByDefault applyServerSideEncryptionByDefault = new ServerSide Encryption ByDefault (
SSEAlgorithm . KMS );
applyServerSideEncryptionByDefault . setKMSMasterKeyID ("<
yourTestKeyId >");
ServerSide Encryption Configuration sseConfig = new
ServerSide Encryption Configuration ();
sseConfig . setApplyServerSideEncryptionByDefault ( applyServerSideEncryptionByDefault );
SetBucketEncryptionRequest request = new SetBucketEncryptionRequest ("< yourBucket Name >", sseConfig );
ossClient . setBucketEncryption ( request );

// Closes the OSSClient instance .
ossClient . shutdown ();
```

Obtain the server-side encryption settings of a bucket

You can run the following code to obtain the server-side encryption settings of a bucket:

```
// This example uses the China East 1 ( Hangzhou )
// endpoint . Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with the AccessKey
// of an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM user account , log on to https :// ram .
// console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance .
```

```

OSS ossClient = new OSSClientBuilder().build(endpoint,
accessKeyId, accessKeySecret);

// Obtains the server-side encryption settings of the
bucket.
ServerSideEncryptionConfiguration sseConfig = ossClient.
getBucketEncryption("< yourBucket Name >");
System.out.println("get Algorithm: " + sseConfig.
getApplyServerSideEncryptionByDefault().getSSEAlgorithm
());
System.out.println("get kmsid: " + sseConfig.getApplySe
rverSideEncryptionByDefault().getKMSEMasterKeyID());

// Closes the OSSClient instance.
ossClient.shutdown();

```

Delete the server-side encryption settings of a bucket

You can run the following code to delete the server-side encryption settings of a bucket:

```

// This example uses the China East 1 (Hangzhou)
endpoint. Specify the actual endpoint based on your
requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance. To
create a RAM user account, log on to https://ram.
console.aliyun.com.
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Creates an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint,
accessKeyId, accessKeySecret);

// Deletes the server-side encryption settings of the
bucket.
ossClient.deleteBucketEncryption("< yourBucket Name >");

// Closes the OSSClient instance.
ossClient.shutdown();

```

2.10 Upload objects

2.10.1 Overview

This topic is the overview of object upload in OSS Java SDK.

In OSS, an object is the basic unit for data operations. OSS Java SDK provides the following file upload methods:

- **Simple upload:** supports the upload of a file up to 5 GB in size. It includes streaming upload and file upload.
- **Form upload:** supports the upload of a file up to 5 GB in size.
- **Append upload:** supports the upload of a file up to 5 GB in size.
- **Resumable upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files. It supports concurrent upload and you can define the size of each part.
- **Multipart upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files.

**Note:**

For more information about the usage scenarios of each upload method, see “Upload files” in OSS Developer Guide.

During the file upload, you can set [Object Meta](#), and view the upload progress in the [upload progress bar](#). After you complete the file upload, you can perform [upload callback](#).

2.10.2 Simple upload

Simple upload includes streaming upload and file upload. Streaming upload uses `InputStream` as the source object. File upload uses a local file as the source object.

For the complete simple upload code, see [GitHub](#).

Streaming upload

Use `ossClient.putObject` to upload a data stream to OSS.

- Upload a string

Run the following code to upload a string:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
```

```

OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// Upload a string .
String content = " Hello OSS ";
ossClient . putObject ( "< yourBucket Name >", "< yourObjectName >", new ByteArrayInputStream ( content . getBytes ( ) ));

// Close your OSSClient instance .
ossClient . shutdown ();

```

- Upload a byte array

Run the following code to upload a byte array:

```

// This example uses endpoint China ( Hangzhou ). Specify the actual endpoint based on your requirements .
String endpoint = " http:// oss - cn - hangzhou . aliyuncs . com ";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS . We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance . To create a RAM account , log on to https:// ram . console . aliyun . com .
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// Upload a byte array .
byte [] content = " Hello OSS ". getBytes ( );
ossClient . putObject ( "< yourBucket Name >", "< yourObjectName >", new ByteArrayInputStream ( content ));

// Close your OSSClient instance .
ossClient . shutdown ();

```

- Upload a network stream

Run the following code to upload a network stream:

```

// This example uses endpoint China ( Hangzhou ). Specify the actual endpoint based on your requirements .
String endpoint = " http:// oss - cn - hangzhou . aliyuncs . com ";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS . We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance . To create a RAM account , log on to https:// ram . console . aliyun . com .
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

```

```
// Upload a network stream .
InputStream inputStream = new URL (" https :// www .
aliyun . com /"). openStream ();
ossClient . putObject ("< yourBucket Name >", "< yourObject Name
>", inputStream );

// Close your OSSClient instance .
ossClient . shutdown ();
```

- **Upload a file stream**

Run the following code to upload a file stream:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . To ensure
cloud security , we recommend that you follow best
practices of Resource Access Management and log on
as a RAM user to access APIs or perform routine
operations and maintenanc e . To create a RAM account
, log on to https :// ram . console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Upload a file stream .
InputStream inputStream = new FileInputS tream ("<
yourlocalF ile >");
ossClient . putObject ("< yourBucket Name >", "< yourObject Name
>", inputStream );

// Close your OSSClient instance .
ossClient . shutdown ();
```

File upload

Run the following code to upload a local file:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
```

```

OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Upload a file . < yourLocalFile > consists of a
local file path and a file name that includes an
extension , for example , / users / local / myfile . txt .
ossClient . putObject ( "< yourBucket Name >" , "< yourObject Name
>" , new File ( "< yourLocalFile >" ));

// Close your OSSClient instance .
ossClient . shutdown ();

```

2.10.3 Form upload

Form upload uses an HTML form to upload a file to a specified bucket. The maximum size of files is 5 GB for each form upload.

For more information about the application scenarios of form upload, see [Form upload](#) in OSS Developer Guide.

Run the following code for form upload:

```

public class PostObject Sample {
    // Upload a file .
    private String localFilePath = "< yourLocalFile >";
    // This example uses endpoint China ( Hangzhou ).
    Specify the actual endpoint based on your requirements .
    private String endpoint = " http :// oss - cn - hangzhou .
aliyuncs . com ";
    // It is highly risky to log on with AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
    private String accessKeyId = "< yourAccess KeyId >";
    private String accessKeySecret = "< yourAccess KeySecret
>";
    // Specify the bucket name .
    private String bucketName = "< yourBucket Name >";
    // Specify the file name .
    private String objectName = "< yourObject Name >";
    /**
     * Start form upload .
     * @ throws Exception
     */
    private void PostObject () throws Exception {
        // Add the bucket name to the URL . The URL
format is http :// yourBucket Name . oss - cn - hangzhou .
aliyuncs . com .
        String urlStr = endpoint . replace ( " http ://", " http
:/" + bucketName + "." );
        // Create Map for the form .
        Map < String , String > formFields = new LinkedHashMap
Map < String , String > ();
        // Specify the file name .
        formFields . put ( " key ", this . objectName );
        // Configure Content - Disposition .

```

```

        formFields . put ( " Content - Dispositio n ", " attachment ;
filename ="
        + localFileP ath );
        // Configure the callback parameter .
        Callback callback = new Callback ();
        // Configure the IP address of the server you
want to send the callback request to , for example ,
http :// oss - demo . aliyuncs . com : 23450 or http :// 127 . 0
. 0 . 1 : 9090 .
        callback . setCallbac kUrl ( "< yourCallba ckServerUr l
>");
        // Configure the host field value in the
callback request header , such as oss - cn - hangzhou .
aliyuncs . com .
        callback . setCallbac kHost ( "< yourCallba ckServerHo st
>");
        // Configure the body field value for the
callback request .
        callback . setCallbac kBody ( "{\\\\" mimeType \\\":${
mimeType },\\\\" size \\\":${ size }}" );
        // Configure Content - Type for the callback
request .
        callback . setCalback BodyType ( CalbackBod yType . JSON );
        // Configure the custom parameters used for
sending the callback request . Each custom parameter
consists of a key and a value . The key must be
in lowercase and start with x : .
        callback . addCallbac kVar ( " x : var1 ", " value1 " );
        callback . addCallbac kVar ( " x : var2 ", " value2 " );
        // Configure the callback parameter for Map in
the form .
        setCallBac k ( formFields , callback );
        // Configure OSSAccessK eyId .
        formFields . put ( " OSSAccessK eyId ", accessKeyI d );
        String policy = "{ \" expiration \": \" 2120 - 01 - 01T12
: 00 : 00 . 000Z \", \" conditions \": [ [ \" content - length - range
\", 0 , 104857600 ] ] }";
        String encodePoli cy = new String ( Base64 .
encodeBase 64 ( policy . getBytes () ));
        // Configure the policy .
        formFields . put ( " policy ", encodePoli cy );
        // Generate the URL for signature .
        String signatureec om = com . aliyun . oss . common
. auth . ServiceSig nature . create () . computeSig nature (
accessKeyS ecret , encodePoli cy );
        // Configure the signature .
        formFields . put ( " Signature ", signatureec om );
        String ret = formUpload ( urlStr , formFields ,
localFileP ath );
        System . out . println ( " Post Object [ " + this .
objectName + " ] to bucket [ " + bucketName + " ] " );
        System . out . println ( " post reponse : " + ret );
    }
    private static String formUpload ( String urlStr , Map <
String , String > formFields , String localFile )
        throws Exception {
        String res = "";
        HttpURLConnection conn = null ;
        String boundary = " 9431149156 168 ";
        try {
            URL url = new URL ( urlStr );
            conn = ( HttpURLConnection ) url . openConnec tion
();
            conn . setConnect Timeout ( 5000 );

```

```

        conn . setReadTim     eout ( 30000 );
        conn . setDoOutput    t ( true );
        conn . setDoInput     ( true );
        conn . setRequest     Method ( " POST " );
        conn . setRequest     Property ( " User - Agent ",
            " Mozilla / 5 . 0 ( Windows ; U ; Windows NT
6 . 1 ; zh - CN ; rv : 1 . 9 . 2 . 6 )" );
        // Configure the MD5 value . The MD5 value
is calculated based on the entire body .
        conn . setRequest     Property ( " Content - MD5 ", "<
yourContentMD5 >" );
        conn . setRequest     Property ( " Content - Type ",
            " multipart / form - data ; boundary =" +
boundary );
        OutputStream          out = new DataOutput Stream ( conn
. getOutputS tream ());
        // Recursively read all Map data in the
form and write the data to the OutputStream .
        if ( formFields != null ) {
            StringBuffer      strBuf = new StringBuffer ();
            Iterator < Entry < String , String >> iter =
formFields . entrySet (). iterator ();
            int i = 0 ;
            while ( iter . hasNext ()) {
                Entry < String , String > entry = iter . next
());
                String        inputName = entry . getKey ();
                String        inputValue = entry . getValue ();
                if ( inputValue == null ) {
                    continue ;
                }
                if ( i == 0 ) {
                    strBuf . append ("--"). append ( boundary ).
append ("\ r \ n ");
                    strBuf . append ( " Content - Dispositio n :
form - data ; name =" +
                        + inputName + "\" \ r \ n \ r \ n " );
                    strBuf . append ( inputValue );
                } else {
                    strBuf . append ("\ r \ n "). append ("--").
append ( boundary ). append ("\ r \ n ");
                    strBuf . append ( " Content - Dispositio n :
form - data ; name =" +
                        + inputName + "\" \ r \ n \ r \ n " );
                    strBuf . append ( inputValue );
                }
                i ++;
            }
            out . write ( strBuf . toString (). getBytes ());
        }
        // Read the file and write it to the
OutputStream          am .
        File            file = new File ( localFile );
        String          filename = file . getName ();
        String          contentType = new MimeTypeMap
(). getConte ntType ( file );
        if ( contentType == null || contentType . equals
("") ) {
            contentType = " applicatio n / octet - stream ";
        }
        StringBuffer    strBuf = new StringBuffer ();
        strBuf . append ("\ r \ n "). append ("--"). append (
boundary )
            . append ("\ r \ n ");

```



```

        strBuf . append ( " Content - Dispositio n : form - data
; name = \" file \"; "
        + " filename = \" " + filename + "\" \ r \ n " );
        strBuf . append ( " Content - Type : " + contentType e +
"\ r \ n \ r \ n " );
        out . write ( strBuf . toString () . getBytes () );
        DataInputS tream in = new DataInputS tream ( new
FileInputS tream ( file ));
        int bytes = 0 ;
        byte [] bufferOut = new byte [ 1024 ];
        while ( ( bytes = in . read ( bufferOut ) ) != - 1 ) {
            out . write ( bufferOut , 0 , bytes );
        }
        in . close ();
        byte [] endData = ( "\ r \ n --" + boundary + "--\ r \
n " ) . getBytes ();
        out . write ( endData );
        out . flush ();
        out . close ();
        // Read the returned data .
        strBuf = new StringBuffer ();
        BufferedRe ader reader = new BufferedRe ader (
new InputStrea mReader ( conn . getInputSt ream () ));
        String line = null ;
        while ( ( line = reader . readLine () ) != null ) {
            strBuf . append ( line ) . append ( "\ n " );
        }
        res = strBuf . toString ();
        reader . close ();
        reader = null ;
    } catch ( Exception e ) {
        System . err . println ( " Send post request
exception : " + e );
        throw e ;
    } finally {
        if ( conn != null ) {
            conn . disconnect ();
            conn = null ;
        }
    }
    }
    return res ;
}

private static void setCallBac k ( Map < String , String
> formFields , Callback callback ) {
    if ( callback != null ) {
        String jsonCb = OSSUtils . jsonizeCal lback (
callback );
        String base64Cb = BinaryUtil . toBase64St ring (
jsonCb . getBytes () );
        formFields . put ( " callback " , base64Cb );
        if ( callback . hasCallbac kVar () ) {
            Map < String , String > varMap = callback .
getCallbac kVar ();
            for ( Entry < String , String > entry : varMap .
entrySet () ) {
                formFields . put ( entry . getKey () , entry .
getValue () );
            }
        }
    }
}

public static void main ( String [] args ) throws
Exception {

```

```

        PostObject Sample ossPostObject = new PostObject
Sample ();
        ossPostObject . PostObject ();
    }
}

```

**Note:**

For more information about callback, see [Callback](#).

2.10.4 Append upload

This topic describes how to use append upload.

You cannot perform copyObject for appended objects.

Run the following code for append upload:

```

// This example uses endpoint China East 1 (Hangzhou
//). Specify the actual endpoint based on your
// requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS. We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance. To
// create a RAM account, log on to https://ram.console
//.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";

String content1 = "Hello OSS A \n";
String content2 = "Hello OSS B \n";
String content3 = "Hello OSS C \n";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId,
    accessKeySecret);

ObjectMetadata meta = new ObjectMetadata();
// Specify the content type of the object you want
// to upload.
meta.setContentType("text/plain");

// Configure parameters with AppendObjectRequest.
AppendObjectRequest appendObjectRequest = new AppendObject
    Request("<yourBucketName>", "<yourObjectName>", new
    ByteArrayInputStream(content1.getBytes()), meta);

// Configure a single parameter with AppendObjectRequest.
// Specify the bucket name.
// appendObjectRequest.setBucketName("<yourBucketName>");
// Specify the object name.
// appendObjectRequest.setKey("<yourObjectName>");
// Configure the type of content you want to append.
// Two types of content are available: InputStream
// and File. Select InputStream.

```

```

// appendObject request . setInputStream ( new   ByteArrayInputStream
    inputStream ( content1 . getBytes ());
// Configure the type of content you want to append
. Two types of content are available : InputStream
and File . Select File .
// appendObject request . setFile ( new   File ("< yourLocalFile
>"));
// Specify object meta , because the specified object
meta takes effect from the first append .
// appendObject request . setMetadata ( meta );

// Start the first append .
// Configure the position where the object is appended
based on the previous object length .
appendObject request . setPosition ( 0L );
AppendObjectResult appendObjectResult = ossClient .
appendObject ( appendObject request );
// Calculate the 64 - bit CRC value . The value is
calculated based on the ECMA - 182 standard . This value
is calculated based on ECMA - 182 criteria .
System . out . println ( appendObjectResult . getObjectCRC ());

// Start the second append .
// nextPosition indicates the position provided for the
next request , namely , the length of the current
object .
appendObject request . setPosition ( appendObjectResult .
getNextPosition ());
appendObject request . setInputStream ( new   ByteArrayInputStream
    inputStream ( content2 . getBytes ());
appendObjectResult = ossClient . appendObject ( appendObject
    request );

// Start the third append .
appendObject request . setPosition ( appendObjectResult .
getNextPosition ());
appendObject request . setInputStream ( new   ByteArrayInputStream
    inputStream ( content3 . getBytes ());
appendObjectResult = ossClient . appendObject ( appendObject
    request );

// Close your OSSClient .
ossClient . shutdown ();

```

For more information about append upload, see [Append object](#) in OSS Developer Guide. For the complete code of append upload, see [GitHub](#).

2.10.5 Resumable upload

You can use resumable upload to split a file you want to upload into several parts and upload them simultaneously. After you have uploaded all parts, you can combine these parts into a complete object. Thus, you can complete uploading the entire object.

For more information about resumable upload, see [Resumable upload](#) in the OSS Developer Guide. For the complete code of resumable upload, see [GitHub](#).

Run the following code for resumable upload:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

ObjectMeta data meta = new ObjectMeta data ();
// Specify the content type of the object you want
// to upload .
meta . setContent Type ( " text / plain " );

// Configure parameters with UploadFile Request .
UploadFile Request uploadFile Request = new UploadFile
Request ( "< yourBucket Name >","< yourObject Name >");

// Configure parameters with UploadFile Request .
// Specify the bucket name .
// uploadFile Request . setBucketN ame ( "< yourBucket Name >");
// Specify the object name .
// uploadFile Request . setKey ( "< yourObject Name >");
// Specify the local file you want to upload .
uploadFile Request . setUploadF ile ( "< yourLocalF ile >");
// Specify the number of threads for simultaneo us
// upload . The default value is 1 .
uploadFile Request . setTaskNum ( 5 );
// Specify the size of a part you want to upload
// . The value can be from 100 KB to 5 GB . The
// default size of a part is calculated as : file size
// / 10 , 000 .
uploadFile Request . setPartSiz e ( 1 * 1024 * 1024 );
// Start resumable upload ( that is disabled by default
// ).
uploadFile Request . setEnableC heckpoint ( true );
// Configure the file used to record the result of
// local part uploads . You need to configure this
// parameter when you enable resumable upload . This file
// stores progress informatio n . If a part upload
// fails , it can be continued based on the progress
// informatio n recorded in this file . After you have
// uploaded the file , the file is deleted . By default ,
// this file shares the same directory ( uploadFile . ucp )
// as the file you want to upload .
uploadFile Request . setCheckpo intFile ( "< yourCheckp ointFile
>");
// Configure object meta .
uploadFile Request . setObjectM etadata ( meta );
// Configure callback . Set the yourCallba ckEvent
// parameter to Callback .
```

```
uploadFile Request . setCallback ("< yourCallbackEvent >");  
  
// Start resumable upload .  
ossClient . uploadFile ( uploadFile Request );  
  
// Close your OSSClient .  
ossClient . shutdown ();
```

2.10.6 Multipart upload

This topic describes how to use multipart upload.

For the complete code of multipart upload, see [GitHub](#).

To enable multipart upload, perform the following steps:

1. Initiate a multipart upload event.

You can call `ossClient.initiateMultipartUpload` to return the globally unique `uploadId` created in OSS.

2. Upload parts.

You can call `ossClient.uploadPart` to upload part data.



Note:

- For parts with a same `uploadId`, parts are sequenced by their part numbers. If you have uploaded a part and use the same part number to upload another part, the later part will replace the former part.
- OSS places the MD5 value of part data in ETag and returns the MD5 value to the user.
- SDK automatically configures Content-MD5. OSS calculates the MD5 value of uploaded data and compares it with the MD5 value calculated by SDK. If the two values vary, the error code of `InvalidDigest` is returned.

3. Complete multipart upload.

After you have uploaded all parts, call `partossClient.completeMultipartUpload` to combine these parts into a complete object.

The following code is used as a complete example that describes the process of multipart upload:

```
// This example uses endpoint China East 1 ( Hangzhou  
// ). Specify the actual endpoint based on your  
// requirements .  
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com  
";
```

```

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
String  accessKeyI d = "< yourAccess  KeyId >";
String  accessKeyS ecret = "< yourAccess  KeySecret >";
String  bucketName = "< yourBucket  Name >";
String  objectName = "< yourObject  Name >";

// Create an OSSClient instance .
OSS  ossClient = new OSSClientB uilder (). build ( endpoint ,
accessKeyI d , accessKeyS ecret );

/* Step 1 : Initiate a multipart upload event .
*/
InitiateMu ltipartUpl oadRequest request = new InitiateMu
ltipartUpl oadRequest ( bucketName , objectName );
InitiateMu ltipartUpl oadResult result = ossClient .
initiateMu ltipartUpl oad ( request );
// An uploadId is returned . It is the unique
// identifier for a part upload event . You can initiate
// related operations ( such as part upload cancelatio n
// and query ) based on the uploadId .
String  uploadId = result . getUploadI d ();

/* Step 2 : Upload parts .
*/
// partETags is a set of PartETags . A PartETag
// consists of an ETag and a part number .
List < PartETag > partETags = new ArrayList < PartETag > ();
// Calculate the total number of parts .
final long  partSize = 1 * 1024 * 1024L ; // 1MB
final File  sampleFile = new File ("< localFile >");
long  fileLength = sampleFile . length ();
int  partCount = ( int ) ( fileLength / partSize );
if ( fileLength % partSize != 0 ) {
    partCount ++;
}
// Upload each part simultaneo usly until all parts
// are uploaded .
for ( int  i = 0 ; i < partCount ; i ++ ) {
    long  startPos = i * partSize ;
    long  curPartSiz e = ( i + 1 == partCount ) ? ( fileLength
- startPos ) : partSize ;
    InputStrea m instream = new FileInputS tream ( sampleFile
);
    // Skip parts that have been uploaded .
    instream . skip ( startPos );
    UploadPart Request uploadPart Request = new UploadPart
Request ();
    uploadPart Request . setBucketN ame ( bucketName );
    uploadPart Request . setKey ( objectName );
    uploadPart Request . setUploadI d ( uploadId );
    uploadPart Request . setInputSt ream ( instream );
    // Configure the size available for each part .
    // Aside from the last part , all parts are sized 100
    // KB at least .
    uploadPart Request . setPartSiz e ( curPartSiz e );
    // Configure part numbers . Each part is configured
    // with a part number . The value can be from 1 to

```

```

10 , 000 . If you configure a number beyond the range
, OSS returns an InvalidArgument error code .
    uploadPart Request . setPartNum ber ( i + 1 );
    // Do not upload each part sequentially . They can
    be uploaded from different OSSClients . Then they are
    sequenced and combined into a complete object based
    on part numbers .
    UploadPart Result uploadPart Result = ossClient .
uploadPart ( uploadPart Request );
    // After you upload a part , OSS returns a result
    that contains a PartETag . The PartETag is stored in
    partETags .
    partETags . add ( uploadPart Result . getPartETa g ());
}

/* Step 3 : Complete multipart upload .
*/
// Sequence parts . partETags must be sequenced by part
numbers in ascending order .
Collection s . sort ( partETags , new Comparator < PartETag >() {
    public int compare ( PartETag p1 , PartETag p2 ) {
        return p1 . getPartNum ber () - p2 . getPartNum ber ();
    }
});
// You must provide all valid PartETags when you
perform this operation . OSS verifies the validity of
all parts one by one after it receives PartETags .
After part verificati on is successful , OSS combines
these parts into a complete object .
CompleteMu ltipartUpl oadRequest completeMu ltipartUpl
oadRequest =
    new CompleteMu ltipartUpl oadRequest ( bucketName ,
objectName , uploadId , partETags );
ossClient . completeMu ltipartUpl oad ( completeMu ltipartUpl
oadRequest );

// Close your OSSClient .
ossClient . shutdown ();

```

Cancel a multipart upload event

You can call `ossClient.abortMultipartUpload` to cancel a multipart upload event.

If you cancel a multipart upload event, you are not allowed to perform any other operations with this `uploadId` anymore. The uploaded parts will be deleted.

Run the following code to cancel a multipart upload event:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .

```

```
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder().build ( endpoint ,
accessKeyId , accessKeySecret );

// Cancel multipart upload . The uploadId is from
InitiateMultipartUpload .
AbortMultipartUploadRequest abortMultipartUploadRequest =
    new AbortMultipartUploadRequest ("< yourBucket Name
>", "< yourObject Name >", "< uploadId >");
ossClient . abortMultipartUpload ( abortMultipartUploadRequest
);

// Close your OSSClient .
ossClient . shutdown ();
```

List uploaded parts

Call `ossClient.listParts` to list all uploaded parts with a specified `uploadId`.

- Simple list

Run the following code for simple list:

```
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder().build ( endpoint
, accessKeyId , accessKeySecret );

// List uploaded parts . The uploadId is returned from
InitiateMultipartUpload .
ListPartsRequest listPartsRequest = new ListPartsR
equest ("< yourBucket Name >", "< yourObject Name >", "< uploadId
>");
// Configure uploadId .
// listPartsRequest . setUploadId ( uploadId );
// Set the maximum number of parts that can be
displayed on each page to 100 . The default number
is 1 , 000 .
listPartsRequest . setMaxParts ( 100 );
// Specify the initial position in the list . Only
the parts with part numbers greater than this
parameter value are listed .
listPartsRequest . setPartNumberMarker ( 2 );
```



```

PartListing partListing = ossClient.listParts (
listPartsRequest);

for ( PartSummary part : partListing.getParts ()) {
    // Obtain part numbers .
    part . getPartNumber ();
    // Obtain the size of part data .
    part . getSize ();
    // Obtain ETag .
    part . getETag ();
    // Obtain the latest time parts are modified .
    part . getLastModified ();
}

// Close your OSSClient .
ossClient . shutdown ();

```

- **List all uploaded parts**

By default, listParts can only list a maximum of 1,000 parts at a time. To list more than 1,000 uploaded parts, run the following code:

```

// This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS . We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance . To create a RAM account , log on to https :// ram . console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint , accessKeyId , accessKeySecret );

// List all uploaded parts .
PartListing partListing ;
ListPartsRequest listPartsRequest = new ListPartsRequest ( "< yourBucket Name >", "< yourObject Name >", "< uploadId >" );

do {
    partListing = ossClient . listParts ( listPartsRequest );

    for ( PartSummary part : partListing.getParts ()) {
        // Obtain part numbers .
        part . getPartNumber ();
        // Obtain the part size .
        part . getSize ();
        // Obtain ETag .
        part . getETag ();
        // Obtain the latest time parts are modified .
        part . getLastModified ();
    }
}

```

```
// Specify the initial position for the list. Only
// the parts with part numbers greater than the
// initial value are listed.
listPartsRequest.setPartNumberMarker ( partListing .
getNextPartNumberMarker ());
} while ( partListing . isTruncated ());

// Close your OSSClient .
ossClient . shutdown ()
```

- List all uploaded parts on one or more pages

Run the following code to specify the maximum number of parts displayed on each page:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint
, accessKeyId , accessKeySecret );

// List all uploaded parts on one or more pages .
PartListing partListing ;
ListPartsRequest listPartsRequest = new ListPartsRequest
( "< yourBucket Name >", "< yourObject Name >", "< uploadId
>");
// Set the maximum number of parts displayed on
// each page to 100 .
listPartsRequest . setMaxParts ( 100 );

do {
    partListing = ossClient . listParts ( listPartsRequest );

    for ( PartSummary part : partListing . getParts ()) {
        // Obtain part numbers .
        part . getPartNumber ();
        // Obtain the part size .
        part . getSize ();
        // Gets the etag of the slice .
        // Obtain ETag .
        // Obtain the latest time parts are modified .
        part . getLastModified ();
    }

    listPartsRequest . setPartNumberMarker ( partListing .
    getNextPartNumberMarker ());
} while ( partListing . isTruncated ());
```

```
// Close your OSSClient .  
ossClient . shutdown ();
```

List all part upload events for a bucket

Call `ossClient.listMultipartUploads` to list all ongoing part upload events (events that have been initiated but not completed or have been canceled). You can configure the following parameters:

Parameter	Description	Configuration method
prefix	Specifies the prefix that must be included in the returned object name . Note that if you use a prefix for query, the returned object name will contain the prefix.	<code>ListMultipartUploadsRequest.setPrefix(String prefix)</code>
delimiter	Specifies a delimiter of a forward slash (/) used to group object names . The object between the specified prefix and the first occurrence of a delimiter of a forward slash (/) is commonPrefixes.	<code>ListMultipartUploadsRequest.setDelimiter(String delimiter)</code>
maxUploads	Specifies the maximum number of part upload events. The maximum value (also default value) you can set is 1,000.	<code>ListMultipartUploadsRequest.setMaxUploads(Integer maxUploads)</code>
keyMarker	Lists all part upload events with the object whose names start with a letter that comes after the keyMarker value in the alphabetical order. You can use this parameter with the uploadIdMarker parameter to specify the initial position for the specified returned result.	<code>ListMultipartUploadsRequest.setKeyMarker(String keyMarker)</code>

Parameter	Description	Configuration method
uploadIdMarker	You can use this parameter with the keyMarker parameter to specify the initial position for the specified returned result. If you do not configure keyMarker, the uploadIdMarker parameter is invalid. If you configure keyMarker, the query result contains: • All objects whose names start with a letter that comes after the keyMarker value. • All objects whose names start with a letter that is the same as the keyMarker value in the alphabetical order and the value of uploadId greater than that of uploadIdMarker.	ListMultipartUploads Request.setUploadIdMarker(String uploadIdMarker)

• Simple list

Run the following code to list multipart upload events:

```
// This example uses endpoint China East 1 (Hangzhou). Specify the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS. We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance. To create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";

// Create an OSSClient instance.
OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
```

```
// Lists multipart upload events . You can list 1 ,
// 000 parts by default .
ListMultipartUploadsRequest listMultipartUploadsRequest
= new ListMultipartUploadsRequest ( bucketName );
MultipartUploadListing multipartUploadListing =
ossClient . listMultipartUploads ( listMultipartUploadsRequest );

for ( MultipartUpload multipartUpload : multipartUploadListing . getMultipartUploads () ) {
    // Obtain uploadId .
    multipartUpload . getUploadId ();
    // Obtain object names .
    multipartUpload . getKey ();
    // Obtain the time part upload events are
    // initiated .
    multipartUpload . getInitiated ();
}

// Close your OSSClient .
ossClient . shutdown ();
```

If the returned result shows that the value of `isTruncated` is false, the values of `nextKeyMarker` and `nextUploadIdMarker` are returned and used as the initial position for the next object reading. If you fail to obtain all part upload events at a time, list them by pagination.

- List all part upload events

By default, `listMultipartUploads` can only list a maximum of 1,000 parts at a time.

To list more than 1,000 parts, run the following code to list all parts:

```
// This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint ,
accessKeyId , accessKeySecret );

// Lists multipart upload events .
MultipartUploadListing multipartUploadListing ;
ListMultipartUploadsRequest listMultipartUploadsRequest
= new ListMultipartUploadsRequest ( bucketName );

do {
```

```

        multipartUploadListing = ossClient.listMultip
artUploads ( listMultip artUploads Request );

        for ( MultipartUpload multipartUpload : multipartU
ploadListing.getMultipartUploads () ) {
            // Obtain uploadId .
            multipartUpload.getUploadId ();
            // Obtain object names .
            multipartUpload.getKey ();
            // Obtain the time part upload events are
initiated .
            multipartUpload.getInitiated ();
        }

        listMultip artUploads Request . setKeyMarker ( multipartU
ploadListing.getNextKey Marker ());

        listMultip artUploads Request . setUploadIdMarker (
multipartUploadListing.getNextUploadIdMarke r ());
    } while ( multipartUploadListing.isTruncated ());

    // Close your OSSClient .
    ossClient.shutdown ();

```

- List part upload events on one or more pages

Run the following code to list part upload events on one or more pages:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSS ossClient = new OSSClientBuilder (). build ( endpoint
, accessKeyId , accessKeySecret );

// Lists multipart upload events .
MultipartUploadListing multipartUploadListing ;
ListMultip artUploads Request listMultip artUploads Request
= new ListMultip artUploads Request ( bucketName );
// Configure the number of part upload events that
can be displayed on each page .
listMultip artUploads Request . setMaxUplo ads ( 50 );

do {
    multipartUploadListing = ossClient . listMultip
artUploads ( listMultip artUploads Request );

    for ( MultipartUpload multipartUpload : multipartU
ploadListing.getMultipartUploads () ) {

```

```

        // Obtain uploadId .
        multipartU pload . getUploadI d ();
        // Obtain object names .
        multipartU pload . getKey ();
        // Obtain the time part upload events are
        initiated .
        multipartU pload . getInitiat ed ();
    }

    listMultip artUploads Request . setKeyMark er ( multipartU
ploadListi ng . getNextKey Marker ());
    listMultip artUploads Request . setUploadI dMarker (
multipartU ploadListi ng . getNextUpl oadIdMarke r ());
} while ( multipartU ploadListi ng . isTruncate d ());

// Close your OSSClient .
ossClient . shutdown ();

```

2.10.7 Upload progress bars

You can use progress bars to indicate the upload or download progress.

The following code is used as an example to describe how to view progress information with `ossClient.putObject`:

```

public class PutObjectP rogressLis tener implements
ProgressLi stener {
    private long bytesWritt en = 0 ;
    private long totalBytes = - 1 ;
    private boolean succeed = false ;

    @ Override
    public void progressCh anged ( ProgressEv ent progressEv
ent ) {
        long bytes = progressEv ent . getBytes ();
        ProgressEv entType eventType = progressEv ent .
getEventTy pe ();
        switch ( eventType ) {
            case TRANSFER_S TARTED_EVE NT :
                System . out . println ( " Start to upload ....." );
                break ;
            case REQUEST_CO NTENT LENG TH_EVENT :
                this . totalBytes = bytes ;
                System . out . println ( this . totalBytes + " bytes
in total will be uploaded to OSS " );
                break ;
            case REQUEST_BY TE_TRANSFE R_EVENT :
                this . bytesWritt en += bytes ;
                if ( this . totalBytes != - 1 ) {
                    int percent = ( int )( this . bytesWritt en *
100 . 0 / this . totalBytes );
                    System . out . println ( bytes + " bytes have
been written at this time , upload progress : " + percent
+ "%(" + this . bytesWritt en + "/" + this . totalBytes + ")");
                } else {
                    System . out . println ( bytes + " bytes have
been written at this time , upload ratio : unknown " +
("(" + this . bytesWritt en + "/...)");
                }
            }
        }
    }
}

```

```

        break ;
        case TRANSFER_COMPLETED_EVENT :
            this . succeed = true ;
            System . out . println ( " Succeed to upload , " +
this . bytesWritten + " bytes have been transferred in
total " );
            break ;
        case TRANSFER_FAILED_EVENT :
            System . out . println ( " Failed to upload , " +
this . bytesWritten + " bytes have been transferred " );
            break ;
        default :
            break ;
    }
}

public boolean isSuccess () {
    return succeed ;
}

public static void main ( String [] args ) {
    // This example uses endpoint China ( Hangzhou ).
    Specify the actual endpoint based on your requirements .
    String endpoint = " http :// oss - cn - hangzhou .
aliyuncs . com " ;
    // It is highly risky to log on with
    AccessKey of an Alibaba Cloud account because the
    account has permissions on all the APIs in OSS
    . We recommend that you log on as a RAM user
    to access APIs or perform routine operations and
    maintenance . To create a RAM account , log on to
    https :// ram . console . aliyun . com .
    String accessKeyId = "< yourAccess KeyId >";
    String accessKeySecret = "< yourAccess KeySecret >";
    String bucketName = "< yourBucket Name >";
    String objectName = "< yourObject Name >";

    // Create an OSSClient instance .
    OSSClient ossClient = new OSSClient ( endpoint ,
accessKeyId , accessKeySecret );

    try {
        // Upload with a progress bar displayed .
        ossClient . putObject ( new PutObjectRequest (
bucketName , objectName , new File ( "< yourLocalFile >" ) ) .
< PutObjectRequest > withProgressListener (
new PutObjectProgressListener ( ) ) );
    } catch ( Exception e ) {
        e . printStackTrace ( );
    }
    // Close your OSSClient instance .
    ossClient . shutdown ( );
}
}

```

You can view progress information with `ossClient.putObject`, `ossClient.getObject`, or `ossClient.uploadPart`. However, you cannot view progress information with `ossClient.uploadFile` or `ossClient.downloadFile`.

For the complete code of upload progress bar, see [GitHub](#).

2.10.8 Upload callback

This topic describes how to use

For the complete code of upload callback, see [GitHub](#).

Run the following code for upload callback:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";
// Specify the IP address of the server you want to
// send the callback request to , for example , http ://
// oss - demo . aliyuncs . com : 23450 or http :// 127 . 0 . 0 . 1 :
// 9090 .
String callbackUr l = "< yourCallba ckServerUr l >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

String content = " Hello OSS ";
PutObjectR equest putObjectR equest = new PutObjectR equest
( bucketName , objectName , new ByteArrayI nputStream ( content .
getBytes ());

// Configure upload callback parameters .
Callback callback = new Callback ();
callback . setCallbac kUrl ( callbackUr l );
// Configure the value of the host field carried in
// the callback request header , such as oss - cn - hangzhou
// . aliyuncs . com .
callback . setCallbac kHost ( " oss - cn - hangzhou . aliyuncs . com
" );
// Configure the value of the body field carried in
// the callback request .
callback . setCallbac kBody ( "{\\\\" mimeType \\\":${ mimeType },\\"
size \\\":${ size } }" );
// Configure Content - Type for the callback request .
callback . setCalbac kBodyType ( CallbackBo dyType . JSON );
// Configure the custom parameters used to initiate a
// callback request . Each custom parameter consists of a
// key and a value . The key must start with x : .
callback . addCallbac kVar ( " x : var1 " , " value1 " );
callback . addCallbac kVar ( " x : var2 " , " value2 " );
putObjectR equest . setCallbac k ( callback );
```

```
PutObjectResult result = ossClient.putObject(
    putObjectRequest);

// Read the message returned from upload callback.
byte[] buffer = new byte[1024];
PutObjectResult result.getResponse().read(buffer);
// If you do not close the reader after the data
// is read, connection leaks may occur. Consequently,
// no available connections are left and an exception
// occurs.
PutObjectResult result.getResponse().close();

// Close your OSSClient instance.
ossClient.shutdown();
```

For more information, see [Upload callback](#) in OSS Developer Guide.

2.11 Download objects

2.11.1 Overview

This topic is the overview of object download in OSS Java SDK.

OSS Java SDK provides the following download methods:

- [Streaming download](#)
- [Download to your local file](#)
- [Range download](#)
- [Resumable download](#)
- [Conditional download](#)

You can view the download progress shown in the [download progress bar](#).

2.11.2 Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use streaming download. Streaming download enables you to download part of the object each time until you have downloaded the entire object.

If you do not close `OssObject` after it is used, connection leaks may occur. Consequently, no connection is available and the program cannot run properly. Run the following code to close `OssObject`:

```
OssObject ossObject = ossClient.getObject(bucketName,
    objectName);
```

```
ossObject . close ();
```

The following code is used for streaming downloads:

```
// This example uses the endpoint China East 1 (
// Hangzhou). Specify the actual endpoint based on your
// requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS. We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance. To
// create a RAM account, log on to https://ram.console
// .aliyun.com.
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// ossObject includes the bucket name , key ( object name
// ), object meta ( meta informatio n ), and InputStrea m .
OSSObject ossObject = ossClient . getObject ( bucketName ,
objectName );

// Read the object content .
System . out . println ( " Object content : " );
BufferedReader reader = new BufferedReader ( new
InputStreamReader ( ossObject . getObjectC ontent ( ) ) );
while ( true ) {
    String line = reader . readLine ( );
    if ( line == null ) break ;

    System . out . println ( "\ n " + line );
}
// If you do not close the reader after the data
// is read , connection leaks may occur . Consequent ly , no
// connection is available and the program cannot run
// properly .
reader . close ( );

// Close your OSSClient .
ossClient . shutdown ( );
```

For the complete code of streaming download, see [GitHub](#).

2.11.3 Download to your local file

This topic describes how to download an object from OSS to a local file.

You can run the following code to download a specified object from OSSClient to your local file:

```
// This example uses the endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Download an object to your local file . If the
// name of the object is the same as that of your
// local file , the object will replace the local file .
// If the name of the object is different from that
// of your local file , the object will be downloaded
// and a new file will be added to your local device
// .
ossClient . getObject ( new GetObjectR equest ( bucketName ,
objectName ) , new File ( "< yourLocalF ile >" ));

// Close your OSSClient .
ossClient . shutdown ();
```

2.11.4 Range download

If you only need part of the data in an object, you can use range downloads to download specified content.

Run the following code for range download:

```
// This example uses the endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
```

```

create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

GetObjectRequest request = new GetObjectRequest ( bucketName , objectName );
// Obtain data between the starting byte value of 0
and the maximum byte value of 1,000 inclusive ( that
is , a total of 1,001 bytes ). If the specified
value range is invalid , for example , the specified
range includes a negative number , or the specified
value is larger than the object size , the entire
object is downloaded .
request . setRange ( 0 , 1000 );

// Start range download .
OSSObject ossObject = ossClient . getObject ( request
);

// Read data .
byte [] buf = new byte [ 1024 ];
InputStream in = ossObject . getObjectContent ();
for ( int n = 0 ; n != - 1 ; ) {
    n = in . read ( buf , 0 , buf . length );
}

// If you do not close the reader after the data
is read , connection leaks may occur . Consequently , no
connections are available and an exception occurs .
in . close ();

// Close your OSSClient .
ossClient . shutdown ();

```

All data cannot be read at one time when it is streamed. To stream a 64 KB block of data, use the following code to read the data multiple times until the entire object is downloaded. For more information, see [InputStream.read](#).

```

byte [] buf = new byte [ 1024 ];
InputStream in = ossObject . getObjectContent ();
for ( int n = 0 ; n != - 1 ; ) {
    n = in . read ( buf , 0 , buf . length );
}
in . close ();

```

2.11.5 Resumable download

Large-sized objects may fail to be downloaded because the network is unstable or the Java program exits abnormally and the entire object needs to be downloaded

again. However, the object may still fail to be downloaded after multiple attempts. Therefore, OSS provides resumable download.

To enable resumable download, you need to split the object you want to download into multiple parts and download them separately. After you have downloaded all parts, these parts will be combined into a complete object.

You can use `ossClient.downloadFile` to enable resumable download. `DownloadFileRequest` contains the following parameters:

Parameter	Description	Required or optional	Default value	Configuration methods
<code>bucket</code>	Specifies the name of a bucket.	Required	None	Constructor
<code>key</code>	Specifies the name of an object.	Required	None	Constructor
<code>downloadFile</code>	Specifies a local file. You can download an object to the local file.	Optional	Name of the object	Constructor or <code>setDownloadFile</code>
<code>partSize</code>	Specifies the size of a part. The value can be between 1 byte to 5 GB.	Optional	Size of an object/10000	<code>setPartSize</code>
<code>taskNum</code>	Specifies the number of parts you need to download simultaneously.	Optional	1	<code>setTaskNum</code>
<code>enableCheckpoint</code>	Specifies whether resumable download is enabled.	Optional	Disable	<code>setEnabledCheckpoint</code>

Parameter	Description	Required or optional	Default value	Configuration methods
checkpointFile	Specifies the file to record the results of partial downloads. You need to configure this parameter to enable resumable download. This file stores information about the download progress. If you fail to download a part, the download will continue based on the recorded progress. After the object is downloaded to your local device, this file will be deleted.	Optional	downloadFile.ucp (share the same directory with DownloadFile)	setCheckpointFile

Run the following code for resumable download:

```
// This example uses the endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
```

```

String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Send a request to download 10 tasks simultaneously
and start resumable download .
DownloadFileRequest downloadFileRequest = new DownloadFile
Request ( bucketName , objectName );
downloadFileRequest . setDownloadFile ( "< yourDownloadFile >");
downloadFileRequest . setPartSize ( 1 * 1024 * 1024 );
downloadFileRequest . setTaskNum ( 10 );
downloadFileRequest . setEnableCheckpoint ( true );
downloadFileRequest . setCheckpointFile ( "< yourCheckpointFile
>");

// Download the object to your local file .
DownloadFileResult downloadResult = ossClient . downloadFile (
downloadFileRequest );
// After the download succeeds , object meta is returned
.
downloadResult . getObjectMetadata ();

// Close your OSSClient .
ossClient . shutdown ();

```

2.11.6 Conditional download

You can configure one or more conditions for downloads. If the specified conditions are met, the object can be downloaded. If the specified conditions are not met, an error code is returned and the object cannot be downloaded. You can configure the following conditions:

Parameter	Description
If-Modified-Since	If the specified time is earlier than the time the object is modified, the object can be downloaded. Otherwise, an error (304 Not modified) is returned.
If-Unmodified-Since	If the specified time is later than or equal to the time the object is modified, the object can be downloaded. Otherwise , an error (412 Precondition failed) is returned.
If-Match	If the specified ETag matches that of an object, the object can be downloaded. Otherwise, an error (412 Precondition failed) is returned.

Parameter	Description
If-None-Match	If the specified ETag does not match that of an object, the object can be downloaded. Otherwise, an error (304 Not modified) is returned.

Both If-Modified-Since and If-Unmodified-Since can exist at the same time as object download conditions. Both If-Match and If-None-Match can exist at the same time as object download conditions.

`ossClient.getObjectMeta` can be used to obtain the ETag.

Run the following code for conditional download:

```
// This example uses the endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

GetObjectRequest request = new GetObjectRequest (
bucketName , objectName );
// Configure conditions .
request . setModifiedSinceConstraint ( new Date ());

// Download an object to a local file .
ossClient . getObject ( request , new File ("< yourLocalFile
>"));

// Close your OSSClient .
ossClient . shutdown ();
```

Conditional download supports `ossClient.getObject` and `ossClient.downloadFile`.

2.11.7 Download progress bars

You can use progress bars to indicate the upload or download progress.

The following code is used as an example to describe how to view progress information with `ossClient.getObject`:

```
static class GetObjectProgressListener implements
ProgressListener {
    private long bytesRead = 0 ;
    private long totalBytes = - 1 ;
    private boolean succeed = false ;
    @Override
    public void progressChanged ( ProgressEvent
progressEvent ) {
        long bytes = progressEvent . getBytes ();
        ProgressEventType eventType = progressEvent .
getEventType ();
        switch ( eventType ) {
            case TRANSFER_STARTED_EVENT :
                System . out . println ( " Start to download
.....");
                break ;
            case RESPONSE_CONTENT_LENGTH_EVENT :
                this . totalBytes = bytes ;
                System . out . println ( this . totalBytes + "
bytes in total will be downloaded to a local file
");
                break ;
            case RESPONSE_BYTE_TRANSFER_EVENT :
                this . bytesRead += bytes ;
                if ( this . totalBytes != - 1 ) {
                    int percent = ( int )( this . bytesRead *
100 . 0 / this . totalBytes );
                    System . out . println ( bytes + " bytes have
been read at this time , download progress : " +
percent + "%(" + this . bytesRead +
"/" + this . totalBytes + ")");
                } else {
                    System . out . println ( bytes + " bytes have
been read at this time , download ratio : unknown " +
 "(" + this . bytesRead + "/...");
                }
                break ;
            case TRANSFER_COMPLETED_EVENT :
                this . succeed = true ;
                System . out . println ( " Succeed to download , "
+ this . bytesRead + " bytes have been transferred in
total ");
                break ;
            case TRANSFER_FAILED_EVENT :
                System . out . println ( " Failed to download , "
+ this . bytesRead + " bytes have been transferred ");
                break ;
            default :
                break ;
        }
    }
    public boolean isSuccess () {
        return succeed ;
    }
}
```

```

public static void main ( String [] args ) {
    // This example uses endpoint China ( Hangzhou ).
    Specify the actual endpoint based on your requirements .
    String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
    com ";
    // It is highly risky to log on with AccessKey
    of an Alibaba Cloud account because the account has
    permission s on all the APIs in OSS . We recommend
    that you log on as a RAM user to access APIs
    or perform routine operations and maintenance . To
    create a RAM account , log on to https :// ram . console
    . aliyun . com .
    String accessKeyId = "< yourAccess KeyId >";
    String accessKeySecret = "< yourAccess KeySecret >";
    String bucketName = "< yourBucket Name >";
    String objectName = "< yourObject Name >";
    OSS ossClient = new OSSClientBuilder (). build ( endpoint
    , accessKeyId , accessKeySecret );
    try {
        // Download with a progress bar displayed .
        ossClient . getObject ( new GetObjectRequest (
        bucketName , objectName ) .
            < GetObjectRequest > withProgressListener (
        new GetObjectProgressListener () ,
            new File ( "< yourLocalFile >" ) ) );
    } catch ( Exception e ) {
        e . printStackTrace ();
    }
    // Close your OSSClient instance .
    ossClient . shutdown ();
}

```

You can view progress information with `ossClient.putObject`, `ossClient.getObject`, or `ossClient.uploadPart`. However, you cannot view progress information with `ossClient.uploadFile` or `ossClient.downloadFile`.

For the complete code of download progress bar, see [GitHub](#).

2.12 Manage objects

2.12.1 Overview

This topic is the overview of object management in OSS Java SDK.

You can perform the following operations on objects in a bucket with APIs:

- [Determine whether an object exists](#)
- [Manage ACL for an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Delete an object](#)

- [Copy an object](#)
- [Restore an archive object](#)
- [Manage a symbolic link](#)

2.12.2 Determine whether a specified object exists

This topic describes how to determine whether a specified object exists

Run the following code to determine whether a specified object exists:

```
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements.
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Determine whether the specified object exists . The
doesObject Exist includes a parameter : isOnlyInOS S . If
the value of this parameter is true , 302 Redirect
or OSS mirroring is ignored . Otherwise , 302 Redirect
or OSS mirroring is considered .

boolean found = ossClient . doesObject Exist ( "< yourBucket
Name >", "< yourObject Name >");
System . out . println ( found );

// Close your OSSClient .
ossClient . shutdown ();
```

2.12.3 Manage ACL for an object

This topic describes how to manage ACL for an object.

The following table describes the permissions included in the Access Control List (ACL) for an object.

Permission	Description	Value
default	The ACL of an object is the same with that of its bucket.	CannedAccessControlList.Default

Permission	Description	Value
Private	Only the object owner and authorized users can read and write the object.	CannedAccessControlList.Private
Public read	Only the object owner and authorized users can read and write the object. Other users can only read the object. Authorize this permission with caution.	CannedAccessControlList.PublicRead
Public read-write	All users can read and write the object. Authorize this permission with caution.	CannedAccessControlList.PublicReadWrite

The ACL of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

Configure an ACL for an object

You can run the following code to configure an ACL for an object:

```
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Configure the ACL for the object to public read .
ossClient . setObjectA cl ("< yourBucket Name >", "< yourObject
Name >", CannedAcce ssControll ist . PublicRead );

// Close your OSSClient .
```

```
ossClient . shutdown ();
```

Obtain the ACL for an object

You can run the following code to obtain the ACL for an object:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Obtain the ACL for an object .
ObjectAcl objectAcl = ossClient . getObjectA cl ("< yourBucket
Name >", "< yourObject Name >");
System . out . println ( objectAcl . getPermiss ion (). toString
());

// Close your OSSClient .
ossClient . shutdown ();
```

2.12.4 Manage Object Meta

Object Meta includes HTTP headers and user-defined Object Meta. For more information, see [Object Meta](#) in OSS Developer Guide.

Configure HTTP headers

- Configure HTTP headers

Run the following code to configure HTTP headers:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
```

```

String content = " Hello OSS ";

// Create Object Meta . You can configure HTTP headers
// with Object Meta .
ObjectMeta data meta = new ObjectMeta data ();

String md5 = BinaryUtil . toBase64St ring ( BinaryUtil .
calculateM d5 ( content . getBytes ());
// Start object content verificati on with MD5 values
. After the verificati on is started , the MD5 value
calculated by OSS is compared with that of an
object . If the two values are different , an error
occurs .
meta . setContent MD5 ( md5 );
// Specify the type of content you want to upload
. The content type determines the form and format
in which your browser reads an object . The object
extension is used as the content type if the
content type is not specified . If no object
extension exists , the default value applicatio n / octet
- stream is used .
meta . setContent Type ( " text / plain " );
// Specify the name of the content when you
download an object .
meta . setContent Dispositio n ( " attachment ; filename = \"
DownloadFi lename \"" );
// Configure the length for an object . If the
actual object length is greater than the configured
length , only the configured length will be uploaded
. If the configured length is greater than the
actual object length , the entire object will be
uploaded .
meta . setContent Length ( content . length ());
// Configure the cache action of a webpage when an
object is downloaded .
meta . setCacheCo ntrol ( " Download Action " );
// Configure the expiration time ( GMT is used ).
meta . setExpirat ionTime ( DateUtil . parseIso86 01Date ( " 2022
- 10 - 12T00 : 00 : 00 . 000Z " ));
// Configure the encoding format when an object is
downloaded .
meta . setContent Encoding ( " utf - 8 " );
// Configure the HTTP header .
meta . setHeader ( "< yourHeader >", "< yourHeader Value >");
// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Upload the object .
ossClient . putObject ( "< yourBucket Name >", "< yourObject Name
>", new ByteArrayI nputStream ( content . getBytes ()), meta
);

// Close your OSSClient .
ossClient . shutdown ();

```

For more information about HTTP headers, see [RFC2616](#).

- **Configure user-defined Object Meta**

You can define user-defined Object Meta for object descriptions.

Run the following code to configure user-defined Object Meta:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

String content = " Hello OSS ";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Create Object Meta .
ObjectMeta data meta = new ObjectMeta data ();

// Set property to property - value for user - defined
// Object Meta . We recommend that you use Base64
// encoding method .
meta . addUserMet adata ( " property ", " property - value " );

// Upload the object .
ossClient . putObject ( "< yourBucket Name >", "< yourObject Name
>", new ByteArrayI nputStream ( content . getBytes ()), meta
);

// Obtain Object Meta .
ossClient . getObjectM etadata ( "< yourBucket Name >", "<
yourObject Name >");

// Close your OSSClient .
ossClient . shutdown ();
```

When an object is downloaded, its Object Meta is also downloaded. An object can contain multiple pieces of Object Meta, with the total maximum size smaller than 8 KB.

Modify Object Meta

Run the following code to modify Object Meta:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
```



```

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https://ram.console
// .aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String sourceBucketName = "< yourSource BucketName >";
String sourceObjectName = "< yourSource ObjectName >";
String destinationBucketName = "< yourDestinationBucketName >";
String destinationObjectName = "< yourDestinationObjectName >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Replace the source object with the target object .
// Call ossClient . copyObject to modify Object Meta .
CopyObjectRequest request = new CopyObjectRequest (
sourceBucketName , sourceObjectName , destinationBucketName
, destinationObjectName );

ObjectMetadata meta = new ObjectMetadata ();
// Specify the type of content you want to upload .
// The content type determines the form and format that
// your browser uses to read an object . The object
// extension is used for the content type if the
// content type is not specified . If no object extension
// exists , the default value application / octet - stream
// is used .
meta . setContentType ( " text / plain " );
// Specify the name of the content when you download
// an object .
meta . setContentDisposition ( " Download File Name " );
// Configure the cache action of a webpage when an
// object is downloaded .
meta . setCacheControl ( " Download Action " );
// Configure the expiration time ( GMT is used ) .
meta . setExpirationTime ( DateUtil . parseIso8601Date ( " 2022 -
10 - 12T00 : 00 : 00 . 000Z " ));
// Configure the encoding type when the object is
// downloaded .
meta . setContentEncoding ( " utf - 8 " );
// Configure the HTTP header .
meta . setHeader ( "< yourHeader > " , "< yourHeader Value > " );
// Set property to property - value for user - defined
// Object Meta .
meta . addUserMetadata ( " property " , " property - value " );
request . setNewObjectMetadata ( meta );

// Modify Object Meta .
ossClient . copyObject ( request );

// Close your OSSClient .

```

```
ossClient . shutdown ();
```

Obtain Object Meta

The following table lists the methods you can use to obtain Object Meta.

Method	Description	Advantage
<code>ossClient.getSimplifiedObjectMeta</code>	Obtains ETag, Size, and LastModified (the latest time an object is modified.) for the object.	More lightweight and faster
<code>ossClient.getObjectMetadata</code>	Obtains all metadata.	None

Run the following code to obtain Object Meta:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

// Obtain partial Object Meta .
Simplified ObjectMeta objectMeta = ossClient . getSimplif
iedObjectM eta ("< yourBucket Name >", "< yourObject Name >");
System . out . println ( objectMeta . getSize ());
System . out . println ( objectMeta . getETag ());
System . out . println ( objectMeta . getLastMod ified ());

// Obtain all Object Meta .
ObjectMeta data metadata = ossClient . getObjectM etadata ("<
yourBucket Name >", "< yourObject Name >");
System . out . println ( metadata . getContent Type ());
System . out . println ( metadata . getLastMod ified ());
System . out . println ( metadata . getExpirat ionTime ());

// Close your OSSClient .
```

```
ossClient . shutdown ();
```

2.12.5 List objects

This topic describes how to list objects.

Objects are listed alphabetically. You can use `ossClient.listObjects` to list objects in a bucket. The following parameters can be used for `ListObjects`:

- `ObjectListing listObjects(String bucketName)`: lists objects in a bucket. A maximum of 100 objects can be listed.
- `ObjectListing listObjects(String bucketName, String prefix)`: lists objects with the specified prefix in a bucket. A maximum of 100 objects can be listed.
- `ObjectListing listObjects(ListObjectsRequest listObjectsRequest)`: provides multiple filtering functions to flexibly query objects.

The following table describes parameters for `ObjectListing`.

Parameter	Description	Configuration method
<code>objectSummaries</code>	Specifies the returned Object Meta.	<code>List<OSSObjectSummary> getObjectSummaries()</code>
<code>prefix</code>	Specifies the prefix you configure to list required objects.	<code>String getPrefix()</code>
<code>delimiter</code>	Specifies a delimiter used to group objects.	<code>String getDelimiter()</code>
<code>marker</code>	Specifies the initial object in the list.	<code>String getMarker()</code>
<code>maxKeys</code>	Specifies the maximum number of objects that can be listed.	<code>int getMaxKeys()</code>
<code>nextMarker</code>	Specifies the initial position for the next object.	<code>String getNextMarker()</code>

Parameter	Description	Configuration method
isTruncated	Specifies whether the listed objects are truncated. · If the objects are all listed without truncation, a false value is returned. · If the listed objects are truncated, a true value is returned.	boolean isTruncated()
commonPrefixes	Specifies a set of objects whose names share a prefix and end with a forward slash (/) delimiter.	List<String> getCommonPrefixes()
encodingType	Specifies the encoding type used in the response.	String getEncodingType()

Simple list

Run the following code to list objects in a specified bucket. A maximum of 100 objects can be listed by default.

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String KeyPrefix = "< yourKeyPre fix >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// List objects . If you do not configure KeyPrefix
// , all objects in the bucket are listed . If you
// configure KeyPrefix , objects with a specified prefix in
// the bucket are listed .
ObjectListing objectListing = ossClient . listObject s (
bucketName , KeyPrefix );
```

```

List < OSSObjectSummary > sums = objectListing.getObjectSummaries();
for ( OSSObjectSummary s : sums ) {
    System.out.println (" \t " + s.getKey());
}

// Close your OSSClient.
ossClient.shutdown();

```

List objects with ListObjectsRequest

You can configure parameters for ListObjectsRequest to flexibly query objects. The following table describes parameters for ListObjectsRequest.

Parameter	Description	Configuration method
prefix	Specifies the prefix that must be included in the returned objects.	setPrefix(String prefix)
delimiter	Specifies a delimiter of a forward slash (/) used to group objects based on their names. The substring between the specified prefix and the first occurrence of the delimiter of a forward slash (/) is commonPrefixes.	setDelimiter(String delimiter)
marker	Lists the objects after the value of marker.	setMarker(String marker)
maxKeys	Specifies the maximum number of objects that can be listed. The default value is 100. The maximum value is 1,000.	setMaxKeys(Integer maxKeys)
encodingType	Specifies the encoding type of an object name in the response body. Only the URL encoding type is supported.	setEncodingType(String encodingType)

- List a specified number of objects

Run the following code to list a specified number of objects:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Configure the maximum number .
final int maxKeys = 200 ;
// List objects .
ObjectList ing objectList ing = ossClient . listObject s (
new ListObject sRequest ( bucketName ). withMaxKey s ( maxKeys
));
List < OSSObjectS ummary > sums = objectList ing . getObjectS
ummaries ();
for ( OSSObjectS ummary s : sums ) {
    System . out . println ( "\ t " + s . getKey ());
}

// Close your OSSClient .
ossClient . shutdown ();
```

- List objects by a specified prefix

Run the following code to list objects with a specified prefix:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
```

```

OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// Specify a prefix .
final String keyPrefix = "< yourkeyPrefix >";

// List objects with a specified prefix . By default , a maximum of 100 objects can be listed .
ObjectListing objectListing = ossClient . listObjects ( new ListObjectsRequest ( bucketName ). withPrefix ( keyPrefix ) );
List< OSSObjectSummary > sums = objectListing . getObjectSummaries ();
for ( OSSObjectSummary s : sums ) {
    System . out . println ( "\t " + s . getKey () );
}

// Close your OSSClient .
ossClient . shutdown ();

```

- List objects by marker

The marker parameter indicates the name of an object from which the listing begins. Run the following code to specify an object (specified with marker) after which the listing begins:

```

// This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS . We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance . To create a RAM account , log on to https :// ram . console . aliyun . com .
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";
String bucketName = "< yourBucketName >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );
final String marker = "< yourMarker >";

// List objects placed after the object ( specified with marker ). By default , a maximum of 100 objects can be listed .
ObjectListing objectListing = ossClient . listObjects ( new ListObjectsRequest ( bucketName ). withMarker ( marker ) );
List< OSSObjectSummary > sums = objectListing . getObjectSummaries ();
for ( OSSObjectSummary s : sums ) {
    System . out . println ( "\t " + s . getKey () );
}

// Close your OSSClient .
ossClient . shutdown ();

```

- List all objects on one or more pages

Use the following code to list all objects on one or more pages. The maximum number of objects listed on each page is specified with `maxKeys`.

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

final int maxKeys = 200 ;
String nextMarker = null ;
ObjectList ing objectList ing ;

do {
    objectList ing = ossClient . listObject s ( new
    ListObject sRequest ( bucketName ). withMarker ( nextMarker ).
    withMaxKey s ( maxKeys ));

    List < OSSObjectS ummary > sums = objectList ing .
    getObjectS ummaries ();
    for ( OSSObjectS ummary s : sums ) {
        System . out . println ( "\ t " + s . getKey ());
    }

    nextMarker = objectList ing . getNextMar ker ();
} while ( objectList ing . isTruncate d ());

// Close your OSSClient .
ossClient . shutdown ();
```

- List objects by a specified prefix on one or more pages

Run the following code to list objects by a specified prefix on one or more pages:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
```



```

permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// A maximum of 200 objects can be listed on each
page .
final int maxKeys = 200 ;
final String keyPrefix = "< yourkeyPre fix >";
String nextMarker = "< yourNextMa rker >";
ObjectListing objectListing ;

do {
    objectListing = ossClient . listObject s ( new
ListObject sRequest ( bucketName ).
        withPrefix ( keyPrefix ). withMarker ( nextMarker ).
withMaxKey s ( maxKeys ));

    List < OSSObjectS ummary > sums = objectListing .
getObjectS ummaries ();
    for ( OSSObjectS ummary s : sums ) {
        System . out . println ( "\ t " + s . getKey ());
    }

    nextMarker = objectListing . getNextMar ker ();
} while ( objectListing . isTruncate d ());

// Close your OSSClient .
ossClient . shutdown ();

```

- Specify the name of an object for encoding.

You must encode the name of an object if the name contains the following special characters. OSS supports encoding with URL.

- Single quotation marks (')
- Double quotations marks (")
- ampersands (&)
- angle brackets (< >)
- Pause marker (\ ,)
- Chinese characters

Run the following code to encode the name of a specified object:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .

```

```

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS. We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance. To
// create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

final int maxKeys = 200 ;
final String keyPrefix = "< yourkeyPrefix >";
String nextMarker = "< yourNextMarker >";
ObjectListing objectListing ;

do {
    ListObjectsRequest listObjectsRequest = new
    ListObjectsRequest ( bucketName );
    listObjectsRequest . setPrefix ( keyPrefix );
    listObjectsRequest . setMaxKeys ( maxKeys );
    listObjectsRequest . setMarker ( nextMarker );

    // Specify the name of an object for encoding .
    listObjectsRequest . setEncodingType ( "url" );

    objectListing = ossClient . listObjects ( listObjectsRequest );

    // Decode the object .
    for ( OSSObjectSummary objectSummary : objectListing . getObjectSummaries () ) {
        System . out . println ( " Key : " + URLDecoder . decode ( objectSummary . getKey () , " UTF - 8 " ) );
    }

    // Decode the commonPrefixes parameter .
    for ( String commonPrefixes : objectListing . getCommonPrefixes () ) {
        System . out . println ( " CommonPrefixes : " + URLDecoder . decode ( commonPrefixes , " UTF - 8 " ) );
    }

    // Decode the nextMarker parameter .
    if ( objectListing . getNextMarker () != null ) {
        nextMarker = URLDecoder . decode ( objectListing . getNextMarker () , " UTF - 8 " );
    }
} while ( objectListing . isTruncated () );

// Close your OSSClient .
ossClient . shutdown ();

```

Folder simulation

OSS does not use folders. All elements are stored as objects. To simulate a folder on the OSS console, you actually create a 0 MB object whose name ends with a forward slash (/). This object can be uploaded and downloaded. By default, the OSS console displays an object whose name ends with a forward slash (/) as a folder.

The delimiter and prefix parameters can be used to simulate folder functions.

- **Prefix:** specifies the prefix as the name of a folder. The folder is used to list all files (in the folder) and subfolders (directories in the folder) that start with this prefix. These files and subfolders are included in the Objects list.
- **Delimiter:** If the delimiter is specified as a forward slash (/), only the files and subfolders (directories) are displayed. Subfolders (directories) are included in the CommonPrefixes list, while the files and folders in subfolders are not displayed.

For more information about folders, see [Folder simulation](#). For the complete code of creating an object, see [GitHub](#).

Assume that the following objects are stored in a bucket: oss.jpg, fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. Forward slashes (/) are used as delimiters for folders. The subsequent examples show how to simulate folder functions.

- List all objects in a bucket

Run the following code to list all objects in a bucket:

```
// This example uses endpoint China East 1 (Hangzhou). Specify the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS. We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance. To create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// Create ListObjectsRequest.
```

```

ListObject sRequest listObject sRequest = new ListObject
sRequest ( bucketName );

// List objects .
ObjectList ing listing = ossClient . listObject s (
listObject sRequest );

// Print all objects .
System . out . println ( " Objects :");
for ( OSSObjectS ummary objectSumm ary : listing .
getObjectS ummaries () ) {
    System . out . println ( objectSumm ary . getKey ());
}

// Print all commonPref ixes .
System . out . println ( " CommonPref ixes :");
for ( String commonPref ix : listing . getCommonP refixes
()) {
    System . out . println ( commonPref ix );
}

// Disable the OSSClient .
// Close your OSSClient .

```

The returned results are as follows:

```

Objects :
fun / movie / 001 . avi
fun / movie / 007 . avi
fun / test . jpg
oss . jpg

CommonPref ixes :

```

- List all objects in a specified folder

Run the following code to list all objects in a specified folder:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Create ListObject sRequest .

```

```

ListObject sRequest listObject sRequest = new ListObject
sRequest ( bucketName );
// Configure the Prefix parameter to obtain all
objects in the fun folder .
listObject sRequest . setPrefix ( " fun /");

// Recursively list all objects in the fun folder .
ObjectListing listing = ossClient . listObject s (
listObject sRequest );

// Print all objects .
System . out . println ( " Objects :");
for ( OSSObjectSummary objectSummary : listing .
getObjectSummaries () ) {
    System . out . println ( objectSummary . getKey ());
}

// Print all commonPrefixes .
System . out . println ( "\ nCommonPrefixes :");
for ( String commonPrefix : listing . getCommonPrefixes
()) {
    System . out . println ( commonPrefix );
}

// Close your OSSClient .
ossClient . shutdown ();

```

The returned results are as follows:

```

Objects :
fun / movie / 001 . avi
fun / movie / 007 . avi
fun / test . jpg

CommonPrefixes :

```

- List objects and subfolders in a folder

Use the following code to list objects and subfolders in a folder:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId ,
accessKeySecret );

```

```
// Create ListObject sRequest .
ListObject sRequest listObject sRequest = new ListObject
sRequest ( bucketName );

// Set forward slashes (/) as the delimiter for
folders .
listObject sRequest . setDelimit er ("/");

// List all objects and folders in the fun folder .
listObject sRequest . setPrefix (" fun /");

ObjectList ing listing = ossClient . listObject s (
listObject sRequest );

// Print all objects .
System . out . println (" Objects :");
// List objects in the fun folder in a objectSumm
aries list .
for ( OSSObjectS ummary objectSumm ary : listing .
getObjectS ummaries ()) {
    System . out . println ( objectSumm ary . getKey ());
}

// Print all commonPref ixes .
System . out . println ("\ nCommonPre fixes :");
// commonPref ixes lists all subfolders in the fun
folder . The fun / movie / 001 . avi and fun / movie / 007
. avi objects are not listed , because they are
objects in the movie folder ( a subfolder of the
fun folder ).
for ( String commonPref ix : listing . getCommonP refixes
()) {
    System . out . println ( commonPref ix );
}

// Close your OSSClient .
ossClient . shutdown ();
```

The returned results are as follows:

```
Objects :
fun / test . jpg

CommonPref ixes :
fun / movie /
```

2.12.6 Delete objects



Warning:

Delete objects with caution because deleted objects cannot be recovered.

Delete an object

Run the following code to delete a single object:

```
// This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS . We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance . To create a RAM account , log on to https :// ram . console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// Delete an object .
ossClient . deleteObject ( bucketName , objectName );

// Close your OSSClient .
ossClient . shutdown ();
```

Delete multiple objects

You can delete a maximum of 1,000 objects simultaneously. Objects can be deleted in two modes: detail (verbose) and simple (quiet) modes.

- **verbose:** returns the list of objects that you have deleted successfully. The default mode is verbose.
- **quiet:** returns the list of objects that you failed to delete.

The following table describes the parameters for DeleteObjectsRequest.

Parameter	Description	Configuration method
Keys	Specifies the objects you want to delete.	setKeys(List<String>)
quiet	Specifies a mode that the deletion result uses. True indicates quiet. False indicates verbose. The default mode is verbose.	setQuiet(boolean)

Parameter	Description	Configuration method
encodingType	Encodes the name of the returned objects. Only URL is supported.	setEncodingType(String)

The following table describes the parameters for DeleteObjectsResult.

Parameter	Description	Configuration method
deletedObjects	Specifies the deletion result. verbose indicates the list of objects you have deleted successfully. quiet indicates the list of objects you failed to delete.	List<String> getDeletedObjects()
encodingType	Encodes the name of objects in deletedObjects.	getEncodingType()

Run the following code to delete multiple objects simultaneously:

```
// This example uses endpoint China East 1 (Hangzhou). Specify the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS. We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance. To create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// Delete objects.
List<String> keys = new ArrayList<String>();
keys.add("key0");
keys.add("key1");
keys.add("key2");

DeleteObjectsResult deleteObjectsResult = ossClient.deleteObjects(new DeleteObjectsRequest(bucketName).withKeys(keys));
List<String> deletedObjects = deleteObjectsResult.getDeletedObjects();

// Close your OSSClient.
```



```
ossClient . shutdown ();
```

For the complete code of deleting multiple objects simultaneously, see [GitHub](#).

2.12.7 Copy objects

Copy a small-sized object

You can use `ossClient.copyObject` to copy an object smaller than 1 GB from a bucket (source bucket) to another bucket (target bucket) in the same region. `ossClient.copyObject` supports the following configuration methods.

Configuration method	Description
<code>CopyObjectResult copyObject(String sourceBucketName, String sourceKey, String destinationBucketName, String destinationKey)</code>	Allows you to specify a source bucket, source object, target bucket, and target object. After the copy operation is completed, the target object content and its Object Meta are the same as those of the source object. This is called simple copy.
<code>CopyObjectResult copyObject(CopyObjectRequest copyObjectRequest)</code>	Allows you to specify target Object Meta and copy conditions. If the URL of the source bucket is the same as that of the target object, the Object Meta of the source object is replaced with that of the target object.

The following table describes the parameters you can configure for `CopyObjectRequest`.

Parameter	Description	Configuration method
<code>sourceBucketName</code>	Specifies the name of the source bucket.	<code>setSourceBucketName(String sourceBucketName)</code>
<code>sourceKey</code>	Specifies the name of a source object.	<code>setSourceKey(String sourceKey)</code>
<code>destinationBucketName</code>	Specifies the name of a target bucket.	<code>setDestinationBucketName(String destinationBucketName)</code>
<code>destinationKey</code>	Specifies the name of a target object.	<code>setDestinationKey(String destinationKey)</code>

Parameter	Description	Configuration method
newObjectMetadata	Specifies the target Object Meta.	<code>setNewObjectMetadata (ObjectMetadata newObjectMetadata)</code>
matchingETagConstraints	Restrictions on copies. Specifies the conditions for object copy. If ETag of a source object is the same as what OSS calculates, the source object is copied . Otherwise, an error is returned.	<code>setMatchingETagConstraints(List<String> matchingETagConstraints)</code>
nonmatchingEtagConstraints	Specifies the conditions for object copy. If ETag of a source object is different from the ETag the user provides, the source object is copied. Otherwise, an error is returned.	<code>setNonmatchingETagConstraints(List<String> nonmatchingEtagConstraints)</code>
unmodifiedSinceConstraint	Specifies the conditions for object copy. If the specified time is later than or equal to the time an object is modified, the source object is copied . Otherwise, an error is returned.	<code>setUnmodifiedSinceConstraint(Date unmodifiedSinceConstraint)</code>
modifiedSinceConstraint	Specifies the conditions for object copy. If a source object is modified after the specified time, the source object is copied. Otherwise , an error is returned.	<code>setModifiedSinceConstraint(Date modifiedSinceConstraint)</code>

The following table describes the parameters you can configure for CopyObject Request.

Parameter	Description	Configuration method
ETag	Specifies the unique identifier of an object.	<code>String getETag()</code>

Parameter	Description	Configuration method
Lastmodified	Specifies the latest time an object is modified.	Date getLastModified()

**Note:**

Object copy rules are as follows:

- Users have permissions to read and write the source object.
 - Data cannot be copied across regions. For example, you cannot copy an object in a bucket from China East1 (Hangzhou) to China North1 (Qingdao).
 - The object size is limited to a maximum of 1 GB.
- Simple copy

Run the following code for simple copy:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

String sourceBuck etName = "< yourSource BucketName >";
String sourceObje ctName = "< yourSource ObjectName >";
String destinatio nBucketNam e = "< yourDestin ationBucke
tName >";
String destinatio nObjectNam e = "< yourDestin ationObjec
tName >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Copy an object .
CopyObject Result result = ossClient . copyObject (
sourceBuck etName , sourceObje ctName , destinatio nBucketNam
e , destinatio nObjectNam e );
System . out . println ( " ETag : " + result . getETag () + "
LastModifi ed : " + result . getLastMod ified () );

// Close your OSSClient .
```

```
ossClient.shutdown();
```

- **CopyObjectRequest-based copy**

Run the following code for CopyObjectRequest-based copy:

```
// This example uses endpoint China East 1 (Hangzhou). Specify the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all the APIs in OSS. We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance. To create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";

String sourceBucketName = "< yourSourceBucketName >";
String sourceObjectName = "< yourSourceObjectName >";
String destinationBucketName = "< yourDestinationBucketName >";
String destinationObjectName = "< yourDestinationObjectName >";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// Create a CopyObjectRequest object.
CopyObjectRequest copyObjectRequest = new CopyObjectRequest(sourceBucketName, sourceObjectName, destinationBucketName, destinationObjectName);

// Configure Object Meta for a new object.
ObjectMetadata meta = new ObjectMetadata();
meta.setContentType("text/html");
copyObjectRequest.setNewObjectMetadata(meta);

// Copy the object.
CopyObjectResult result = ossClient.copyObject(copyObjectRequest);
System.out.println("ETag: " + result.getETag() + " LastModified: " + result.getLastModified());

// Close your OSSClient.
ossClient.shutdown();
```

Copy a large-sized object

To copy an object larger than 1 GB, you need to use UploadPartCopy. In other words, you need to split the object into multiple parts and copy each of them simultaneously.

To enable UploadPartCopy, perform the following steps:

1. Use `ossClient.initiateMultipartUpload` to initiate an UploadPartCopy task.

2. Start UploadPartCopy with `ossClient.uploadPartCopy`. Aside from the last part, all parts must be larger than 100 KB.
3. Submit the UploadPartCopy task with `ossClient.completeMultipartUpload`.

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

String sourceBuck etName = "< yourSource BucketName >";
String sourceObje ctName = "< yourSource ObjectName >";
String destinatio nBucketNam e = "< yourDestin ationBucke
tName >";
String destinatio nObjectNam e = "< yourDestin ationObjec
tName >";

ObjectMeta data objectMeta data = ossClient . getObjectM
etadate ( sourceBuck etName , sourceObje ctName );
// Obtain the size of the object you want to copy .
long contentLen gth = objectMeta data . getContent Length ();

// Set the minimum size to 10 MB for each part .
long partSize = 1024 * 1024 * 10 ;

// Calculate the total number of parts .
int partCount = ( int ) ( contentLen gth / partSize );
if ( contentLen gth % partSize != 0 ) {
    partCount ++;
}
System . out . println ( " total part count : " + partCount );

// Initiate an UploadPart Copy task . You can specify
// Object Meta for the target object with InitiateMu
ltipartUpl oadRequest .
InitiateMu ltipartUpl oadRequest initiateMu ltipartUpl
oadRequest = new InitiateMu ltipartUpl oadRequest ( destinatio
nBucketNam e , destinatio nObjectNam e );
InitiateMu ltipartUpl oadResult initiateMu ltipartUpl oadResult
= ossClient . initiateMu ltipartUpl oad ( initiateMu ltipartUpl
oadRequest );
String uploadId = initiateMu ltipartUpl oadResult . getUploadI
d ();

// Start the UploadPart Copy task .
List < PartETag > partETags = new ArrayList < PartETag >();
for ( int i = 0 ; i < partCount ; i ++ ) {
    // Calculate the size of each part .
    long skipBytes = partSize * i ;
    long size = partSize < contentLen gth - skipBytes ?
    partSize : contentLen gth - skipBytes ;
```

```

// Create UploadPart CopyRequest. You can specify
conditions with UploadPart CopyRequest.
UploadPart CopyRequest uploadPart CopyRequest =
    new UploadPart CopyRequest ( sourceBucketName , destinationBucketName , destination
nObjectName );
uploadPart CopyRequest . setUploadId ( uploadId );
uploadPart CopyRequest . setPartSize ( size );
uploadPart CopyRequest . setBeginIndex ( skipBytes );
uploadPart CopyRequest . setPartNumber ( i + 1 );
UploadPart CopyResult uploadPart CopyResult = ossClient .
uploadPart Copy ( uploadPart CopyRequest );

// Save the returned ETag of parts to partETags .
partETags . add ( uploadPart CopyResult . getPartETag ());
}

// Submit the UploadPart Copy task .
CompleteMultipartUploadRequest completeMultipartUploadRequest = new CompleteMultipartUploadRequest (
    destinationBucketName , destination
nObjectName , uploadId , partETags );
ossClient . completeMultipartUpload ( completeMultipartUploadRequest );

// Close your OSSClient .
ossClient . shutdown ();

```

For more information about UploadPartCopy, see [UploadPartCopy](#). For the complete code of UploadPartCopy, see [GitHub](#).

2.12.8 Restore an archive object

This topic describes how to restore an archive object.

For the complete code of restoring an archive object, see [GitHub](#).

You must restore an archive object before you read it. Do not call `restoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

1. An archive object is in the frozen state.
2. After you submit it for restoration, the server restores the object. The object is in the restoring state.
3. You can read the object after it is restored. The restored state of the object lasts one day by default. You can prolong this period to a maximum of seven days. Once this period expires, the object returns to the frozen state.

Run the following code to restore an object:

```

// This example uses endpoint China East 1 (Hangzhou).
Specify the actual endpoint based on your requirements.

```

```

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https://ram.console.aliyun
// . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

ObjectMeta objectMeta = ossClient . getObjectM
etadata ( bucketName , key );

// Verify whether the object is an archive object .//
// Verify that the file is an archive file .
StorageClass storageClass = objectMeta . getObjectS
torageClass ();
if ( storageClass == StorageClass . Archive ) {
    // Restore the object .
    ossClient . restoreObject ( bucketName , objectName );

    // Wait until the object is restored .
    do {
        Thread . sleep ( 1000 );
        objectMeta = ossClient . getObjectM etadata (
bucketName , objectName );
    } while (! objectMeta . isRestoreC ompleted ());
}

// Obtain the restored object .
OSSObject ossObject = ossClient . getObject ( bucketName ,
objectName );
ossObject . getObjectC ontent (). close ();

// Close your OSSClient .
ossClient . shutdown ();

```

For more information about the archive storage classes, see [Introduction to storage classes](#).

2.12.9 Manage a symbolic link

This topic describes how to manage a symbolic link.

Create a symbolic link

A symbolic link is a special object that maps to an object similar to a shortcut used in Windows. You can configure user-defined Object Meta for symbolic links.

Run the following code to create a symbolic link:

```
// This example uses endpoint China East 1 (Hangzhou). Specify the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all APIs in OSS. We recommend that you log on as a RAM user to access APIs or perform routine operations and maintenance. To create a RAM account, log on to https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String symLink = "<yourSymLink>";
String destinationObjectName = "<yourDestinationObjectName>";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint, accessKeyId, accessKeySecret);

// Create Object Meta.
ObjectMeta metadata = new ObjectMeta();
metadata.setContentType("text/plain");
// Set property to property-value for user-defined metadata.
metadata.addUserMetadata("property", "property-value");

// Create CreateSymLinkRequest.
CreateSymLinkRequest createSymLinkRequest = new CreateSymLinkRequest(bucketName, symLink, destinationObjectName);

// Configure Object Meta.
createSymLinkRequest.setMetadata(metadata);

// Create symbolic links.
// Create a symbolic link.

// Close your OSSClient.
ossClient.shutdown();
```

For more information about symbolic links, see [PutSymLink](#).

Obtain object content for a symbolic link

To obtain a symbolic link, you must have the read permission on it. Run the following code to obtain the object content for a symbolic link:

```
// This example uses endpoint China East 1 (Hangzhou). Specify the actual endpoint based on your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has
```



```

permissions on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenance . To create
a RAM account , log on to https :// ram . console . aliyun
.com .
String  accessKeyId = "< yourAccess KeyId >";
String  accessKeySecret = "< yourAccess KeySecret >";
String  bucketName = "< yourBucket Name >";
String  symLink = "< yourSymLink >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Obtain a symbolic link .
OSSSymlink symbolicLink = ossClient . getSymlink ( bucketName
, symLink );
// Print the content of the object that the symbolic
link directs to .
System . out . println ( symbolicLink . getSymlink () );
System . out . println ( symbolicLink . getTarget () );
System . out . println ( symbolicLink . getRequest Id () );

// Close your OSSClient .
ossClient . shutdown ();

```

For more information about symbolic links, see [GetSymlink](#).

2.13 Data security

OSS Java SDK provides data integrity checks and file encryption on the server and client to guarantee data security during file (object) uploads, downloads, and replication.

Data integrity checks

OSS Java SDK provides MD5 authentication and cyclic redundancy checks (CRC) to guarantee data integrity for object (file) uploads, downloads, and replication.

- MD5 authentication

If you configure Content-MD5 for file uploads, OSS calculates the MD5 value based on the received content. If the MD5 value is inconsistent with that of the uploaded file, `InvalidDigest` is returned. Thus, data integrity is guaranteed.

Run the following code for MD5 authentication during the file upload:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
String  endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has

```

```

permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// Upload a string .
String content = "Hello OSS ";

ObjectMetadata meta = new ObjectMetadata ();
// Configure MD5 authentication .
String md5 = BinaryUtil . toBase64String ( BinaryUtil .
calculateMD5 ( content . getBytes ()));
meta . setContentMD5 ( md5 );

ossClient . putObject ( "< yourBucket Name >", "< yourObjectName >", new
ByteArrayInputStream ( content . getBytes ()), meta
);

// Close your OSSClient instance .
ossClient . shutdown ();

```

**Note:**

MD5 authentication can be used for `putObject`, `getObject`, `appendObject`, `postObject`, and `uploadPart`.

- **CRC64**

CRC is enabled by default during object (file) uploads, downloads, and replication to guarantee data integrity.

Run the following code to check whether CRC takes effect during the file upload:

```

// This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId , accessKeySecret );

// Upload a string .

```

```
String content = " Hello OSS ";

PutObjectResult result = ossClient.putObject("<
yourBucket Name >", "< yourObject Name >", new ByteArrayInputStream
(inputStream ( content.getBytes ())) );

// Check whether CRC is enabled. True indicates it
is enabled while false indicates it is disabled.
if (ossClient.getClientConfiguration().isCrcCheck
Enabled()) {
// Check whether CRC takes effect based on the
response.
OSSUtils.checkChecksum (result.getClientCRC (),
result.getServerCRC (), result.getRequestId ());
}
// Close your OSSClient instance.
ossClient.shutdown ();
```

**Note:**

- You can use CRC64 authentication for putObject, getObject, appendObject, and uploadPart.
- CRC64 authentication affects the upload and download progress because CRC64 uses CPU space.

File encryption on the server

You can set the serverSideEncryption parameter in Object Meta to AES256 or KMS for file uploads to encrypt the file and save it on the server. For more information, see [Server-side encryption](#) in the OSS Developer Guide.

Run the following code for file encryption on the server during file uploads:

```
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements.
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
.aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Upload a string .
String content = " Hello OSS ";

ObjectMeta metaData = new ObjectMeta ();
```

```
// Set the server-side encryption code to AES256 for the file.
meta.setServerSideEncryption ( ObjectMeta data . AES_256_SERVER_SIDE_ENCRYPTION );

ossClient.putObject ("< yourBucket Name >", "< yourObject Name >", new ByteArrayInputStream ( content . getBytes () ), meta );

// Close your OSSClient instance .
ossClient.shutdown ();
```

Client-side encryption

To encrypt a file on the client, you can encrypt the data on your local device, upload it to OSS, download it to the local device, and decrypt it. For more information, see [Introduction to client-side encryption SDK](#) in OSS Developer Guide.

2.14 Authorized access

This topic describes how to authorize access to users.

Use STS for temporary access authorization

OSS supports Alibaba Cloud Security Token Service (STS) for temporary access authorization. STS is a web service that provides a temporary access token to a cloud computing user. Through the STS, you can assign a third-party application or a RAM user (you can manage the user ID) an access credential with a custom validity period and permissions. For more information about STS, see [STS introduction](#).

STS advantages:

- Your long-term key (AccessKey) is not exposed to a third-party application. You only need to generate an access token and send the access token to the third-party application. You can customize access permissions and the validity of this token.
- You do not need to keep track of permission revocation issues. The access token automatically becomes invalid when it expires.

For more information about the process of access to OSS with STS, see [RAM and STS scenario practices](#) in OSS Developer Guide.

Run the following code to create a signature request with STS:

```
// This example uses endpoint China ( Hangzhou ). Specify the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all APIs in OSS . We recommend that
```

```

    you log on as a RAM user to access APIs or
    perform routine operations and maintenance. To create
    a RAM account, log on to https://ram.console.aliyun
    .com.
    String accessKeyId = "< yourAccess KeyId >";
    String accessKeySecret = "< yourAccess KeySecret >";
    String securityToken = "< yourSecurityToken >";

    // After a user obtains a temporary STS credential,
    the OSSClient is generated with the security token
    and temporary access key (AccessKeyId and AccessKeyS
    ecret).
    // Create an OSSClient instance.
    OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
    , accessKeySecret , securityToken );

    // Perform operations on OSS.

    // Close your OSSClient.
    ossClient.shutdown();

```

Sign a URL to authorize temporary access

- Sign a URL

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time for a URL to restrict the period of access from visitors.

- Sign a URL for access with HTTP GET

Use the following code to sign a URL that allows access with HTTP GET:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements.
String endpoint = " http://oss-cn-hangzhou.aliyuncs.
com ";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all APIs in OSS. We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenance. To create
a RAM account, log on to https://ram.console.
aliyun.com.
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Set the expiration time of a URL to one hour.
Date expiration = new Date ( new Date (). getTime () +
3600 * 1000 );
// Generate the URL that allows access with HTTP
GET. Visitors can use a browser to access relevant
content.

```

```

URL url = ossClient.generatePresignedUrl ( bucketName ,
objectName , expiration );

// Close your OSSClient .
ossClient.shutdown ();

```

- **Sign a URL for access with other HTTP methods**

A URL needs to be signed for temporary access from a visitor to perform other operations such as file upload and deletion. For example:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = "oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenance . To create
a RAM account , log on to https://ram.console.
aliyun.com .
String accessKeyId = "< yourAccessKeyId >";
String accessKeySecret = "< yourAccessKeySecret >";
String securityToken = "< yourSecurityToken >";
String bucketName = "< yourBucketName >";
String objectName = "< yourObjectName >";

// After a user obtains a temporary STS credential ,
the OSSClient is generated with the security token
and temporary access key ( AccessKeyId and AccessKeyS
ecret ).
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeySecret , securityToken );

GeneratePresignedUrlRequest request = new GeneratePr
esignedUrlRequest ( bucketName , objectName , HttpMethod . PUT
);
// Set the expiration time of a URL to one hour .
Date expiration = new Date ( new Date (). getTime () +
3600 * 1000 );
request.setExpiration ( expiration );
// Set ContentType .
request.setContentType ( DEFAULT_OBJECT_CONTENT_TYPE );
// Configure custom Object Meta .
request.addUserMetadata ( "author", "aliy");

// Sign a URL that allows access with HTTP PUT .
URL signedUrl = ossClient.generatePresignedUrl ( request );

Map < String , String > requestHeaders = new HashMap <
String , String > ();
requestHeaders.put ( HttpHeaders . CONTENT_TYPE ,
DEFAULT_OBJECT_CONTENT_TYPE );
requestHeaders.put ( OSS_USER_METADATA_PREFIX + "author
", "aliy");

// Upload a file with the signed URL .
ossClient.putObject ( signedUrl , new ByteArrayInputStream
(" Hello OSS ". getBytes ()), - 1 , requestHeaders , true );

// Close your OSSClient .

```

```
ossClient . shutdown ();
```

Visitors can use a signed URL to upload a file by passing in the `HttpMethod.PUT` parameter.

- Add specified parameters to a URL

Run the following code to add specified parameters to a URL:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console .
// aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Create a request .
GeneratePr esignedUrl Request generatePr esignedUrl Request
= new GeneratePr esignedUrl Request ( bucketName , objectName
);
// Set HttpMethod to PUT .
generatePr esignedUrl Request . setMethod ( HttpMethod . PUT );
// Add custom Object Meta .
generatePr esignedUrl Request . addUserMet adata ( " author ", "
baymax " );
// Add Content - Type .
generatePr esignedUrl Request . setContent Type ( " applicatio n
/ octet - stream " );
// Set the expiration time of a URL to one hour .
Date expiration = new Date ( new Date (). getTime () +
3600 * 1000 );
generatePr esignedUrl Request . setExpirat ion ( expiration );
// Generate the signed URL .
URL url = ossClient . generatePr esignedUrl ( generatePr
esignedUrl Request );

// Close your OSSClient .
```

```
ossClient . shutdown ();
```

- Use a signed URL to obtain or upload an object
- Use a signed URL to obtain an object

Run the following code to obtain a specified object through a signed URL:

```
// This example uses endpoint China ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey
// of an Alibaba Cloud account because the account
// has permission s on all APIs in OSS . We
// recommend that you log on as a RAM user to
// access APIs or perform routine operations and
// maintenanc e . To create a RAM account , log on to
// https :// ram . console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint ,
accessKeyI d , accessKeyS ecret );

Date expiration = DateUtil . parseRfc82 2Date ( " Wed , 18
Mar 2022 14 : 20 : 00 GMT " );
GeneratePr esignedUrl Request request = new GeneratePr
esignedUrl Request ( bucketName , objectName , HttpMethod .
GET );
// Configure the expiration time .
request . setExpirat ion ( expiration );
// Generate a signed URL that allows HTTP GET
// access .
URL signedUrl = ossClient . generatePr esignedUrl ( request
);
System . out . println ( " signed url for getObject : " +
signedUrl );

// Use the signed URL to send a request .
Map < String , String > customHead ers = new HashMap <
String , String > ();
// Add a request header to GetObject .
customHead ers . put ( " Range " , " bytes = 100 - 1000 " );
OSSObject object = ossClient . getObject ( signedUrl ,
customHead ers );

// Close your OSSClient .
ossClient . shutdown ();
```

- Use a signed URL to upload a file

Run the following code to upload a file with a signed URL:

```
// This example uses endpoint China ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
```



```

String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey
// of an Alibaba Cloud account because the account
// has permissions on all APIs in OSS. We
// recommend that you log on as a RAM user to
// access APIs or perform routine operations and
// maintenance. To create a RAM account, log on to
// https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint,
    accessKeyId, accessKeySecret);

// Generate the signed URL.
Date expiration = DateUtil.parseRfc822Date("Thu, 19
Mar 2019 18:00:00 GMT");
GeneratePresignedUrlRequest request = new GeneratePresignedUrlRequest(
    bucketName, objectName, HttpMethod.PUT);
// Configure the expiration time.
request.setExpiration(expiration);
// Configure Content-Type.
request.setContentType("application/octet-stream");
// Configure custom Object Meta.
request.addUserMetadata("author", "aliy");
// Generate a signed URL that allows access with
// HTTP PUT.
URL signedUrl = ossClient.generatePresignedUrl(request);
System.out.println("signed url for putObject: " +
    signedUrl);

// Use the signed URL to send a request.
File f = new File("<yourLocalFile>");
FileInputStream fin = new FileInputStream(f);
// Add a request header to PutObject.
Map<String, String> customHeaders = new HashMap<
String, String>();
customHeaders.put("Content-Type", "application/octet-stream");
customHeaders.put("x-oss-meta-author", "aliy");

PutObjectResult result = ossClient.putObject(signedUrl,
    fin, f.length(), customHeaders);

// Close your OSSClient.
ossClient.shutdown();

```

2.15 Manage lifecycle rules

This topic describes how to manage lifecycle rules.

You can use OSS to configure lifecycle rules to reduce storage costs. The lifecycle management rules can be used for automatic deletion of expired objects and

fragments, or storage class conversion to Infrequent Access for objects that will expire soon. Each rule includes:

- **Rule ID:** It identifies a rule. Ensure that each rule ID in a bucket is unique.
- **Policy:** You can configure a policy by using the following configuration methods. Only one method can be configured for a bucket.
 - **Configure by Prefix:** You can create multiple rules with this method. Ensure that each prefix is unique.
 - **Configure for Entire Bucket:** You can configure only one rule with this method.
- **Expired:** You can configure the following configuration methods:
 - **Set by Number of Days:** It specifies the number of days from the last modification time of objects.
 - **Set by Date:** It specifies the date prior to which objects are created. If objects are created prior to the date, these objects are deleted.
- **Status:** This lifecycle rule takes effect or not.

The parts uploaded by uploadPart method also support lifecycle rules. The last modification time of an object is the time the multipart upload event is initiated.

For more information about lifecycle management, see [Manage object lifecycle](#).

Configure lifecycle rules

Run the following code to configure lifecycle rules:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

SetBucketLifecycleRequest request = new SetBucketLifecycleRequest ( bucketName );

// Configure the rule ID and object name prefix .
String ruleId0 = " rule0 ";
```

```

String matchPrefi x0 = " A0 /";
String ruleId1 = " rule1 ";
String matchPrefi x1 = " A1 /";
String ruleId2 = " rule2 ";
String matchPrefi x2 = " A2 /";
String ruleId3 = " rule3 ";
String matchPrefi x3 = " A3 /";

// Set the expiration duration to 3 days after last
modified date .
request . AddLifecyc leRule ( new LifecycleR ule ( ruleId0 ,
matchPrefi x0 , RuleStatus . Enabled , 3 ));

// Configure expiration date to delete objects that are
created before the date .
LifecycleR ule rule = new LifecycleR ule ( ruleId1 ,
matchPrefi x1 , RuleStatus . Enabled );
rule . setCreated BeforeDate ( DateUtil . parseISO86 01Date ( " 2022
- 10 - 12T00 : 00 : 00 . 000Z "));
request . AddLifecyc leRule ( rule );

// Set the expiration duration to 3 days for parts
in an object .
rule = new LifecycleR ule ( ruleId2 , matchPrefi x2 ,
RuleStatus . Enabled );
LifecycleR ule . AbortMulti partUpload abortMulti partUpload =
new LifecycleR ule . AbortMulti partUpload ();
abortMulti partUpload . setExpirat ionDays ( 3 );
rule . setAbortMu ltipartUpl oad ( abortMulti partUpload );
request . AddLifecyc leRule ( rule );

// Configure expiration date to delete parts that are
created before the date .
rule = new LifecycleR ule ( ruleId3 , matchPrefi x3 ,
RuleStatus . Enabled );
abortMulti partUpload = new LifecycleR ule . AbortMulti
partUpload ();
abortMulti partUpload . setCreated BeforeDate ( DateUtil .
parseISO86 01Date ( " 2022 - 10 - 12T00 : 00 : 00 . 000Z "));
rule . setAbortMu ltipartUpl oad ( abortMulti partUpload );
request . AddLifecyc leRule ( rule );

ossClient . setBucketL ifecycle ( request );

// Close your OSSClient .
ossClient . shutdown ();

```

View lifecycle rules

Run the following code to view lifecycle rules:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .

```

```

String  accessKeyId = "< yourAccess KeyId >";
String  accessKeySecret = "< yourAccess KeySecret >";
String  bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

List< LifecycleRule > rules = ossClient . getBucketLifecycle
( bucketName );
for ( LifecycleRule rule : rules ) {
    System . out . println ( rule . getId ());
    System . out . println ( rule . getPrefix ());
    System . out . println ( rule . getExpirationDays ());
    System . out . println ( rule . getCreatedBeforeDate ());
    if ( rule . hasAbortMultipartUpload () ) {
        System . out . println ( rule . getAbortMultipartUpload ().
getExpirationDays ());
        System . out . println ( rule . getAbortMultipartUpload ().
getCreatedBeforeDate ());
    }
}

// Close your OSSClient .
ossClient . shutdown ();

```

Clear lifecycle rules

Run the following code to clear lifecycle rules:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String  accessKeyId = "< yourAccess KeyId >";
String  accessKeySecret = "< yourAccess KeySecret >";
String  bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

ossClient . deleteBucketLifecycle ( bucketName );

// Close your OSSClient .

```

```
ossClient . shutdown ();
```

2.16 CORS

This topic describes how to perform CORS.

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) and [PutBucketcors](#) in OSS Developer Guide.

Configure CORS rules

Run the following code to configure CORS rules for the specified bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

SetBucketC ORSRequest request = new SetBucketC ORSRequest (
bucketName );

// Create a CORS rule . A maximum of 10 rules can
// be configured for each bucket .
ArrayList < CORSRule > putCorsRul es = new ArrayList < CORSRule
>();

CORSRule corRule = new CORSRule ();

ArrayList < String > allowedOri gin = new ArrayList < String
>();
// Specify the source of the cross - origin access
// request .
allowedOri gin . add ( " http :// www . b . com " );

ArrayList < String > allowedMet hod = new ArrayList < String
>();
// Specify the cross - region request methods ( GET , PUT
// , DELETE , POST , and HEAD ) that are allowed .
allowedMet hod . add ( " GET " );
```

```

ArrayList < String > allowedHeader = new ArrayList < String
>();
// Specify whether the header specified in Access -
Control - Request - Headers of pre - flight request ( OPTIONS
) is allowed .
allowedHeader . add ( " x - oss - test " );

ArrayList < String > exposedHeader = new ArrayList < String
>();
// Specify the response header that allows user access
from applications .
exposedHeader . add ( " x - oss - test1 " );
// AllowedOrigins and AllowedMethods allow only one
wildcard asterisk (*). Wildcard asterisks (*) indicate
that all sources of the cross - origin requests and
operations are allowed .
corRule . setAllowed Methods ( allowedMethod );
corRule . setAllowed Origins ( allowedOrigin );
// AllowedHeaders and ExposeHeaders do not allow
wildcard asterisks (*).
corRule . setAllowed Headers ( allowedHeader );
corRule . setExposeHeaders ( exposedHeader );
// Specify the cache time ( seconds ) for the response
of browser pre - flight ( OPTIONS ) requests to a
specific resource .
corRule . setMaxAgeSeconds ( 10 );

// A maximum of 10 rules is allowed .
putCorsRules . add ( corRule );
// The existing rules will be replaced .
request . setCorsRules ( putCorsRules );
ossClient . setBucketCORS ( request );

// Close your OSSClient .
ossClient . shutdown ();

```

Obtain CORS rules

Run the following code to obtain CORS rules:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

ArrayList < CORSRule > corsRules ;
// Obtain the CORS rule list .

```

```

corsRules = ( ArrayList < CORSRule >) ossClient . getBucketC
ORSRules ( bucketName );
for ( CORSRule rule : corsRules ) {
    for ( String allowedOri gin1 : rule . getAllowed Origins
()) {
        // Obtain allowed sources of cross - origin requests .
        System . out . println ( allowedOri gin1 );
    }

    for ( String allowedMet hod1 : rule . getAllowed Methods
()) {
        // Obtain the allowed cross - origin request method .
        System . out . println ( allowedMet hod1 );
    }

    if ( rule . getAllowed Headers (). size () > 0 ){
        for ( String allowedHea der1 : rule . getAllowed Headers
()) {
            // Obtain the header list for allowed cross -
origin requests .
            System . out . println ( allowedHea der1 );
        }
    }

    if ( rule . getExposeH eaders (). size () > 0 ) {
        for ( String exposeHead er : rule . getExposeH eaders
()) {
            // Obtain the header for an allowed cross -
origin request .
            System . out . println ( exposeHead er );
        }
    }

    if ( null != rule . getMaxAgeS econds () ) {
        System . out . println ( rule . getMaxAgeS econds () );
    }
}

// Close your OSSClient .
ossClient . shutdown ();

```

Delete CORS rules

Run the following code to delete all CORS rules for the specified bucket:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .

```

```

OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Delete all CORS rules for a specified bucket .
ossClient . deleteBucketCORSRules ( bucketName );

// Close your OSSClient .
ossClient . shutdown ();

```

2.17 Set logging

You can enable access logs to record bucket access to log files, which are stored in a specified bucket.

The log file format is as follows:

```

< TargetPrefix >< SourceBucket >- YYYY - mm - DD - HH - MM - SS -
UniqueString

```

For more information about access log files, see [Set access logging](#).

Enable access logging

Run the following code to enable bucket access logging:

```

// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

SetBucketLoggingRequest request = new SetBucketL
oggingRequest ("< yourSource BucketName >");
// Configure the bucket that stores log files .
request . setTargetBucket ("< yourTarget BucketName >");
// Configure the log file storage directory .
request . setTargetPrefix ("< yourTarget Prefix >");
ossClient . setBucketLogging ( request );

// Close your OSSClient .

```



```
ossClient . shutdown ();
```

View access logging configurations

Run the following code to view the access logging configurations for a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

BucketLoggingResult result = ossClient . getBucketLogging ( "<
yourSource BucketName >");
System . out . println ( result . getTargetBucket () );
System . out . println ( result . getTargetPrefix () );

// Close your OSSClient .
ossClient . shutdown ();
```

Disable access logging

Run the following code to disable access logging for a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

SetBucketLoggingRequest request = new SetBucketLoggingRequest ( "< yourSource BucketName >");
request . setTargetBucket ( null );
request . setTargetPrefix ( null );
ossClient . setBucketLogging ( request );

// Close your OSSClient .
```

```
ossClient . shutdown ();
```

2.18 Static website hosting

You can set your bucket configuration to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

For more information about static website hosting, see [Static website hosting](#).

Configure static website hosting

Run the following code to configure static website hosting:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

SetBucketW ebsiteRequ est request = new SetBucketW
ebsiteRequ est ("< yourBucket Name >");
request . setIndexDo cument (" index . html ");
request . setErrorDo cument (" error . html ");
ossClient . setBucketW ebsite ( request );

// Close your OSSClient .
ossClient . shutdown ();
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
```

```
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance.
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

BucketWebsiteResult result = ossClient . getBucketWebsite ( "<
yourBucket Name >");
System . out . println ( result . getIndexDocument ());
System . out . println ( result . getErrorDocument ());

// Close your OSSClient .
ossClient . shutdown ();
```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

ossClient . deleteBucketWebsite ( "< yourBucket Name >");

// Close your OSSClient .
ossClient . shutdown ();
```

2.19 Anti-leech

This topic describes how to use anti-leech.

To prevent your data on OSS from being leached, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leaching, see [Anti-leeching settings](#).

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

List < String > refererLis t = new ArrayList < String > ();
// Add the referer field . The referer field allows
// question marks (?) and asterisks (*) for wildcard use .
// refererLis t . add ( " http :// www . aliyun . com " );
// refererLis t . add ( " http :// www . *. com " );
// refererLis t . add ( " http :// www . ?. aliyuncs . com " );
// Configure the referer list for a bucket . Set the
// parameter to true , which allows the referer field to
// be empty .
BucketRefe rer br = new BucketRefe rer ( true , refererLis
t );
ossClient . setBucketR eferer ( bucketName , br );

// Close your OSSClient .
ossClient . shutdown ();
```

Obtain a referer whitelist

Run the following code to obtain a referer whitelist:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );
```

```
// Obtain the referer list for a bucket .
BucketReferer br = ossClient . getBucketReferer ( bucketName
);
List < String > refererList = br . getReferer List ();
for ( String referer : refererList ) {
    System . out . println ( referer );
}

// Close your OSSClient .
ossClient . shutdown ();
```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// You cannot clear a referer whitelist directly . To
clear a referer whitelist , you need to create the
rule that allows an empty referer field and replace
the original rule with the new rule .
BucketReferer br = new BucketReferer ();
ossClient . setBucketReferer ( bucketName , br );

// Close your OSSClient .
ossClient . shutdown ();
```

2.20 Cross-region replication

Cross-region replication is used to automatically and asynchronously copy objects across buckets in different regions. Any changes (creation, replacement, and deletion) to objects in the source bucket will be synchronized to the target bucket. This function is used to meet the requirements of cross-region disaster recovery and data replication.

For more information about cross-region replication, see [Manage cross-region replication](#).

Enable cross-region replication

Run the following code to enable cross-region replication:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

AddBucketR eplication Request request = new AddBucketR
eplication Request ( bucketName );
request . setReplica tionRuleID ( "< yourRuleId >");
request . setTargetB ucketName ( "< yourTarget BucketName >");
// This example uses the target endpoint China ( Beijing
// ).
request . setTargetB ucketLocat ion ( " oss - cn - beijing " );
// Disable historical data replicatio n . Historical data
// replicatio n is enabled by default .
request . setEnableH istoricalO bjectRepli cation ( false );
ossClient . addBucketR eplication ( request );

// Close your OSSClient .
ossClient . shutdown ();
```

View cross-region replication configurations

Use the following code to view the cross-region replication configurations for a bucket:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
```

```

OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

List < ReplicationRule > rules = ossClient . getBucketR
eplication ( bucketName );
for ( ReplicationRule rule : rules ) {
    System . out . println ( rule . getReplica tionRuleID ());
    System . out . println ( rule . getTargetB ucketLocat ion ());
    System . out . println ( rule . getTargetB ucketName ());
}

// Close your OSSClient .
ossClient . shutdown ();

```

View cross-region replication progress

Cross-region replication progress includes the progress of historical and real-time data synchronization.

- Percent signs (%) indicate the progress of historical data synchronization. This function only works for buckets for which historical data synchronization has been enabled.
- Time points (the point of latest time the data is written) indicate the progress of real-time data synchronization. If a time point is displayed, the data created before this time point has been synchronized.

Run the following code to view the cross-region replication progress:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

BucketRepl icationPro gress process = ossClient . getBucketR
eplication Progress ( bucketName , "< yourRuleId >");
System . out . println ( process . getReplica tionRuleID ());
// Check whether historical data synchroniz ation is
enabled .
System . out . println ( process . isEnabledHi storicalOb
jectRepl ication ());
// Print historical data synchroniz ation progress .

```

```

System . out . println ( process . getHistori calObjectP rogress
());
// Print real - time data synchroniz ation progress .
System . out . println ( process . getNewObje ctProgress ());

// Close your OSSClient .
ossClient . shutdown ();

```

View the target regions for synchronization

Run the following code to obtain the target regions to which buckets can be synchronized:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI d
, accessKeyS ecret );

List < String > locations = ossClient . getBucketR eplication
Location ( bucketName );
for ( String loc : locations ) {
    System . out . println ( loc );
}

// Close your OSSClient .
ossClient . shutdown ();

```

Disable cross-region replication

Use the following code to disable cross-region replication:

```

// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";

```



```
String bucketName = "< yourBucket Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyId
, accessKeySecret );

// Disable cross - region replication . After this
function is disabled , objects in the target bucket
exist and all changes in the source objects cannot
be synchronized .
ossClient . deleteBucketReplication ( bucketName , "< yourRuleId
>");

// Close your OSSClient .
ossClient . shutdown ();
```

2.21 Image processing

Image Processing (IMG) is a massive, secure, cost-effective and highly reliable image processing service. After source images are uploaded to OSS, you can process images on any Internet device at any anytime, from anywhere through simple RESTful APIs.

For more information about IMG, see [Image Processing](#).

For the complete code of IMG, see [GitHub](#).

Basic features

OSS offers the following IMG features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Image processing](#)
- [Image processing access rules](#): calls multiple IMG functions.

Usage

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- **Anonymous access**

You can use the following format in level-3 domains to access a processed image:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Specifies the name of a bucket.
endpoint	Specifies endpoint used to access a region.
object	Specifies the name of an image object.
image	Specifies the reserved identifier for IMG .
action	Specifies the operations on an image, such as scaling, cropping, and rotating.

Parameter	Description
param	Specifies the parameter that indicates the operation on an image.

- Basic operations

For example, scale an image to a width of 100 px. Adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customized image styles

You can use the following format in level-3 domains to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- **style**: specifies a reserved identifier of a customized image style.
- **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image sequentially.

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- HTTPS access

IMG supports access through HTTPS. Example:

```
https :// image - demo . oss - cn - hangzhou . aliyuncs . com /
example . jpg ? x - oss - process = image / resize , w_100
```

- **Authorized access**

Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

Run the following code to generate a signed URL for IMG:

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";
// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );
// Customize the image style .
String style = " image / resize , m_fixed , w_100 , h_100 /
rotate , 90 ";
// Set the expiration time to 10 minutes .
Date expiration = new Date ( new Date (). getTime () +
1000 * 60 * 10 );
GeneratePr esignedUrl Request req = new GeneratePr
esignedUrl Request ( bucketName , objectName , HttpMethod . GET
);
req . setExpirat ion ( expiration );
req . setProcess ( style );
URL signedUrl = ossClient . generatePr esignedUrl ( req );
System . out . println ( signedUrl );
// Close your OSSClient .
```

```
ossClient . shutdown (); sClient . shutdown ();
```

- Access with SDK

You can use SDK to access and process an image object with any ACLs.

SDK allows you to customize image styles, HTTPS access, and cascade operations.

- Basic operations

Run the following code to perform basic operations on an image:

```
// This example uses endpoint China East 1 (
// Hangzhou). Specify the actual endpoint based on
// your requirements.
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// It is highly risky to log on with AccessKey
// of an Alibaba Cloud account because the account
// has permissions on all the APIs in OSS. We
// recommend that you log on as a RAM user to
// access APIs or perform routine operations and
// maintenance. To create a RAM account, log on to
// https://ram.console.aliyun.com.
String accessKeyId = "<yourAccessKeyId>";
String accessKeySecret = "<yourAccessKeySecret>";
String bucketName = "<yourBucketName>";
String objectName = "<yourObjectName>";
// Create an OSSClient instance.
OSSClient ossClient = new OSSClient(endpoint,
accessKeyId, accessKeySecret);
// Scale the image.
String style = "image/resize,m_fixed,w_100,h_100";
GetObjectRequest request = new GetObjectRequest(
bucketName, objectName);
request.setProcess(style);
ossClient.getObject(request, new File("example-resize.jpg"));
// Crop the image.
style = "image/crop,w_100,h_100,x_100,y_100,r_1";
request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
ossClient.getObject(request, new File("example-crop.jpg"));
// Rotate the image.
style = "image/rotate,90";
request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
ossClient.getObject(request, new File("example-rotate.jpg"));
// Sharpen the image.
style = "image/sharpen,100";
request = new GetObjectRequest(bucketName, objectName);
request.setProcess(style);
ossClient.getObject(request, new File("example-sharpen.jpg"));
// Add watermarks.
style = "image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ";
```

```

request = new GetObjectRequest ( bucketName , objectName
);
request . setProcess ( style );
ossClient . getObject ( request , new File ( " example -
watermark . jpg " ));
// Convert the image format .
style = " image / format , png ";
request = new GetObjectRequest ( bucketName , objectName
);
request . setProcess ( style );
ossClient . getObject ( request , new File ( " example - format
. png " ));
// Obtain image information .
style = " image / info ";
request = new GetObjectRequest ( bucketName , objectName
);
request . setProcess ( style );
ossClient . getObject ( request , new File ( " example - info .
txt " ));
// Close your OSSClient .
ossClient . shutdown ();

```

- Customize an image style

Run the following code to customize an image style:

```

// This example uses endpoint China ( Hangzhou
). Specify the actual endpoint based on your
requirements .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey
of an Alibaba Cloud account because the account
has permissions on all APIs in OSS . We
recommend that you log on as a RAM user to
access APIs or perform routine operations and
maintenance . To create a RAM account , log on to
https :// ram . console . aliyun . com .
String accessKeyId = "< yourAccess KeyId >";
String accessKeySecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint ,
accessKeyId , accessKeySecret );

// Customize the image style .
String style = " style /< yourCustom StyleName >";
GetObjectRequest request = new GetObjectRequest (
bucketName , objectName );
request . setProcess ( style );

ossClient . getObject ( request , new File ( " example - new .
jpg " ));

// Close your OSSClient .

```

```
ossClient . shutdown ();
```

- Cascade operations

Run the following code to perform cascade operations on an image:

```
// This example uses endpoint China ( Hangzhou ). Specify the actual endpoint based on your
// requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs .
com ";
// It is highly risky to log on with AccessKey
// of an Alibaba Cloud account because the account
// has permission s on all APIs in OSS . We
// recommend that you log on as a RAM user to
// access APIs or perform routine operations and
// maintenanc e . To create a RAM account , log on to
// https :// ram . console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String objectName = "< yourObject Name >";

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint ,
accessKeyI d , accessKeyS ecret );

// Perform cascade operations .
String style = " image / resize , m_fixed , w_100 , h_100 /
rotate , 90 ";
GetObjectR equest request = new GetObjectR equest (
bucketName , objectName );
request . setProcess ( style );

ossClient . getObject ( request , new File ( " example - new .
jpg " ));

// Disable the OSSClient .
ossClient . shutdown ();
```

Image processing persistence

Run the following code for image processing persistence:

```
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e .
// To create a RAM account , log on to https :// ram .
// console . aliyun . com .
String accessKeyI d = "< yourAccess KeyId >";
String accessKeyS ecret = "< yourAccess KeySecret >";
String bucketName = "< yourBucket Name >";
String sourceImag e = "< yourSource ImageName >";

try {
```

```

// Create an OSSClient instance .
OSSClient ossClient = new OSSClient ( endpoint , accessKeyI
d , accessKeyS ecret );

// Image processing persistenc e : Scale the image .
StringBuilder sbStyle = new StringBuilder ();
Formatter styleForma tter = new Formatter ( sbStyle );
String styleType = " image / resize , m_fixed , w_100 , h_100 ";
String targetImag e = " example - resize . png ";
styleForma tter . format ("% s | sys / saveas , o_ % s , b_ % s " ,
styleType ,
    BinaryUtil . toBase64St ring ( targetImag e . getBytes ()),
    BinaryUtil . toBase64St ring ( bucketName . getBytes ()));
System . out . println ( sbStyle . toString ());
ProcessObj ectRequest request = new ProcessObj ectRequest (
bucketName , sourceImag e , sbStyle . toString ());
GenericRes ult processRes ult = ossClient . processObj ect (
request );
String json = IOUtils . readStream AsString ( processRes ult .
getRespons e (). getContent (), " UTF - 8 ");
processRes ult . getRespons e (). getContent (). close ();
System . out . println ( json );

// Image processing persistenc e : Clip the image .
sbStyle . delete ( 0 , sbStyle . length ());
styleType = " image / crop , w_100 , h_100 , x_100 , y_100 , r_1 ";
targetImag e = " example - crop . png ";
styleForma tter . format ("% s | sys / saveas , o_ % s , b_ % s " ,
styleType ,
    BinaryUtil . toBase64St ring ( targetImag e . getBytes ()),
    BinaryUtil . toBase64St ring ( bucketName . getBytes ()));
System . out . println ( sbStyle . toString ());
request = new ProcessObj ectRequest ( bucketName , sourceImag
e , sbStyle . toString ());
processRes ult = ossClient . processObj ect ( request );
json = IOUtils . readStream AsString ( processRes ult .
getRespons e (). getContent (), " UTF - 8 ");
processRes ult . getRespons e (). getContent (). close ();
System . out . println ( json );

// Image processing persistenc e : Rotate the object .
sbStyle . delete ( 0 , sbStyle . length ());
styleType = " image / rotate , 90 ";
targetImag e = " example - rotate . png ";
styleForma tter . format ("% s | sys / saveas , o_ % s , b_ % s " ,
styleType ,
    BinaryUtil . toBase64St ring ( targetImag e . getBytes ()),
    BinaryUtil . toBase64St ring ( bucketName . getBytes ()));
request = new ProcessObj ectRequest ( bucketName , sourceImag
e , sbStyle . toString ());
processRes ult = ossClient . processObj ect ( request );
json = IOUtils . readStream AsString ( processRes ult .
getRespons e (). getContent (), " UTF - 8 ");
processRes ult . getRespons e (). getContent (). close ();
System . out . println ( json );

// Image processing persistenc e : Sharpen the image .
sbStyle . delete ( 0 , sbStyle . length ());
styleType = " image / sharpen , 100 ";
targetImag e = " example - sharpen . png ";
styleForma tter . format ("% s | sys / saveas , o_ % s , b_ % s " ,
styleType ,
    BinaryUtil . toBase64St ring ( targetImag e . getBytes ()),
    BinaryUtil . toBase64St ring ( bucketName . getBytes ()));

```



```

    request = new ProcessObjectRequest ( bucketName , sourceImage , sbStyle . toString ());
    processResult = ossClient . processObject ( request );
    json = IOUtils . readStreamAsString ( processResult .
getResponse (). getContent (), " UTF - 8 ");
    processResult . getResponse (). getContent (). close ();
    System . out . println ( json );

// Image processing persistence : Add watermarks .
sbStyle . delete ( 0 , sbStyle . length ());
styleType = " image / watermark , text_SGVsb G8g5Zu - 54mH5pyN5YqhIQ ";
targetImage = " example - watermark . png ";
styleFormatter . format ("% s | sys / saveas , o_ % s , b_ % s " ,
styleType ,
    BinaryUtil . toBase64String ( targetImage . getBytes ()),
    BinaryUtil . toBase64String ( bucketName . getBytes ()));
    request = new ProcessObjectRequest ( bucketName , sourceImage , sbStyle . toString ());
    processResult = ossClient . processObject ( request );
    json = IOUtils . readStreamAsString ( processResult .
getResponse (). getContent (), " UTF - 8 ");
    processResult . getResponse (). getContent (). close ();
    System . out . println ( json );

// Image processing persistence : Convert the image
format .
sbStyle . delete ( 0 , sbStyle . length ());
styleType = " image / format , jpg ";
targetImage = " example - formatconvert . jpg ";
styleFormatter . format ("% s | sys / saveas , o_ % s , b_ % s " ,
styleType ,
    BinaryUtil . toBase64String ( targetImage . getBytes ()),
    BinaryUtil . toBase64String ( bucketName . getBytes ()));
    request = new ProcessObjectRequest ( bucketName , sourceImage , sbStyle . toString ());
    processResult = ossClient . processObject ( request );
    json = IOUtils . readStreamAsString ( processResult .
getResponse (). getContent (), " UTF - 8 ");
    processResult . getResponse (). getContent (). close ();
    System . out . println ( json );

} catch ( Exception e ) {
    e . printStackTrace ();
}

```

IMG tools

You can use the IMG viewer [ImageStyleViewer](#) to directly view the IMG result.

2.22 Exception handling

OSS Java SDK has two types of exceptions: `ClientException` and `OSSEException`. Both share the properties of `RuntimeException`.

Example for handling exceptions

Run the following code to handle exceptions:

```
try {
    // Perform operations on OSS , for example , object
    uploads .
    ossClient . putObject (...);
} catch ( OSSEExcepti on oe ) {
    System . out . println (" Caught an OSSEExcepti on , which
    means your request made it to OSS , "
    + " but was rejected with an error response
    for some reason ." );
    System . out . println (" Error Message : " + oe . getErrorCo
    de ());
    System . out . println (" Error Code : " + oe .
    getErrorCo de ());
    System . out . println (" Request ID : " + oe .
    getRequest Id ());
    System . out . println (" Host ID : " + oe .
    getHostId ());
} catch ( ClientExce ption ce ) {
    System . out . println (" Caught an ClientExce ption ,
    which means the client encountere d "
    + " a serious internal problem while trying to
    communicat e with OSS , "
    + " such as not being able to access the
    network ." );
    System . out . println (" Error Message : " + ce . getMessage
    ());
} finally {
    if ( ossClient != null ) {
        ossClient . shutdown ();
    }
}
```

ClientException

The `ClientException` exception occurs when the client attempts to send requests and when the data is transmitted to OSS. For example, when a request is sent, `ClientException` may occur due to unavailable network connections. `ClientException` may occur during file uploads due to IO exceptions.

OSSEException

`ServiceException` indicates a server exception. The exception occurs because a server error message is parsed. `OSSEException` includes the error code and information returned from OSS. It is used to locate and handle problems.

The following table describes common OSSException information.

Parameter	Description
Code	Specifies the error code returned by OSS.
Message	Specifies the detailed error information returned by OSS.
RequestId	Specifies the UUID. It is unique and used to identify a request. If a problem persists, send the RequestId to OSS developers for help.
HostId	Identifies an OSS cluster. Ensure that the value of HostId is consistent with the Host (endpoint to access a bucket) used in the request.

Common OSS error codes

Error Code	Description	HTTP status code
AccessDenied	Access denied	403
BucketAlreadyExists	The bucket already exists.	409
BucketNotEmpty	The bucket is not empty.	409
EntityTooLarge	The entity size exceeds the maximum limit.	400
EntityTooSmall	The entity size is below the minimum limit.	400
FileGroupTooLarge	The total file group size exceeds the maximum limit.	400
FilePartNotExist	No such part exists.	400
FilePartStale	The part has expired.	400
InvalidArgument	The parameter format is invalid.	400
InvalidAccessKeyId	No such AccessKeyId exists.	403
InvalidBucketName	The bucket name is invalid.	400
InvalidDigest	The digest is invalid.	400

Error Code	Description	HTTP status code
InvalidObjectName	The object name is invalid.	400
InvalidPart	The part is invalid.	400
InvalidPartOrder	The part order is invalid.	400
InvalidTargetBucketForLogging	The buckets that store log files are invalid.	400
InternalError	Internal OSS error	500
MalformedXML	The XML format is invalid.	400
MethodNotAllowed	The method is not allowed.	405
MissingArgument	Parameters are not configured.	411
MissingContentLength	The content length is not configured.	411
NoSuchBucket	No such bucket exists.	404
NoSuchKey	No such object exists.	404
NoSuchUpload	No such uploadId exists.	404
NotImplemented	The methods are not implemented.	501
PreconditionFailed	Precondition error	412
RequestTimeTooSkewed	The local time set for OSSClient is deviated from the time set for the OSS server by over 15 minutes.	403
RequestTimeout	Request timeout	400
SignatureDoesNotMatch	Signature mismatch	403
InvalidEncryptionAlgorithmError	The specified encryption algorithm (the entropy encoding type) is invalid.	400

2.23 FAQ

Jar conflicts

- Cause analysis

If the following errors occur when you use OSS Java SDK, jar conflicts may exist in your project.

```
Exception in thread "main" java.lang.NoClassDef
FoundError : org/apache/http/ssl/TrustStrat egy
    at com.aliyun.oss.OSSClient.<init>(OSSClient.java
: 268 )
    at com.aliyun.oss.OSSClient.<init>(OSSClient.java
: 193 )
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS .
java : 77 )
Caused by : java.lang.ClassNotFo undExcepti on : org .
apache . http . ssl . TrustStrat egy
    at java.net.URLClassLoader $ 1 . run (URLClassLo ader
. java : 366 )
    at java.net.URLClassLo ader $ 1 . run (URLClassLo ader
. java : 355 )
    at java.security.AccessCont roller . doPrivileg ed (
Native Method )
    at java.net.URLClassLo ader . findClass (URLClassLo
ader . java : 354 )
    at java.lang.ClassLoade r . loadClass (ClassLoade r .
java : 425 )
    at sun.misc.Launcher $ AppClassLo ader . loadClass (
Launcher . java : 308 )
    at java.lang.ClassLoade r . loadClass (ClassLoade r .
java : 358 )
    ... 3 more
```

Or

```
Exception in thread "main" java.lang.NoSuchFiel dError
: INSTANCE
    at org.apache.http.impl.io.DefaultHtt pRequestWr
iterFactor y.<init>(DefaultHtt pRequestWr iterFactor y .
java : 52 )
    at org.apache.http.impl.io.DefaultHtt pRequestWr
iterFactor y.<init>(DefaultHtt pRequestWr iterFactor y .
java : 56 )
    at org.apache.http.impl.io.DefaultHtt pRequestWr
iterFactor y.<clinit>(DefaultHtt pRequestWr iterFactor y .
java : 46 )
    at org.apache.http.impl.conn.ManagedHtt pClientCon
nectionFac tory.<init>(ManagedHtt pClientCon nectionFac
tory . java : 82 )
    at org.apache.http.impl.conn.ManagedHtt pClientCon
nectionFac tory.<init>(ManagedHtt pClientCon nectionFac
tory . java : 95 )
    at org.apache.http.impl.conn.ManagedHtt pClientCon
nectionFac tory.<init>(ManagedHtt pClientCon nectionFac
tory . java : 104 )
```

```

at org.apache.http.impl.conn.ManagedHttpClientConnectionFactory.<clinit>(ManagedHttpClientConnectionFactory.java:62)
at org.apache.http.impl.conn.PoolingHttpClientConnectionFactory.<init>(PoolingHttpClientConnectionFactory.java:572)
at org.apache.http.impl.conn.PoolingHttpClientConnectionFactory.<init>(PoolingHttpClientConnectionFactory.java:174)
at org.apache.http.impl.conn.PoolingHttpClientConnectionFactory.<init>(PoolingHttpClientConnectionFactory.java:158)
at org.apache.http.impl.conn.PoolingHttpClientConnectionFactory.<init>(PoolingHttpClientConnectionFactory.java:149)
at org.apache.http.impl.conn.PoolingHttpClientConnectionFactory.<init>(PoolingHttpClientConnectionFactory.java:125)
at com.aliyun.oss.common.comm.DefaultServiceClient.createHttpClientManager(DefaultServiceClient.java:237)
at com.aliyun.oss.common.comm.DefaultServiceClient.<init>(DefaultServiceClient.java:78)
at com.aliyun.oss.OSSClient.<init>(OSSClient.java:268)
at com.aliyun.oss.OSSClient.<init>(OSSClient.java:193)
at OSSManagerImpl.upload(OSSManagerImpl.java:42)
at OSSManagerImpl.main(OSSManagerImpl.java:63)

```

The cause is that OSS Java SDK uses Apache httpclient 4.4.1, while your project uses Apache httpclient or commons-httpclient jar that conflicts with Apache httpclient 4.4.1. To view the jar file and its version used for a project, open OSS Java SDK and run the `mvn dependency:tree` command. The following figure shows that a project that uses Apache httpclient 4.3.

```

[INFO] --- maven-dependency-plugin:2.2:tree (default-cli) @ maven-demo ---
[INFO] com.aliyun.oss:maven-demo:jar:0.1.1-SNAPSHOT
[INFO] +- junit:junit:jar:4.10:test
[INFO] | \- org.hamcrest:hamcrest-core:jar:1.1:test
[INFO] +- org.apache.httpcomponents:httpclient:jar:4.3:compile
[INFO] | +- org.apache.httpcomponents:httpcore:jar:4.3:compile
[INFO] | +- commons-logging:commons-logging:jar:1.1.3:compile
[INFO] | \- commons-codec:commons-codec:jar:1.6:compile
[INFO] \- com.aliyun.oss:aliyun-sdk-oss:jar:2.2.1:compile
[INFO]     +- org.jdom:jdom:jar:1.1:compile
[INFO]     \- net.sf.json-lib:json-lib:jar:jdk15:2.4:compile
[INFO]         +- commons-beanutils:commons-beanutils:jar:1.8.0:compile
[INFO]         +- commons-collections:commons-collections:jar:3.2.1:compile
[INFO]         +- commons-lang:commons-lang:jar:2.5:compile
[INFO]         \- net.sf.ezmorph:ezmorph:jar:1.0.6:compile

```

- There are two solutions to jar conflicts.
 - Use a unified version. If your project uses the version that conflicts with Apache httpclient 4.4.1, use Apache httpclient 4.4.1 and delete Apache httpclient

dependencies of other versions in the pom.xml file. If your project uses commons-httpclient, jar conflicts may also exist. In this case, delete commons-httpclient.

- Resolve dependency conflicts. If your project uses multiple third-party dependencies that have various Apache httpclient versions, dependency conflicts may exist in your project. Use exclusion to resolve the conflicts. For more information, see [Maven Guides](#).

OSS Java SDK is dependent on the following jar versions. Conflict solutions are similar to the httpclient method.

```
[INFO] com.aliyun.oss:maven-demo:jar:0.1.1-SNAPSHOT
[INFO] +- junit:junit:jar:4.10:test
[INFO] | \- org.hamcrest:hamcrest-core:jar:1.1:test
[INFO] \- com.aliyun.oss:aliyun-sdk-oss:jar:2.2.1:compile
[INFO]     +- org.apache.httpcomponents:httpclient:jar:4.4.1:compile
[INFO]     | +- org.apache.httpcomponents:httpcore:jar:4.4.1:compile
[INFO]     | +- commons-logging:commons-logging:jar:1.2:compile
[INFO]     | \- commons-codec:commons-codec:jar:1.9:compile
[INFO]     +- org.jdom:jdom:jar:1.1:compile
[INFO]     \- net.sf.json-lib:json-lib:jar:jdk15:2.4:compile
[INFO]         +- commons-beanutils:commons-beanutils:jar:1.8.0:compile
[INFO]         +- commons-collections:commons-collections:jar:3.2.1:compile
[INFO]         +- commons-lang:commons-lang:jar:2.5:compile
[INFO]         \- net.sf.ezmorph:ezmorph:jar:1.0.6:compile
```

Missing files

- Cause analysis

If the following errors occur when you use OSS Java SDK, your project may lack the necessary files for OSS Java SDK compilation and execution.

```
Exception in thread "main" java.lang.NoClassDef
FoundError: org/apache/http/auth/Credentials
    at com.aliyun.oss.OSSClient.<init>(OSSClient.
java : 268 )
    at com.aliyun.oss.OSSClient.<init>(OSSClient.
java : 193 )
    at com.aliyun.oss.demo.HelloOSS.main(HelloOSS
.java : 76 )
Caused by: java.lang.ClassNotFoundException: org.
apache.http.auth.Credentials
    at java.net.URLClassLoader$1.run(URLClassLo
ader.java : 366 )
    at java.net.URLClassLoader$1.run(URLClassLo
ader.java : 355 )
    at java.security.AccessController.doPrivileged
(Native Method)
    at java.net.URLClassLoader.findClass(
URLClassLoader.java : 354 )
    at java.lang.ClassLoader.loadClass(ClassLoade
r.java : 425 )
```

```

        at sun.misc.Launcher$AppClassLoader.loadClass (
Launcher.java : 308 )
        at java.lang.ClassLoader.loadClass (ClassLoader
r.java : 358 )
        ... 3 more

```

Or

```

Exception in thread "main" java.lang.NoClassDef
FoundError : org/apache/http/protocol/HttpContext
        at com.aliyun.oss.OSSClient.<init>(OSSClient.
java : 268 )
        at com.aliyun.oss.OSSClient.<init>(OSSClient.
java : 193 )
        at com.aliyun.oss.demo.HelloOSS.main (HelloOSS
.java : 76 )
Caused by : java.lang.ClassNotFoundException : org.
apache.http.protocol.HttpContext
        at java.net.URLClassLoader$1.run (URLClassLo
ader.java : 366 )
        at java.net.URLClassLoader$1.run (URLClassLo
ader.java : 355 )
        at java.security.AccessController.doPrivileged
(Native Method)
        at java.net.URLClassLoader.findClass (
URLClassLoader.java : 354 )
        at java.lang.ClassLoader.loadClass (ClassLoader
r.java : 425 )
        at sun.misc.Launcher$AppClassLoader.loadClass (
Launcher.java : 308 )
        at java.lang.ClassLoader.loadClass (ClassLoader
r.java : 358 )
        ... 3 more

```

Or

```

Exception in thread "main" java.lang.NoClassDef
FoundError : org/jdom/input/SAXBuilder
        at com.aliyun.oss.internal.ResponseParsers.
getXmlRootElement (ResponseParsers.java : 645 )
        at ...
        at com.aliyun.oss.OSSClient.doesBucketExist (
OSSClient.java : 471 )
        at com.aliyun.oss.OSSClient.doesBucketExist (
OSSClient.java : 465 )
        at com.aliyun.oss.demo.HelloOSS.main (HelloOSS
.java : 82 )
Caused by : java.lang.ClassNotFoundException : org.
jdom.input.SAXBuilder
        at java.net.URLClassLoader$1.run (URLClassLo
ader.java : 366 )
        at java.net.URLClassLoader$1.run (URLClassLo
ader.java : 355 )
        at java.security.AccessController.doPrivileged
(Native Method)
        at java.net.URLClassLoader.findClass (
URLClassLoader.java : 354 )
        at java.lang.ClassLoader.loadClass (ClassLoader
r.java : 425 )

```



```

        at sun.misc.Launcher$AppClassLoader.loadClass (
Launcher.java : 308 )
        at java.lang.ClassLoader.loadClass (ClassLoader
.java : 358 )
        ... 11 more

```

Dependencies of OSS Java SDK are as follows:

- aliyun-sdk-oss-2.2.1.jar
- hamcrest-core-1.1.jar
- jdom-1.1.jar
- commons-codec-1.9.jar
- httpclient-4.4.1.jar
- commons-logging-1.2.jar
- httpcore-4.4.1.jar
- log4j-1.2.15.jar

In the preceding dependencies, all files except the log4j-1.2.15.jar file are required . You need to add this file to enable access logging.

• Solutions

Add OSS Java SDK dependencies to your project. Methods:

- If your project is in Eclipse, see [Method 2: Import a JAR package](#) to your Eclipse project in Java SDK.
- If your project is in Ant, add OSS Java SDK dependencies to the lib directory.
- If you directly use the .javac or .java files, run the `-classpath` command or the `-cp` command to specify the path where OSS Java SDK dependencies are stored or save the Java SDK dependencies to classpath.

Connection timeout

• Cause analysis

If the following errors occur when the OSS Java SDK program is run, possible causes are endpoint errors or unavailable networks:

```

com.aliyun.oss.ClientException: SocketException
    at com.aliyun.oss.common.utils.ExceptionFactory.createNetworkException (ExceptionFactory.java : 71 )
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore (DefaultServiceClient.java : 116 )
    at com.aliyun.oss.common.comm.ServiceClient.sendRequestImpl (ServiceClient.java : 121 )

```

```

    at com.aliyun.oss.common.comm.ServiceClient.sendRequest (ServiceClient.java:67)
    at com.aliyun.oss.internal.OSSOperation.send (OSSOperation.java:92)
    at com.aliyun.oss.internal.OSSOperation.doOperation (OSSOperation.java:140)
    at com.aliyun.oss.internal.OSSOperation.doOperation (OSSOperation.java:111)
    at com.aliyun.oss.internal.OSSBucketOperation.getBucketInfo (OSSBucketOperation.java:1152)
    at com.aliyun.oss.OSSClient.getBucketInfo (OSSClient.java:1220)
    at com.aliyun.oss.OSSClient.getBucketInfo (OSSClient.java:1214)
    at com.aliyun.oss.demo.HelloOSS.main (HelloOSS.java:94)
Caused by: org.apache.http.conn.HttpHostConnectException: Connect to oss-test.oss-cn-hangzhou-internal.aliyuncs.com:80 [oss-test.oss-cn-hangzhou-internal.aliyuncs.com/10.84.135.99] failed: Connection timed out: connect
    at org.apache.http.impl.conn.DefaultHttpClientConnectionOperator.connect (DefaultHttpClientConnectionOperator.java:151)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.connect (PoolingHttpClientConnectionManager.java:353)
    at org.apache.http.impl.execchain.MainClientExec.establishRoute (MainClientExec.java:380)
    at org.apache.http.impl.execchain.MainClientExec.execute (MainClientExec.java:236)
    at org.apache.http.impl.execchain.ProtocolExec.execute (ProtocolExec.java:184)
    at org.apache.http.impl.execchain.RedirectExec.execute (RedirectExec.java:110)
    at org.apache.http.impl.client.InternalHttpClient.doExecute (InternalHttpClient.java:184)
    at org.apache.http.impl.client.CloseableHttpClient.execute (CloseableHttpClient.java:82)
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore (DefaultServiceClient.java:113)
    ... 9 more

```

• Solution

You can use [ossutil](#) for fast troubleshooting.

org.apache.http.NoHttpResponseException: The target server failed to respond

- Cause analysis

The following error occurs when the OSS Java SDK program is run:

```
com.aliyun.oss.ClientException: Unknown
    at com.aliyun.oss.common.utils.ExceptionFactory.createNetworkException(ExceptionFactory.java:68) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore(DefaultServiceClient.java:115) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.common.comm.ServiceClient.sendRequestImpl(ServiceClient.java:121) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.common.comm.ServiceClient.sendRequest(ServiceClient.java:67) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSOperation.send(OSSOperation.java:92) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSOperation.doOperation(OSSOperation.java:140) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSOperation.doOperation(OSSOperation.java:111) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.aliyun.oss.internal.OSSMultipartOperation.initiateMultipartUpload(OSSMultipartOperation.java:206) ~[aliyun-sdk-oss-2.1.0.jar:na]
    at com.taobao.agoo.dump.client.OSSTools.multipartUpload(OSSTools.java:79) ~[agoo-dump-client-2.0.0-SNAPSHOT.jar:na]
    at com.taobao.agoo.dump.biz.manager.TaskExecutorManager$UploadTask.run(TaskExecutorManager.java:114) ~[agoo-dump-biz-2.0.0-SNAPSHOT.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:471) [na:1.7.0_51]
    at java.util.concurrent.FutureTask.run(FutureTask.java:262) [na:1.7.0_51]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145) [na:1.7.0_51]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615) [na:1.7.0_51]
    at java.lang.Thread.run(Thread.java:744) [na:1.7.0_51]
Caused by: org.apache.http.NoHttpResponseException: The target server failed to respond
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:143) ~[httpclient-4.4.jar:4.4]
    at org.apache.http.impl.conn.DefaultHttpResponseParser.parseHead(DefaultHttpResponseParser.java:57) ~[httpclient-4.4.jar:4.4]
    at org.apache.http.impl.io.AbstractMessageParser.parse(AbstractMessageParser.java:261) ~[httpcore-4.4.jar:4.4]
    at org.apache.http.impl.DefaultBHttpClientConnection.receiveResponseHeader(DefaultBHttpClientConnection.java:165) ~[httpcore-4.4.jar:4.4]
    at org.apache.http.impl.conn.CPoolProxy.receiveResponseHeader(CPoolProxy.java:167) ~[httpclient-4.4.jar:4.4]
    at org.apache.http.protocol.HttpRequestExecutor.doReceiveResponse(HttpRequestExecutor.java:272) ~[httpcore-4.4.jar:4.4]
```

The preceding error may occur if expired connections are used. This error occurs only when an earlier version of Java SDK 2.1.2 is used.

- Solutions

Upgrade OSS Java SDK to 2.1.2 or a later version.

OSS Java SDK stops responding when the Java program is called

- Cause analysis

OSS Java SDK stops responding when the Java program is called. Run the `jstack`

– `l pid` command to view the stacks. The exception is as follows:

```
" main " prio = 6   tid = 0x00000000 0291e000   nid = 0xc40
waiting on condition [ 0x00000000 02dae000 ]
java . lang . Thread . State :  WAITING  ( parking )
    at  sun . misc . Unsafe . park ( Native Method )
        - parking to wait for < 0x00000007 d85697f8 > ( a
java . util . concurrent . locks . AbstractQu euedSynchr onizer $
Condition0 bject )
    at  java . util . concurrent . locks . LockSupport . park (
LockSupport . java : 186 )
    at  java . util . concurrent . locks . AbstractQu euedSynchr
onizer $ Condition0 bject . await ( AbstractQu euedSynchr onizer
. java : 2043 )
    at  org . apache . http . pool . PoolEntryF uture . await (
PoolEntryF uture . java : 138 )
    at  org . apache . http . pool . AbstractCo nnPool .
getPoolEnt ryBlocking ( AbstractCo nnPool . java : 306 )
    at  org . apache . http . pool . AbstractCo nnPool . access $
000 ( AbstractCo nnPool . java : 64 )
    at  org . apache . http . pool . AbstractCo nnPool $ 2 .
getPoolEnt ry ( AbstractCo nnPool . java : 192 )
    at  org . apache . http . pool . AbstractCo nnPool $ 2 .
getPoolEnt ry ( AbstractCo nnPool . java : 185 )
    at  org . apache . http . pool . PoolEntryF uture . get (
PoolEntryF uture . java : 107 )
```

```

    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager.leaseConnection (PoolingHttpClientConnectionManager.java:276)
    at org.apache.http.impl.conn.PoolingHttpClientConnectionManager$1.get (PoolingHttpClientConnectionManager.java:263)
    at org.apache.http.impl.execchain.MainClientExec.execute (MainClientExec.java:190)
    at org.apache.http.impl.execchain.ProtocolExec.execute (ProtocolExec.java:184)
    at org.apache.http.impl.execchain.RedirectExec.execute (RedirectExec.java:110)
    at org.apache.http.impl.client.InternalHttpClient.doExecute (InternalHttpClient.java:184)
    at org.apache.http.impl.client.CloseableHttpClient.execute (CloseableHttpClient.java:82)
    at com.aliyun.oss.common.comm.DefaultServiceClient.sendRequestCore (DefaultServiceClient.java:113)
    at com.aliyun.oss.common.comm.ServiceClient.sendRequestImpl (ServiceClient.java:123)
    at com.aliyun.oss.common.comm.ServiceClient.sendRequest (ServiceClient.java:68)
    at com.aliyun.oss.internal.OSSOperation.send (OSSOperation.java:94)
    at com.aliyun.oss.internal.OSSOperation.doOperation (OSSOperation.java:146)
    at com.aliyun.oss.internal.OSSOperation.doOperation (OSSOperation.java:113)
    at com.aliyun.oss.internal.OSSObjectOperation.getObject (OSSObjectOperation.java:229)
    at com.aliyun.oss.OSSClient.getObject (OSSClient.java:629)
    at com.aliyun.oss.OSSClient.getObject (OSSClient.java:617)
    at samples.HelloOSS.main (HelloOSS.java:49)

```

The cause is due to the connection leak in the connection pool (possibly because `ossObject` is left unclosed after it is used).

• Solutions

Check your program and ensure that no connection leak occurs. Use the following code to close `ossObject`:

```

// Read an object .
OSSObject ossObject = ossClient . getObject ( bucketName ,
objectName );
// Perform operations on OSS .
// Close your ossObject .
ossObject . close ();

```

Connection closure

- Cause analysis

If the following errors occur when you use `ossClient.getObject`:

```
Exception in thread "main" org.apache.http.
ConnectionClosedException: Premature end of Content
- Length delimited message body ( expected : 11990526 ;
received : 202880
    at org.apache.http.impl.io.ContentLengthInputStream.
    read (ContentLengthInputStream.java : 180 )
    at org.apache.http.impl.io.ContentLengthInputStream.
    read (ContentLengthInputStream.java : 200 )
    at org.apache.http.impl.io.ContentLengthInputStream.
    close (ContentLengthInputStream.java : 103 )
    at org.apache.http.impl.execchain.ResponseEntityProxy.
    streamClosed (ResponseEntityProxy.java : 128 )
    at org.apache.http.conn.EofSensorInputStream.
    checkClose (EofSensorInputStream.java : 228 )
    at org.apache.http.conn.EofSensorInputStream.
    close (EofSensorInputStream.java : 174 )
    at java.io.FilterInputStream.close (FilterInput
    tStream.java : 181 )
    at java.io.FilterInputStream.close (FilterInput
    tStream.java : 181 )
    at com.aliyun.oss.event.ProgressInputStream.close
    (ProgressInputStream.java : 147 )
    at java.io.FilterInputStream.close (FilterInput
    tStream.java : 181 )
    at samples.HelloOSS.main (HelloOSS.java : 39 )
```

The cause is that the time between two consecutive data reading exceeds one minute. OSS will close the connections if the time between two consecutive data reading exceeds one minute.

- Solution

If you only need to read part of the data during unfixed periods of time, use [Range download](#) to avoid connection closure during data reading.

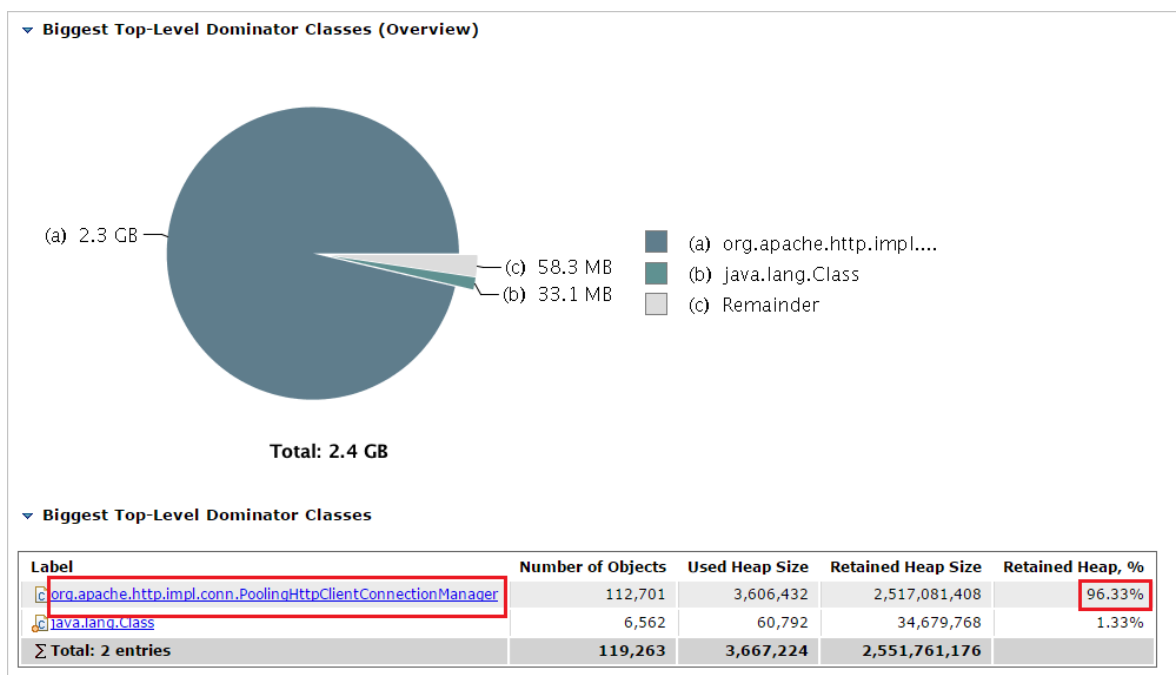
Memory leaks

- Cause analysis

Memory leaks occur when the OSS Java SDK program is run for a period of time (from several hours to several days based on access traffic). We recommend that

you use [Eclipse Memory Analyzer \(MAT\)](#) to analyze memory usage. For more information about usage methods, see [Use MAT for Head Dump File Analysis](#).

If the analysis result is similar to the following figure (PoolingHttpClientConnectionManager occupies 96 percent of memory), the cause may be that new OSSClient is run multiple times while `ossClient.shutdown` is not called. Consequently, memory leaks occur.



- a

Ensure that `ossClient.shutdown` is enabled after `OSSClient` is enabled.

`InterruptedException` occurs when `ossClient.shutdown` is called

- Error Cause

The following exception is thrown when `OSSClient.shutdown` is called:

```
java . lang . InterruptedException : sleep interrupted
    at java . lang . Thread . sleep ( Native Method )
    at com . aliyun . oss . common . comm . IdleConnectionReaper . run ( IdleConnectionReaper : 76 )
```

The cause is that the backend thread (`IdleConnectionReaper`) of `ossClient` will close connections periodically. If `ossClient.shutdown` is called when `IdleConnectionReaper` is in the Sleep mode, the preceding exception occurs.

- **Solution**

OSS Java SDK 2.3.0 has rectified this error.

For versions earlier than OSS Java SDK 2.3.0, use the following code to ignore the error:

```
try {
    ossClient.shutdown();
} catch (Exception e) {
}
```

Other errors

For more information about how to rectify other errors returned by OSS, see [Errors and troubleshooting](#).

3 Python

3.1 Preface

Source code

For the source code, see [GitHub](#).

Code samples

OSS Python SDK provides a variety of code samples for your reference or use. The following table describes the content of code samples.

Sample file	Content
object_basic.py	Quick start , including creating a bucket, and uploading, downloading, listing, and deleting objects
object_extra.py	Upload and manage objects, including managing Object , copying objects , append upload
upload.py	Upload objects, including resumable upload and multipart upload
download.py	Download objects, including streaming download , range download , resumable download
object_check.py	Data verification methods during uploads and downloads, including MD5 and CRC
object_progress.py	Upload progress bars and download progress bars
object_callback.py	Upload callback in object upload
object_post.py	Operations related to form upload
sts.py	STS usage, including obtaining the temporary user key for playing a role and using the temporary user key to access the OSS
image.py	Operations related to image processing

Sample file	Content
bucket.py	Bucket management , including creating, deleting, and listing buckets, configuring static website hosting , and configuring lifecycle rules .

3.2 Installation

Environment requirements

This version of Python SDK is applicable to Python 2.6, 2.7, 3.3, 3.4 and 3.5.

- Installation environment

Follow the instructions on [python official website](#) to install an appropriate version of Python.

- Check Python version

Run `python - V` to check the Python version.

In Windows, if a “Not an internal or external command” message prompts, edit the `Path` in environment variables to add the installation path of Python and pip. In general, pip is installed in the Script directory located in the installation path of Python.



Note:

You may need to restart your computer make sure the environment variable takes effect.

Download SDK

- [Download from GitHub](#)
- [Download previous versions](#)

For more information, see [OSS API documents](#).

Install Python-devel

Python SDK uses the crcmod library which depends on the Python.h to caculate CRC . Therefore, if the Python.h file is missing, Python SDK can be successfully installed while the installation of crcmod C extension mode fails, which significantly reduces the efficiency of operations, such as upload and download. Therefore, you must install the python-devel package if it is missing.

- You do not need to install the python-devel package in Windows and Mac OS X because header files that Python depends on are installed together with Python.
- For CentOS, RHEL, and Fedora, run the following command to install python-devel:

```
yum install python - devel
```

- For Debian and Ubuntu, run the following command to install python-devel:

```
apt - get install python - dev
```

Install SDK

- Install SDK by Pip

Run the following command:

```
pip install oss2
```



Note:

If Pip is not installed in your environment, see [Pip official website](#) to install Pip.

- Install SDK by source code

Download the SDK package of an appropriate version from [GitHub](#) and decompress the package. Make sure that the setup.py file exists in the directory and run the following command:

```
python setup . py install
```

Verification

- Verify the SDK version
 1. Input `python` in the command line and press Enter to enter the Python environment.
 2. Run the following command to verify the SDK version:

```
>>> import oss2
>>> oss2 . __version__ _
' 2 . 0 . 0 '
```

If the preceding information is displayed, it means that you have successfully installed the Python SDK 2.0.0.

- Verify the crcmod

Two methods are available for crcmod to calculate CRC: the C extension mode (in which crcmod calls the extension library `_crcfunext.so` to calculate CRC) and the Python-only mode. The performance of C extension mode is far better than the Python-only mode. The crcmod library is automatically installed when oss2 is installed. For more information about crcmod, see [crcmod introduction](#).

If you find that the speed of upload and download interfaces is significantly slow compared with other tools (such as [ossutil](#)) or other SDKs after the Python SDK is installed, it is probably because that the C extension mode of crcmod library is not installed successfully, resulting that the Python-only mode is used to calculate CRC in upload and download.

You can verify whether the C extension mode of crcmode is successfully installed as follows:

1. Input `python` in the command line and press Enter to enter the Python environment.
2. Input `import crcmod ._crcfunext` and press Enter.
 - If no exception is reported, it indicates that the C extension mode has been successfully installed.
 - If the following message is displayed, it indicates that the C extension mode is not successfully installed.

```
>>> import crcmod ._crcfunext
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named _crcfunext
```

The error occurs because the `Python.h` file that `_crcfunext.so` depends on is missing when the crcmod is compiled, resulting in that the `_crcfunext.so` fails to be generated. In this case, CRC is calculated in the Python-only method.

The SDK is installed successfully. However, the efficiency of operations such as upload and download is low.

Perform the following operations if the problem occurs.

a. Run the following command to uninstall crcmod:

```
pip uninstall crcmod
```

b. [Install python-devel](#).

c. Run the following command to install crcmod again:

```
pip install crcmod
```

If the installation still fails, uninstall crcmod and run the following command to check detailed information:

```
pip install crcmod -v
```

Uninstall SDK

If the SDK fails to be installed, we recommend you to uninstall the SDK using pip and install it again. You can run the following command to uninstall the SDK.

```
pip uninstall oss2
```

3.3 Quick start

This topic describes how to use OSS Python SDK to perform routine operations such as bucket creation, object uploads, and object downloads.

Create a bucket

Run the following code to create a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
# the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')
```

```
# Set the ACL of the bucket to private .
bucket . create_bucket ( oss2 . models . BUCKET_ACL_PRIVATE )
```

For more information about endpoints, see [Regions and endpoints](#).

Upload objects

Run the following code to upload a file to OSS:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
.aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# < yourLocalFile > consists of a local file path and
a file name with extension , for example : / users / local
/ myfile . txt
bucket . put_object_from_file ('< yourObject Name >', '<
yourLocalFile >')
```

Download objects

Run the following code to download a specified object to a local file:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
.aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# < yourLocalFile > consists of a local file path and
a file name with extension , for example : / users / local
/ myfile . txt
```

```
bucket . get_object _to_file ('< yourObject Name >', '< yourLocalF
ile >')
```

List objects

Run the following code to list 10 objects in a specified bucket:

```
# -*- coding : utf - 8 -*-
import oss2
from itertools import islice

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We strongly
recommend that you create a RAM account and use it
for API access and daily O & M . Log on to https
:// ram . console . aliyun . com to create a RAM account .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# oss2 . ObjectIter ator is used to traverse objects .
for b in islice ( oss2 . ObjectIter ator ( bucket ), 10 ):
    print ( b . key )
```

Delete objects

Run the following code to delete a specified object:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . delete_obj ect ('< yourObject Name >')
```

3.4 Initialization

This topic describes how to initialize OSS Python SDK.

Most operations in OSS Python SDK are performed through `oss2.Service` and `oss2.Bucket`.

- The `oss2.Service` class is used to list buckets.
- The `oss2.Bucket` class is used to upload, download, and delete objects, and configure buckets.

To initialize the two classes, you must specify an endpoint. The `oss2.Service` class does not support access with the custom domain (CNAME). For more information about endpoints, see [Regions and endpoints](#) and [Bind a custom domain name](#).

Initialize the `oss2.Service` class

For more information, see the bucket listing part in [Manage buckets](#).

Initialize the `oss2.Bucket` class.

- Use a OSS domain name to initialize the class

Run the following code to initialize the `oss2.Bucket` class with a OSS domain name.

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram .
# console . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This case uses the Hangzhou endpoint as an
# example . Fill in the region endpoint name according
# to the actual circumstan ces .
endpoint = ' http :// oss - cn - hangzhou . aliyuncs . com '

bucket = oss2 . Bucket ( auth , endpoint , '< yourBucket Name
>')
```

- Use a custom domain name to initialize the class

Run the following code to initialize the `oss2.Bucket` class with a custom domain name

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram .
# console . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
```

```
# Use a custom domain name : my - domain . com as an
example . is_cname = True indicates that the CNAME is
enabled . Cname means binding a custom domain name
to the storage space . CNAME indicates a custom
domain bound to a bucket .
cname = ' http :// my - domain . com '
bucket = oss2 . Bucket ( auth , cname , '< yourBucket Name >',
is_cname = True )
```

- **Set connection timeout**

Run the following code to set connection timeout:

```
# -*- coding : utf - 8 -*-
import oss2

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
// This example uses endpoint China ( Hangzhou ) . Specify
the actual endpoint based on your requiremen ts .
endpoint = ' http :// oss - cn - hangzhou . aliyuncs . com '

# Set the connection timeout to 30 seconds .
bucket = oss2 . Bucket ( auth , endpoint , '< yourBucket Name
>', connect_ti meout = 30 )
```

- **Disable CRC verification**

CRC verification is enabled by default during uploads and downloads to ensure data integrity during uploads and downloads. Run the following code to disable CRC verification:



Warning:

We recommended that you do not disable CRC verification. If you disable CRC verification, the data integrity during uploads and downloads is not guaranteed.

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram .
console . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
// This example uses endpoint China ( Hangzhou ) . Specify
the actual endpoint based on your requiremen ts .
```



```
endpoint = 'http://oss-cn-hangzhou.aliyuncs.com'

bucket = oss2.Bucket(auth, endpoint, '<yourBucket Name>', enable_crc = False)
```

3.5 Bucket

A bucket serves as a container that stores objects. Objects belong to a bucket.

Create a bucket

Run the following code to create a bucket:

```
# -*- coding: utf-8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS. We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenance. To
# create a RAM account, log on to https://ram.console
# .aliyun.com.
auth = oss2.Auth('<yourAccessKeyId>', '<yourAccessKeySecret>')
# You can create a new bucket in a specified region
# by specifying the endpoint and bucket name. This
# example uses endpoint China (Hangzhou). Specify the
# actual endpoint based on your requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.aliyuncs.com', '<yourBucket Name>')

# The default storage class and ACL of the created
# bucket are set to standard and private read
# respectively.
bucket.create_bucket()
```

For more information about bucket naming rules, see [naming conventions](#) in [Basic concepts](#).

You can specify the ACL and the storage class when you create a bucket, as shown in [Bucket-level permissions](#) and [Storage class](#). The sample code is as follows:

```
# Set the bucket ACL to public read and set the
# storage class to Infrequent Access.
bucket.create_bucket(oss2.BUCKET_ACL_PUBLIC_READ, oss2
    .models.BucketCreateConfig(oss2.BUCKET_STORAGE_CLASS_IA))
```

List buckets

Run the following code to list buckets:

```
# -*- coding: utf-8 -*-
import oss2
```

```
# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance. To
create a RAM account, log on to https://ram.console
.aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China (Hangzhou). Specify
the actual endpoint based on your requirements.
service = oss2.Service(auth, 'http://oss-cn-hangzhou.
aliyuncs.com')

print([b.name for b in oss2.BucketIterator(service)])
```

Configure an ACL for a bucket

The ACL of a bucket includes the following permissions.

Permission	Description	Value
Private	The bucket owner and the authorized users can read and write objects in the bucket. Other users cannot perform any operation on the objects.	oss2.BUCKET_ACL_PRIVATE
Public read	The bucket owner and the authorized users can read and write objects in the bucket. Other users can only read the objects in the bucket. Authorize this permission with caution.	oss2.BUCKET_ACL_PUBLIC_READ
Public read-write	All users can read and write objects in the bucket. Authorize this permission with caution.	oss2.BUCKET_ACL_PUBLIC_READ_WRITE

Run the following code to configure an ACL for a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance. To
create a RAM account, log on to https://ram.console
.aliyun.com.
```

```
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Set the ACL for the bucket to private .
bucket . put_bucket _acl ( oss2 . BUCKET_ACL _PRIVATE )
```

Obtain the ACL for a bucket

Run the following code to obtain the ACL for a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

print ( bucket . get_bucket _acl (). acl )
```

Obtain bucket information

Run the following code to obtain bucket information:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Obtain bucket informatio n
bucket_info = bucket . get_bucket _info ()
print (' name : ' + bucket_info . name )
print (' storage class : ' + bucket_info . storage_cl ass )
print (' creation date : ' + bucket_info . creation_d ate )
print (' intranet_e ndpoint : ' + bucket_info . intranet_e
ndpoint )
```

```
print (' extranet_endpoint ' + bucket_info . extranet_endpoint )
print (' owner : ' + bucket_info . owner . id )
print (' grant : ' + bucket_info . acl . grant )
```

Delete a bucket

Before a bucket is deleted, ensure that all objects in the bucket and fragments that are generated from multipart upload are deleted.



Note:

To delete the fragments that are generated from multipart upload, use [Bucket.ListMultipartUploads](#) to list all fragments, and then use [Bucket.AbortMultipartUpload](#) to delete the fragments.

Run the following code to delete a bucket:

```
# -*- coding : utf - 8 -*-
import oss2
# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyunos . com ', '< yourBucket Name >')
try :
    # Delete the bucket .
    bucket . delete_bucket ()
except oss2 . exceptions . BucketNotEmpty :
    print (' bucket is not empty .')
except oss2 . exceptions . NoSuchBucket :
    print (' bucket does not exist ')
```

For buckets which are not empty, you can list and delete objects stored in these buckets first (stop uploading for objects uploaded by multipart upload), and then delete the buckets.

3.6 Single-connection bandwidth throttling

This topic describes how to ensure other applications have sufficient bandwidth when you perform upload and download operations on OSS. You can include the `x-oss-traffic-limit` parameter in your requests and set bandwidth limits.



Note:

For more information about the scenarios and precautions of the single-connection bandwidth throttling feature, see [#unique_47](#) in the OSS Developer Guide.

Configure bandwidth throttling for common uploads and downloads

Use the following code to configure bandwidth throttling when you upload and download objects to and from OSS:

```
# -*- coding : utf - 8 -*-

import oss2
from oss2 . models import OSS_TRAFFI C_LIMIT

# Security risks may arise if you log on with the
# AccessKey pair of an Alibaba Cloud account because
# the account has permission s on all API operations
# in OSS . We recommend that you log on with a
# RAM user account to call API operations or perform
# routine operations and maintenanc e . To create a RAM
# user account , log on to https :// ram . console . aliyun .
# com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses China ( Hangzhou ) as the endpoint .
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourReques tBucketNam e >')

object_nam e = '< yourobject Name >'
local_file _name = '< yourLocalF ileName >'
down_file_ name = '< yourDownlo adFileName >'

# Set the bandwidth limit to 100 KB / s ( 819200 bit /
# s ) in headers .
limit_spee d = ( 100 * 1024 * 8 )
headers = dict ()
headers [ OSS_TRAFFI C_LIMIT ] = str ( limit_spee d );

# Configure bandwidth throttling for the object to be
# uploaded .
result = bucket . put_object _from_file ( object_nam e ,
local_file _name , headers = headers )
print (' http response status :', result . status )

# Configure bandwidth throttling for the object to be
# downloaded .
result = bucket . get_object _to_file ( object_nam e ,
down_file_ name , headers = headers )
```

```
print (' http response status :', result . status )
```

Configure bandwidth throttling for uploads and downloads that use signed URLs

Use the following code to configure bandwidth throttling when you upload and download objects from and to OSS by using signed URLs.

```
# -*- coding : utf - 8 -*-

import oss2
from oss2 . models import OSS_TRAFFI C_LIMIT

# Security risks may arise if you log on with the
# AccessKey pair of an Alibaba Cloud account because
# the account has permission s on all API operations
# in OSS . We recommend that you log on with a
# RAM user account to call API operations or perform
# routine operations and maintenanc e . To create a RAM
# user account , log on to https :// ram . console . aliyun .
# com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses China ( Hangzhou ) as the endpoint .
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourReques tBucketNam e >')

object_nam e = '< yourobject Name >'
local_file _name = '< yourLocalF ileName >'
down_file_ name = '< yourDownlo adFileName >'

# Set the bandwidth limit to 100 KB / s ( 819200 bit /
# s ) in request parameters .
limit_spee d = ( 100 * 1024 * 8 )
params = dict ()
params [ OSS_TRAFFI C_LIMIT ] = str ( limit_spee d );

# Create a signed URL for uploading an object and
# set the valid period of the URL to 60s .
url = bucket . sign_url (' PUT ', object_nam e , 60 , params =
params )
print (' put object url :', url )

# Configure bandwidth throttling for the object to be
# uploaded .
result = bucket . put_object _with_url_ from_file ( url ,
local_file _name )
print (' http response status :', result . status )

# Create a signed URL for downloadin g an object and
# set the valid period of the URL to 60s .
url = bucket . sign_url (' GET ', object_nam e , 60 , params =
params )
print (' get object url :', url )

# Configure bandwidth throttling for the object to be
# downloaded .
result = bucket . get_object _with_url_ to_file ( url ,
down_file_ name )
```

```
print (' http response status :', result . status )
```

3.7 Bucket tagging

By using the bucket tagging function, you can classify your buckets to manage them. For example, you can specify that only buckets with specified tags are returned in ListBucket operations.

- Only the bucket owner and authorized RAM users can add tags to a bucket. Otherwise, the 403 Forbidden error is returned with the error code: AccessDenied.
- You can add a maximum of 20 tags (key-value pairs) to a bucket.
- The maximum size of a key is 64 bytes. The key of a tag cannot be null and cannot be prefixed with `http ://`, `https ://`, or `Aliyun`.
- The maximum size of a tag value is 128 bytes. The value of a tag can be null.
- The key and value of a tag must be UTF-8 encoded.
- If you use PutBucketTagging to add tags to a bucket, the original tags added to the bucket are completely overwritten.

Add a tag to a bucket

You can run the following code to add a tag to a bucket:

```
# -*- coding : utf - 8 -*-
import oss2
from oss2 . models import Tagging , TaggingRule

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenance . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Creates a tagging rule .
rule = TaggingRule ()
rule . add (' key1 ', ' value1 ')
rule . add (' key2 ', ' value2 ')

# Creates a tag .
tagging = Tagging ( rule )
# Adds the tag to the bucket .
result = bucket . put_bucket_tagging ( tagging )
# Checks the returned HTTP status code .
```

```
print (' http status :', result . status )
```

Obtain the tags added to a bucket

You can run the following code to obtain the tags to a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .

auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Obtains the tagging informatio n about to a
bucketsult = bucket . get_bucket _tagging ()
# Views the obtained tagging rule .
tag_rule = result . tag_set . tagging_ru le
print (' tag rule :', tag_rule )
```

List buckets with a specified tag

Run the following code to list buckets with a specified tag:

```
# -*- coding : utf - 8 -*-
import oss2

# Creates a server object .
# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requiremen ts .
service = oss2 . Service ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ')

# Inputs the tag - key and tag - value fields to the
params parameter of the list_bucke ts interface .
params = {}
params [' tag - key '] = '< yourTaggin g_key >'
params [' tag - value '] = '< yourTaggin g_value >'

# Lists buckets with the specified tag .
```



```
result = service.list_buckets(params=params)
# Views the list results.
for bucket in result.buckets:
    print('result bucket_name:', bucket.name)
```

Delete the tags added to a bucket

You can run the following code to delete the tags added to a bucket:

```
# -*- coding: utf-8 -*-
import oss2

# It is highly risky to log on with the AccessKey
# of an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS. We recommend
# that you log on as a RAM user to access APIs or
# perform routine operations and maintenance. To create
# a RAM user account, log on to https://ram.console
# .aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
# KeySecret >')
# This example uses the China East 1 (Hangzhou)
# endpoint. Specify the actual endpoint based on your
# requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
# aliyuncs.com', '< yourBucketName >')

# Deletes the tags added to the bucket.
result = bucket.delete_bucket_tagging()
# Checks the returned HTTP status code.
print('http status:', result.status)
```

3.8 Object tagging

3.8.1 Configure object tagging

This topic describes how to configure object tagging.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).



Note:

- You can add tags to an object when uploading it or add tags for an uploaded object. If add tags to an object that already has tags, the original tags are overwritten. For more information about adding tags, see [#unique_52](#).
- To add tags to an object, you must have the permission to call PutObjectTagging.
- The Last-Modified value of an object is not updated when its tags are changed.
- A tag can contain letters, numbers, spaces, and the following symbols: + - = . _ : /

Add tags to an object when uploading it

- Add tags to an object when uploading it by calling `put_object`.

Run the following code to add tags to an object when uploading it by calling `put_object`:

```
# -*- coding : utf - 8 -*-

import oss2
from oss2 . headers import OSS_OBJECT_TAGGING

# It is highly risky to log on with the AccessKey
# of an Alibaba Cloud account because the account
# has permission s on all the APIs in OSS . We
# recommend that you log on as a RAM user to
# access APIs or perform routine operations and
# maintenanc e . To create a RAM user account , log on
# to https :// ram . console . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
# endpoint . Specify an actual endpoint based on your
# requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

object_name = '< yourObject Name >'

# Configures the tags in the HTTP headers .
headers = dict ()
headers [ OSS_OBJECT_TAGGING ] = " k1 = v1 & k2 = v2 & k3 = v3 "

# Set the headers of put_object so that the tags
# are added to the object to be uploaded .
result = bucket . put_object ( object_name , ' content ',
headers = headers )
print (' http response status : ', result . status )

# View the tags added to the uploaded object .
result = bucket . get_object_tagging (' yourObject Name ')
for key in result . tag_set . tagging_rule :
    print (' tagging key : {}, value : {}'. format ( key ,
result . tag_set . tagging_rule [ key ]))
```

- Add tags to an object when uploading it by calling `multipart_upload`.

Run the following code to add tags to an object when uploading it by calling `multipart_upload`:

```
# -*- coding : utf - 8 -*-

import os
import oss2
from oss2 import SizedFileAdapter , determine_part_size
from oss2 . models import PartInfo
from oss2 . headers import OSS_OBJECT_TAGGING

# It is highly risky to log on with the AccessKey
# of an Alibaba Cloud account because the account
# has permission s on all the APIs in OSS . We
```

```

    recommend that you log on as a RAM user to
    access APIs or perform routine operations and
    maintenance. To create a RAM user account, log on
    to https://ram.console.aliyun.com.
    auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
    KeySecret >')
# This example uses the China East 1 (Hangzhou)
endpoint. Specify an actual endpoint based on your
requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
aliyuncs.com', '< yourBucketName >')

object_name = '< yourObjectName >'
filename = '< yourLocalFile >'

total_size = os.path.getsize(filename)
# Uses the determine_part_size method to determine
the part size.
part_size = determine_part_size(total_size, preferred_
size=100 * 1024)

# Configures the tags in the HTTP headers.
headers = dict()
headers[OSS_OBJECT_TAGGING] = 'k1=v1 & k2=v2'

# Initializes the multipart upload task.
# Set the headers of init_multipart_upload so that
the tags are added to the object to be uploaded.
upload_id = bucket.init_multipart_upload(object_name,
headers=headers).upload_id
parts = []

# Uploads the parts one by one.
with open(filename, 'rb') as fileobj:
    part_number = 1
    offset = 0
    while offset < total_size:
        num_to_upload = min(part_size, total_size -
offset)
        # SizedFileAdapter(fileobj, size) generates a new
file object and recalculates the position where the
next upload task starts.
        result = bucket.upload_part(object_name,
upload_id, part_number,
                                SizedFileAdapter(fileobj,
num_to_upload))
        parts.append(PartInfo(part_number, result.etag))

        offset += num_to_upload
        part_number += 1

# Completes the multipart upload task.
result = bucket.complete_multipart_upload(object_name,
upload_id, parts)
print('http response status: ', result.status)

# Views the tags added to the object.
result = bucket.get_object_tagging(object_name)
for key in result.tag_set.tagging_rule:
    print('tagging key: {}, value: {}'.format(key,
result.tag_set.tagging_rule[key]))

```

```
# Verifies the result of the multipart upload task .
with open ( filename , ' rb ' ) as fileobj :
    assert bucket . get_object ( object_name ) . read () ==
fileobj . read ()
```

- Add tags to an object when uploading it by calling `append_object`.

Run the following code to add tags to an object when uploading it by calling `append_object`:

```
# -*- coding : utf - 8 -*-

import oss2
from oss2 . headers import OSS_OBJECT_TAGGING

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account
has permissions on all the APIs in OSS . We
recommend that you log on as a RAM user to
access APIs or perform routine operations and
maintenance . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
auth = oss2 . Auth ( '< yourAccess KeyId >' , '< yourAccess
KeySecret >' )
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ' , '< yourBucket Name >' )

object_name = '< yourObject Name >'

# Configures the tags in the HTTP headers .
headers = dict ()
headers [ OSS_OBJECT_TAGGING ] = " k1 = v1 & k2 = v2 & k3 = v3 "

# Uploads the object by calling append_object . Set
the headers of append_object so that the tags are
added to the object to be uploaded .
# Only the tags added when the object is uploaded
by calling append_object for the first time take
effect .
result = bucket . append_object ( object_name , 0 , '<
yourContent >' , headers = headers )

# Views the tags added to the object .
result = bucket . get_object_tagging ( ' yourObject Name ' )
for key in result . tag_set . tagging_rule :
    print ( ' tagging key : {} , value : {}' . format ( key ,
result . tag_set . tagging_rule [ key ] ) )
```

- Add tags to an object when uploading it by calling `resumable_upload`.

Run the following code to add tags to an object when uploading it by calling `resumable_upload`:

```
# -*- coding : utf - 8 -*-
import oss2
from oss2 . headers import OSS_OBJECT_TAGGING
```

```
# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account
has permissions on all the APIs in OSS. We
recommend that you log on as a RAM user to
access APIs or perform routine operations and
maintenance. To create a RAM user account, log on
to https://ram.console.aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 (Hangzhou)
endpoint. Specify an actual endpoint based on your
requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
aliyuncs.com', '< yourBucketName >')

object_name = '< yourObjectName >'
local_file = '< yourLocalFileName >'

# Configures the tags in the HTTP headers.
headers = dict()
headers[OSS_OBJECT_TAGGING] = "k1=v1 & k2=v2 & k3=v3"

# An object is uploaded by calling resumable_upload
when the object size is equal to or larger
than the value of the optional parameter multiple_t
hreshold (10 MB by default). If the store parameter
is not specified, a .py-oss-upload directory is
created under HOME to store breakpoint information.
# Set the headers of resumable_upload so that the
tags are added to the object to be uploaded.

oss2.resumable_upload(bucket, object_name, local_file
, headers=headers)

result = bucket.get_object_tagging(object_name)
for key in result.tag_set.tagging_rule:
    print('object tagging key: {}, value: {}'.format(
key, result.tag_set.tagging_rule[key]))
```

Add tags to an uploaded object or modify the tags added to an object

Run the following code to add tags to an uploaded object or modify the tags added to an object:

```
# -*- coding: utf-8 -*-

import oss2
from oss2.models import Tagging, TaggingRule

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenance. To create
a RAM user account, log on to https://ram.console
.aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 (Hangzhou)
endpoint. Specify an actual endpoint based on your
requirements.
```

```

bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Creates a tagging rule .
rule = TaggingRule ()
rule . add (' key1 ', ' value1 ')
rule . add (' key2 ', ' value2 ')

# Creates a tag .
tagging = Tagging ( rule )

# Adds the tag to the object .
result = bucket . put_object _tagging ('< yourObject Name >',
tagging )
# Views the returned HTTP status code .
print (' http response status :', result . status )

```

Add tags to an object when copying it

You can specify one of the following tagging rules when copying an object:

- **Copy (default):** Copy the tags of the source object to the destination object.
- **Replace:** Add the tags specified in the request to the destination object.

Examples of adding tags to objects of different sizes when copying them are described as follows:

- Run the following code to add tags to an object smaller than 1 GB when directly copying it:

```

# -*- coding : utf - 8 -*-

import oss2
from oss2 . headers import OSS_OBJECT_TAGGING ,
OSS_OBJECT_TAGGING_COPY_DIRECTIVE

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account
has permissions on all the APIs in OSS . We
recommend that you log on as a RAM user to
access APIs or perform routine operations and
maintenance . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Specifies the source object .
src_object_name = '< yourSrcObjectName >'
# Specifies the destination object 1 .
dest_object_name1 = '< yourDestObjectName1 >'
# Specifies the destination object 2 .
dest_object_name2 = '< yourDestObjectName2 >'

```

```
# Specifies OSS_OBJECT_TAGGING_COPY_DIRECTIVE to COPY
or keeps the default value in the HTTP headers ,
so that the tags added to the source object is
also added to the destination object 1 .
headers = dict ()
headers [ OSS_OBJECT_TAGGING_COPY_DIRECTIVE ] = ' COPY '
bucket . copy_object ( bucket . bucket_name , src_object
_name , dest_object_name1 , headers = headers )

# Specifies OSS_OBJECT_TAGGING_COPY_DIRECTIVE to
REPLACE in the HTTP headers , so that the tags
specified in OSS_OBJECT_TAGGING are added to the
destination object 2 .
headers [ OSS_OBJECT_TAGGING_COPY_DIRECTIVE ] = ' REPLACE '
headers [ OSS_OBJECT_TAGGING ] = " key4 = value4 & key5 = value5 "
bucket . copy_object ( bucket . bucket_name , src_object
_name , dest_object_name2 , headers = headers )

# Views the tags added to the object src_object_name
.
result = bucket . get_object_tagging ( src_object_name )
for key in result . tag_set . tagging_rule :
    print ( ' src tagging key : {}, value : {}'. format ( key
, result . tag_set . tagging_rule [ key ]))

# Views the tags added to the destination object
1 . The tags are the same as those of the source
object .
result = bucket . get_object_tagging ( dest_object_name1 )
for key in result . tag_set . tagging_rule :
    print ( ' dest1 object tagging key : {}, value : {}'.
format ( key , result . tag_set . tagging_rule [ key ]))

# Views the tags added to the destination object
2 . The added tags are specified in OSS_OBJECT
_TAGGING .
result = bucket . get_object_tagging ( dest_object_name2 )
for key in result . tag_set . tagging_rule :
    print ( ' dest2 object tagging key : {}, value : {}'.
format ( key , result . tag_set . tagging_rule [ key ]))
```

- Run the following code to add tags to an object larger than 1 GB when copying it by calling multipart_upload:

```
# -*- coding : utf - 8 -*-
import os
import oss2
from oss2 import determine_part_size
from oss2 . models import PartInfo
from oss2 . headers import OSS_OBJECT_TAGGING

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account
has permissions on all the APIs in OSS . We
recommend that you log on as a RAM user to
access APIs or perform routine operations and
maintenance . To create a RAM user account , log on
to https :// ram . console . aliyun . com .
auth = oss2 . Auth ( '< yourAccessKeyId >' , '< yourAccess
KeySecret >')
```

```

# This example uses the China East 1 (Hangzhou)
endpoint. Specify an actual endpoint based on your
requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
aliyuncs.com', '<yourBucket Name>')

# Specifies the source object.
src_object_name = '<yourSrcObject>'
# Specifies the destination object.
dest_object_name = '<yourDestObject>'

# Obtains the size of the source object.
head_info = bucket.head_object(src_object_name)
total_size = head_info.content_length
print('src object size:', total_size)

# Calls determine_part_size to determine the part size.
part_size = determine_part_size(total_size, preferred_
size=100 * 1024)
print('part_size:', part_size)

# Configures the tags in the HTTP headers.
headers = dict()
headers[OSS_OBJECT_TAGGING] = 'k3=v3'

# Initialize a multipart upload task.
# Set the headers of multipart upload so that the
tags are added to the destination object.
upload_id = bucket.init_multipart_upload(dest_objec
t_name, headers=headers).upload_id
parts = []

# Uploads the parts one by one.
part_number = 1
offset = 0
while offset < total_size:
    num_to_upload = min(part_size, total_size - offset)
    end = offset + num_to_upload - 1;
    result = bucket.upload_part_copy(bucket.bucket_nam
e, src_object_name, (offset, end), dest_objec t_name,
upload_id, part_number)
    # Saves the part information.
    parts.append(PartInfo(part_number, result.etag))

    offset += num_to_upload
    part_number += 1

# Completes the multipart upload task.
result = bucket.complete_multipart_upload(dest_objec
t_name, upload_id, parts)

# Obtains the metadata of the destination object.
head_info = bucket.head_object(dest_objec t_name)

# Views the size of the destination object.
dest_objec t_size = head_info.content_length
print('dest object size:', dest_objec t_size)

# Compares the size of the destination object with
that of the source object.
assert dest_objec t_size == total_size

# Views the tags added to the source object.

```



```

result = bucket.get_object_tagging(src_object_name)
for key in result.tag_set.tagging_rule:
    print('src tagging key: {}, value: {}'.format(key,
        result.tag_set.tagging_rule[key]))

# Views the tags added to the destination object.
result = bucket.get_object_tagging(dest_object_name)
for key in result.tag_set.tagging_rule:
    print('dest tagging key: {}, value: {}'.format(key,
        result.tag_set.tagging_rule[key]))

```

Add tags to a symbolic link object

Run the following code to add tags to a symbolic link object:

```

# -*- coding: utf-8 -*-

import oss2
from oss2.headers import OSS_OBJECT_TAGGING

# It is highly risky to log on with the AccessKey
# of an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS. We recommend
# that you log on as a RAM user to access APIs or
# perform routine operations and maintenance. To create
# a RAM user account, log on to https://ram.console
# .aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
# KeySecret >')
# This example uses the China East 1 (Hangzhou)
# endpoint. Specify an actual endpoint based on your
# requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
# aliyuncs.com', '< yourBucketName >')

object_name = '< yourObjectName >'
symlink_name = '< yourSymlinkName >'

# Configures the tags in the HTTP headers.
headers = dict()
headers[OSS_OBJECT_TAGGING] = "k1 = v1 & k2 = v2 & k3 = v3"

# Adds a symbolic link to the object.
# Set the headers of put_symbolic so that the tags
# are added to the symbolic link object.
result = bucket.put_symlink(object_name, symlink_name,
# headers=headers)
print('http response status:', result.status)

# Views the tags added to the symbolic link object.
result = bucket.get_object_tagging(symlink_name)
for key in result.tag_set.tagging_rule:

```

```
print (' tagging key : {}, value : {}'.format ( key , result
. tag_set . tagging_rule [ key ]))
```

3.8.2 Obtain the tags added to an object

This topic describes how to obtain the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to obtain the tags added to an object:

```
# -*- coding : utf - 8 -*-

import oss2
from oss2 . models import Tagging , TaggingRule

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenance . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Obtains the tags added to the object .
result = bucket . get_object _tagging ('< yourObject Name >')

# Views the tags added to the object .
for key in result . tag_set . tagging_rule :
    print (' tagging key : {}, value : {}'.format ( key , result
. tag_set . tagging_rule [ key ]))
```

For more information about how to obtain the tags added to an object, see [#unique_54](#).

3.8.3 Delete the tags added to an object

This topic describes how to delete the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to delete the tags added to an object:

```
# -*- coding : utf - 8 -*-

import oss2
from oss2 . models import Tagging , TaggingRule
```

```
# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ( '< yourAccess KeyId >', '< yourAccess
KeySecret >' )
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >' )

# Deletes the tags added to the object .
result = bucket . delete_obj ect_taggin g ( '< yourObject Name
>' )
print ( ' http response status : ', result . status )
```

For more information about how to delete the tags added to an object, see [#unique_56](#).

3.8.4 Object tagging and lifecycle rules

Lifecycle rules can take effect on objects with specified prefixes or tags. You can also specify prefixes and tags as the condition of a lifecycle rule at the same time.



Note:

If you set a specified tag as the condition of a lifecycle rule, the rule applies to an object only when the key and value in the tag of the object both match the specified tag. If you specify a prefix and multiple tags as the condition of a lifecycle rule, the rule applies to an object only when the prefix and tags of the object match the prefix and all tags specified in the rule.

Set tags as the matching condition of a lifecycle rule

Run the following code to set tags as the matching condition of a lifecycle rule:

```
# -*- coding : utf - 8 -*-
import oss2
import datetime
from oss2 . models import ( LifecycleE xpiration , LifecycleR
ule ,
                                BucketLife cycle , AbortMulti partUpload ,
                                TaggingRul e , Tagging , StorageTra
nsition )

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
```

```

a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Sets an expiration rule for objects so that objects
are expired three days after they are modified for
the last time .
rule1 = LifecycleRule ( ' rule1 ', ' tests /',
                        status = LifecycleRule . ENABLED ,
                        expiration = LifecycleExpiration ( days = 3
))

# Sets an expiration rule for objects so that objects
created before the specified date are expired .
rule2 = LifecycleRule ( ' rule2 ', ' logging -',
                        status = LifecycleRule . DISABLED ,
                        expiration = LifecycleExpiration ( created_before_date = datetime
. date ( 2018 , 12 , 12 )))

# Sets an expiration rules for parts so that parts
are expired 3 days after they are generated .
rule3 = LifecycleRule ( ' rule3 ', ' tests1 /',
                        status = LifecycleRule . ENABLED ,
                        abort_multipart_upload = AbortMultipartUpload (
days = 3 ))

# Sets an expiration rules for parts so that parts
generated before the specified date are expired .
rule4 = LifecycleRule ( ' rule4 ', ' logging1 -',
                        status = LifecycleRule . DISABLED ,
                        abort_multipart_upload = AbortMultipart
partUpload ( created_before_date = datetime . date ( 2018 , 12 ,
12 )))

# Sets tags as the matching condition .
tagging_rule = TaggingRule ()
tagging_rule . add ( ' key1 ', ' value1 ')
tagging_rule . add ( ' key2 ', ' value2 ')
tagging = Tagging ( tagging_rule )

# Sets a storage class conversion rule for objects so
that the storage class of an object is converted
to Archive 365 days after it is modified for the
last time .
# Different from the preceding rules , rule5 includes
tags that are specified as the matching condition ,
so that it only applies to objects with the key1 =
value1 and key2 = value2 tags at the same time .
rule5 = LifecycleRule ( ' rule5 ', ' logging2 -',
                        status = LifecycleRule . ENABLED ,
                        storage_transitions = [ StorageTransition
( days = 356 , storage_class = oss2 . BUCKET_STORAGE_CLASS
_ARCHIVE )],
                        tagging = tagging )

lifecycle = BucketLifecycle ([ rule1 , rule2 , rule3 , rule4
, rule5 ])

```

```
bucket . put_bucket _lifecycle ( lifecycle )
```

View the tags set as the matching condition of a lifecycle rule

Run the following code to view the tags set as the matching condition of a lifecycle rule:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify an actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Views the lifecycle rules set for a bucket .
lifecycle = bucket . get_bucket _lifecycle ()

for rule in lifecycle . rules :
    # Views the expiration rules for parts .
    if rule . abort_mult ipart_uplo ad is not None :
        print (' id ={ 0 }, prefix ={ 1 }, tagging ={ 2 }, status
        ={ 3 }, days ={ 4 }, created_be fore_date ={ 5 }'
              . format ( rule . id , rule . prefix , rule .
tagging , rule . status ,
                      rule . abort_mult ipart_uplo ad . days ,
                      rule . abort_mult ipart_uplo ad . created_be
fore_date ))

    # Views the expiration rules for objects
    if rule . expiration is not None :
        print (' id ={ 0 }, prefix ={ 1 }, tagging ={ 2 }, status
        ={ 3 }, days ={ 4 }, created_be fore_date ={ 5 }'
              . format ( rule . id , rule . prefix , rule .
tagging , rule . status ,
                      rule . expiration . days ,
                      rule . expiration . created_be fore_date ))
    # Views the storage class conversion rule .
    if len ( rule . storage_tr ansitions ) > 0 :
        storage_tr ans_info = ''
        for storage_ru le in rule . storage_tr ansitions :
            storage_tr ans_info += ' days ={ 0 }, created_be
fore_date ={ 1 }, storage_cl ass ={ 2 } **** ' . format (
                storage_ru le . days , storage_ru le . created_be
fore_date , storage_ru le . storage_cl ass )

        print (' id ={ 0 }, prefix ={ 1 }, tagging ={ 2 }, status
        ={ 3 },, StorageTra nsition ={ 4 }')
```

```
        .format ( rule . id , rule . prefix , rule .  
tagging , rule . status , storage_tr  ans_info ))
```

3.9 Upload objects

3.9.1 Overview

This topic is the overview of object upload in OSS Python SDK.

In OSS, an object is the basic unit for data operations. OSS Python SDK provides the following file upload methods:

- [Simple upload](#): supports the upload of a file up to 5 GB in size.
- [Append upload](#): supports the upload of a file up to 5 GB in size.
- [Resumable upload](#): supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files. It supports concurrent upload and you can define the size of each part.
- [Multipart upload](#): supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files.



Note:

For more information about the usage scenarios of each upload method, see “Upload files” in OSS Developer Guide.

During the file upload, you can set [Object Meta](#), and view the upload progress in the [upload progress bar](#). After you complete the file upload, you can perform [upload callback](#).

3.9.2 Simple upload

In simple uploads, you can upload files of the following five types: strings, bytes, Unicode, local files, and file streams.

You can use the `bucket.put_object` method to upload a file. This method supports multiple types of inputs, which are listed in the following table.

Type	Upload method
String	Direct uploaded
Bytes	Direct uploaded
Unicode	Uploaded after being converted to bytes encoded in UTF-8.

Type	Upload method
Local files	Uploaded as File objects, which must be opened in the binary mode.
File stream	Uploaded as Iterable objects in Chunked Encoding method.

**Note:**

If you upload an object whose name is the same as that of an existing object that you can access, the existing object is overwritten by the uploaded object, and a 200 OK message is returned.

For more information about PutObject, see [Detail analysis](#).

Upload a string

You can run the following code to upload a string:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenance . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
# the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Return value
result = bucket . put_object ('< yourObject Name >', ' content
of object ')
# HTTP return code
print (' http status : { 0 }'. format ( result . status ))
# Request ID The request ID is the unique identifica
tion of a request . We strongly recommend you to add
this parameter in the program log .
print (' request_id : { 0 }'. format ( result . request_id ))
# Etag is a unique attribute of the value returned
by put_object .
print (' ETag : { 0 }'. format ( result . etag ))
# HTTP response header
```

```
print (' date : { 0 }'. format ( result . headers [' date ']))
```

Upload bytes

You can run the following code to upload bytes:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . put_object ('< yourObject Name >', b ' content of
object ')
```

Upload Unicode

You can run the following code to upload Unicode:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . put_object ('< yourObject Name >', u ' content of
object ')
```

Upload local files

You can run the following code to upload a local file:

```
# -*- coding : utf - 8 -*-
import oss2
```



```
# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Files must be opened in binary mode because the
number of bytes included in the file is required
for upload .
with open ('< yourLocalF ile >', ' rb ') as fileobj :
    # The seek method specifies that the read or
write operations start from the 1000th byte . The
upload starts from the 1000th byte until the end of
the file .
    fileobj . seek ( 1000 , os . SEEK_SET )
    # The tell method is used to return to the
current location .
    current = fileobj . tell ()
    bucket . put_object ('< yourObject Name >', fileobj )
```

OSS Python SDK provides a more convenient method to upload local files:

```
bucket . put_object _from_file ('< yourObject Name >', '<
yourLocalF ile >')
```

Upload a file stream

You can run the following code to upload a file stream:

```
# -*- coding : utf - 8 -*-
import oss2
import requests

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# The requests . get method returns an Iterable object
, and the Python SDK uploads the file stream in
Chunked Encoding method .
input = requests . get (' http :// www . aliyun . com ')
```

```
bucket . put_object ('< yourObject Name >', input )
```

3.9.3 Append upload

This topic describes how to use append upload.

Run the following code for append upload:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenance . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Set the position ( the Position parameter ) where the
# object is appended to 0 .
result = bucket . append_obj ect ('< yourObject Name >', 0 , '
content of first append ')
# If the object is not uploaded for the first time
# , you can use the bucket . head_objec t method or the
# next_posit ion attribute returned in the last append
# upload to get the append position .
bucket . append_obj ect ('< yourObject Name >', result .
next_posit ion , ' content of second append ')
```

If the object already exists, exceptions occur in the following two scenarios:

- The file cannot be uploaded with append upload. In this case, an `ObjectNotAppendable` exception occurs.
- The file can be uploaded with append upload, but the append position does not match the current file length. In this case, a `PositionNotEqualToLength` exception occurs.

3.9.4 Resumable upload

You can use resumable upload to split a file you want to upload into several parts and upload them simultaneously. After you have uploaded all parts, you can combine

these parts into a complete object. Thus, you can complete uploading the entire object.

You can use the `oss2.resumable_upload` method to upload a specified file with resumable upload. You can configure the following parameters.

Parameter	Description	Required or not	Default value
bucket	Specifies the bucket name.	Required	None
key	Specifies the object name.	Required	None
filename	Specifies the name of the local file to upload.	Required	None
store	Specifies the directory where the progress information is stored.	Optional	Directory 'py-oss-upload' under the HOME directory
headers	Specifies the HTTP header.	Optional	None
multipart_threshold	Specifies a object length threshold. If the object length is larger than the threshold, the resumable upload is used.	Optional	10 MB
part_size	Specifies the part size.	Optional	Auto-calculated
progress_callback	Specifies the upload progress callback function.	Optional	None
num_threads	Specifies the number of concurrent upload threads.	Optional	1

Run the following code for resumable upload:

```
# -*- coding : utf - 8 -*-
```

```
import oss2
# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')
# When the file length is larger than or be equal
# to the value of the optional parameter multipart_
# threshold ( 10 MB by default ), the resumable upload
# is used . If you do not use the store parameter
# to specify the directory , a py - oss - upload directory
# is created under the HOME directory to save upload
# progress informatio n .
oss2 . resumable_ upload ( bucket , '< yourObject Name >', '<
yourLocalF ile >')
```

Python SDK 2.1.0 and later support resumable upload with optional parameters, as shown in the following code:

```
# If you use the store parameter to specify the
# directory , the progress informatio n is stored in the
# specified directory . If you use the num_thread s to
# set the number of concurrent upload threads , ensure
# that the value of oss2 . defaults . connection _pool_size
# is no less than the thread number .. The number of
# concurrent threads is 1 by default .
oss2 . resumable_ upload ( bucket , '< yourObject Name >', '<
yourLocalF ile >',
    store = oss2 . ResumableS tore ( root = '/ tmp ' ),
    multipart_ threshold = 100 * 1024 ,
    part_size = 100 * 1024 ,
    num_thread s = 4 )
```

For more information about resumable upload, see [Resumable upload](#) in the OSS Developer Guide.



Note:

- Multiple threads are launched in resumable uploads or downloads. Therefore, you do not need to encapsulate multiple upload or download threads when calling the method. Otherwise, data may be transmitted repeatedly.
- We recommend you set a larger part size when the network connection is stable. Otherwise, set a smaller part size.

3.9.5 Multipart upload

This topic describes how to use multipart upload.

To enable multipart upload, perform the following steps:

1. Initialization (using `bucket.init_multipart_upload`): Obtain the Upload ID.
2. Upload parts (using `bucket.upload_part`): Upload data parts. You can upload multiple parts concurrently.
3. Completing multipart upload (using `bucket.complete_multipart_upload`): After you have uploaded all parts, combine these parts into a complete object.

Run the following code for multipart upload:

```
# -*- coding : utf - 8 -*-
import os
from oss2 import SizedFileAdapter, determine_part_size
from oss2.models import PartInfo
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

key = '< yourObject Name >'
filename = '< yourLocalF ile >'

total_size = os . path . getsize ( filename )
# Use determine_ part_size to determine the part size .
part_size = determine_ part_size ( total_size , preferred_ size
= 100 * 1024 )

# Initialize a multipart upload event .
upload_id = bucket . init_multi part_uploa d ( key ). upload_id
parts = []

# Upload parts one by one .
with open ( filename , ' rb ' ) as fileobj :
    part_numbe r = 1
    offset = 0
    while offset < total_size :
        num_to_upl oad = min ( part_size , total_size - offset
    )
    # The SizedFileA dapter ( fileobj , size ) method generates
    # a new object , and re - calculates the initial append
    # location .
```

```

        result = bucket.upload_part(key, upload_id,
                                     part_number,
                                     SizedFileAdapter(fileobj,
                                                         num_to_upload))
        parts.append(PartInfo(part_number, result.etag))

        offset += num_to_upload
        part_number += 1

# Complete multipart upload.
bucket.complete_multipart_upload(key, upload_id, parts)

# Verify the multipart upload.
with open(filename, 'rb') as fileobj:
    assert bucket.get_object(key).read() == fileobj.read()

```

**Note:**

We recommend you set a larger part size when the network connection is stable. Otherwise, set a smaller part size.

3.9.6 Upload progress bars

You can use progress bars to indicate the upload or download progress.

For the complete code of progress bars, see [GitHub](#).

The following code is used as an example to describe how to view progress information with `bucket.put_object`:

```

# -*- coding: utf-8 -*-
from __future__ import print_function
import os, sys
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS. We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenance. To
# create a RAM account, log on to https://ram.console
# .aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
# KeySecret >')
# This example uses endpoint China (Hangzhou). Specify
# the actual endpoint based on your requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
# aliyuncs.com', '< yourBucketName >')
# If the length of the data to be uploaded cannot
# be determined, the value of total_bytes is None.
def percentage(consumed_bytes, total_bytes):
    if total_bytes:
        rate = int(100 * (float(consumed_bytes) / float(
# total_bytes)))
        print('\r {0}% '.format(rate), end='')
        sys.stdout.flush()
# progress_callback is an optional parameter used to
# implement progress bars.

```

```
bucket . put_object ('< yourObject Name >', ' a ' * 1024 * 1024 ,
progress_c allback = percentage )
```

3.9.7 Upload callback

This topic describes how to use upload callback.

For the complete code of upload callback, see [GitHub](#).

Run the following code for upload callback:

```
# -*- coding : utf - 8 -*-
import json
import base64
import os
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ) . Specify
# the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Prepare callback parameters .
callback_d ict = {}
# Configure the IP address of the server you want
# to send the callback request to , for example , http
# :// oss - demo . aliyuncs . com : 23450 or http :// 127 . 0 . 0 .
# 1 : 9090 .
callback_d ict [' callbackUr l ' ] = ' http :// oss - demo .
aliyuncs . com : 23450 '
# Configure the value of the Host field carried in
# the callback request header , such as oss - cn - hangzhou
# . aliyuncs . com .
callback_d ict [' callbackHo st ' ] = ' oss - cn - hangzhou .
aliyuncs . com '
# Configure the value of the body field carried in
# the callback request .
callback_d ict [' callbackBo dy ' ] = ' filename = ${ object } & size
= ${ size } & mimeType = ${ mimeType } '
# Configure Content - Type for the callback request .
callback_d ict [' callbackBo dyType ' ] = ' applicatio n / x - www
- form - urlencoded '
# The callback parameter is in the JSON format and
# Base64 - encoded .
callback_p aram = json . dumps ( callback_d ict ) . strip ()
base64_cal lback_body = base64 . b64encode ( callback_p aram )
# Encoded callback parameters are carried in the
# request header and are sent to OSS .
headers = { ' x - oss - callback ' : base64_cal lback_body }

# Upload and callback .
```

```
result = bucket.put_object('< yourObject Name >', 'a' * 1024 * 1024, headers)
```

The `put_object`, `put_object_from_file`, and `complete_multipart_upload` methods provide upload callback features. For more information about upload callback, see [Upload callback](#) in the Development Guide.

3.10 Download objects

3.10.1 Overview

This topic is the overview of object download in OSS Python SDK.

OSS Python SDK provides the following download methods:

- [Streaming download](#)
- [Download to your local file](#)
- [Range download](#)
- [Resumable download](#)

You can view the download progress shown in the [download progress bar](#).

3.10.2 Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use streaming download. Streaming download enables you to download part of the object each time until you have downloaded the entire object.

Run the following code for streaming download:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
// This example uses the endpoint China East 1 (
Hangzhou ). Specify the actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# The return value of bucket . get_object is a file -
like object , which is also an iterable object .
object_str eam = bucket . get_object ('< yourObject Name >')
```



```
print ( object_str eam . read ())

# The get_object interface returns a stream , which must
  be read by read () before being used to calculate
  the CRC checksum of the returned object data .
  Therefore , you must perform CRC verificati on after
  calling the interface .
if object_str eam . client_crc != object_str eam . server_crc
:
    print " The CRC checksum between client and server
  is inconsiste nt !"
```

Run the following code to download data to a local file:

```
import shutil

# object_str eam is a file - like object . Therefore , you
  can use the shutil . copyfileob j method to download
  data to a local file .
object_str eam = bucket . get_object ('< yourObject Name >')
with open ('< yourLocalF ile >', ' wb ') as local_file obj :
    shutil . copyfileob j ( object_str eam , local_file obj )
```

Run the following code to copy an object to another object using streaming copy.

```
# object_str eam is an iterable object , which can be
  copied to another object using streaming copy .
object_str eam = bucket . get_object ('< yourObject Name >')
bucket . put_object ('< yourBackup ObjectName >', object_str eam )
```

3.10.3 Download to your local file

This topic describes how to download an object from OSS to a local file.

You can run the following code to download the specified OSS object to a local file:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
  an Alibaba Cloud account because the account has
  permission s on all the APIs in OSS . We recommend
  that you log on as a RAM user to access APIs
  or perform routine operations and maintenanc e . To
  create a RAM account , log on to https :// ram . console
  . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
  KeySecret >')
# This example uses the endpoint China East 1 (
  Hangzhou ). Specify the actual endpoint based on your
  requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
  aliyuncs . com ', '< yourBucket Name >')

# Download an object to your local file . If the
  name of the object is the same as that of your
  local file , the object will replace the local file .
  If the name of the object is different from that
  of your local file , the object will be downloaded
```

```
and a new file will be added to your local device
bucket.get_object_to_file('< yourObject Name >', '< yourLocalFile >')
```

3.10.4 Range download

If you only need part of the data in an object, you can use range downloads to download specified content.

Run the following code for range download:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the endpoint China East 1 (
# Hangzhou ). Specify the actual endpoint based on your
# requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Obtain the data ranging from byte 0 to the 99th
# byte of the object ( including byte 0 and the 99th
# byte , a total of 100 bytes ). If the specified
# value range is invalid , for example , the specified
# range includes a negative number , or the specified
# value is larger than the object size , the entire
# object is downloaded .
object_stream = bucket . get_object ('< yourObject Name >',
byte_range =( 0 , 99 ))
```

3.10.5 Resumable download

Large-sized objects may fail to be downloaded because the network is unstable or the program exits abnormally and the entire object needs to be downloaded again. However, the object may still fail to be downloaded after multiple attempts. Therefore, OSS provides resumable download.

The process of resumable download is as follows:

1. A temporary local file is created. The file name consists of the original object name and a random suffix.

2. Read the OSS object based on the range you specified in the Range header in the HTTP request and write the read content to the corresponding position of the temporary file.
3. After the download is complete, rename the temporary file as the target file. If the target file already exists, it is overwritten. If the target file does not exist, a new file is created.

You can use `oss2.resumable_download` to enable resumable download. It contains the following parameters:

Parameter	Description	Required or optional	Default value
bucket	Specifies the name of a bucket.	Required	None
key	Specifies the name of an object.	Required	None
filename	Specifies the name of a local file. The OSS object is download to the local file.	Required	None
store	Specifies the local file that records the download progress information. Download progress information is stored in the file . If the download fails at a point, the next download starts from the point based on the recorded progress information.	Optional	The .py-OSS-download directory created under the HOME directory

Parameter	Description	Required or optional	Default value
<code>multipart_threshold</code>	Specifies a length threshold for the object size. If the object size exceeds the threshold, resumable download is enabled.	Optional	10 MB
<code>part_size</code>	Specifies the size of a part.	Optional	Auto-calculated
<code>progress_callback</code>	Specifies the download progress callback function	Optional	None
<code>num_threads</code>	Specifies the number of parts you need to download simultaneously.	Optional	1

Run the following code for resumable download:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

oss2 . resumable_ download ( bucket , '< yourObject Name >', '<
yourLocalF ile >')
```

Python SDK 2.1.0 and later support optional parameters for resumable download, as shown in the following code:

```
# -*- coding : utf - 8 -*-
import oss2
```

```
# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Make sure that the value of oss2 . defaults .
connection _pool_size is no less than the number of
the download threads , and the value of part_size is
no less than the value of oss2 . defaults . multiget_p
art_size .
oss2 . resumable_ download ( bucket , '< yourObject Name >', '<
yourLocalF ile >',
store = oss2 . ResumableD ownloadSto re ( root ='/ tmp '),
multiget_t hreshold = 20 * 1024 * 1024 ,
part_size = 10 * 1024 * 1024 ,
num_thread s = 3 )
```

**Note:**

Do not allow multiple programs (threads) to download the same object to the same target file simultaneously by calling the method. Because the progress information of the threads overwrites each other, and the temporary file names specified by the threads may conflict.

3.10.6 Download progress bars

You can use progress bars to indicate the upload or download progress.

For the complete code of progress bars, see [GitHub](#).

The following code is used as an example to describe how to view progress information with `bucket.get_object_to_file`:

```
# -*- coding : utf - 8 -*-
from __future__ import print_func tion
import os , sys
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
```

```

auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')
# If the HTTP response header does not carry Content
- Length , the value of total_byte s is None .
def percentage ( consumed_b ytes , total_byte s ):
    if total_byte s :
        rate = int ( 100 * ( float ( consumed_b ytes ) / float (
total_byte s )))
        print ('\ r { 0 }% '. format ( rate ), end='')

        sys . stdout . flush ()

# progress_c allback is an optional parameter used to
implement progress bar .
bucket . get_object _to_file ('< yourObject Name >', '< yourLocalF
ile >', progress_c allback = percentage )

```

3.11 Manage objects

3.11.1 Overview

This topic is the overview of object management in OSS Python SDK.

You can perform the following operations on objects in a bucket with APIs:

- [Determine whether an object exists](#)
- [Manage ACL for an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Delete an object](#)
- [Copy an object](#)
- [Restore an archive object](#)
- [Manage a symbolic link](#)

3.11.2 Determine whether a specified object exists

This topic describes how to determine whether a specified object exists.

Run the following code to determine whether a specified object exists:

```

# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs

```

```

    or perform routine operations and maintenance. To
    create a RAM account, log on to https://ram.console
    .aliyun.com.
    auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
    KeySecret >')
    # This example uses endpoint China East 1 (Hangzhou).
    Specify the actual endpoint based on your requiremen
    ts .
    bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
    aliyuncs . com ', '< yourBucket Name >')

    exist = bucket . object_exi sts ('< yourObject Name >')
    # If the returned value is true, it indicates that
    the specified object exists. If the returned value is
    false, it indicates that the specified object does
    not exist .
    if exist :
        print (' object exist ')
    else :
        print (' object not exist ')

```

3.11.3 Manage ACL for an object

The following table describes the permissions included in the Access Control List (ACL) for an object.

Permission	Description	Value
default	The ACL of an object is the same with that of its bucket.	oss2. OBJECT_ACL_DEFAULT
Private	Only the object owner and authorized users can read and write the object.	oss2. OBJECT_ACL_PRIVATE
Public read	Only the object owner and authorized users can read and write the object. Other users can only read the object. Authorize this permission with caution.	oss2. OBJECT_ACL_PUBLIC_READ
Public read-write	All users can read and write the object. Authorize this permission with caution.	oss2. OBJECT_ACL_PUBLIC_READ_WRITE

The ACL of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and

write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

Configure an ACL for an object

Run the following code to configure an ACL for an object:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Configure the ACL for the object .
bucket . put_object _acl ('< yourObject Name >', oss2 .
OBJECT_ACL _PUBLIC_RE AD )
```

Obtain the ACL for an object

You can run the following code to obtain the ACL for an object:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')
```



```
print ( bucket . get_object _acl ('< yourObject Name >'). acl )
```

3.11.4 Manage Object Meta

This topic details how to manage Object Meta. For more information, see [Object Meta](#) in the OSS Developer Guide.

Configure HTTP headers

OSS supports user-defined HTTP headers. Run the following code to configure HTTP headers:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . put_object ('< yourObject Name >', '{" age ": 1 }',
headers ={' Content - Type ': ' applicatio n / json ; charset = utf
- 8 '})
```

For more information about HTTP headers, see [RFC2616](#).

Configure user-defined Object Meta

You can define user-defined Object Meta for object descriptions. Run the following code to configure user-defined Object Meta:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
```

```
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . put_object (', ' a novel ', headers ={' x - oss - meta -
author ': ' 0 . Henry ', ' Content - Type ': ' applicatio n / json
; charset = utf - 8 '})
```

**Note:**

Parameters with the x-oss-meta- prefix are used as the headers of user-defined Object Meta. For more information, see [PutObject](#).

Modify Object Meta

Run the following code to modify Object Meta:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operation and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . update_obj ect_meta ('< yourObject Name >', {' x - oss -
meta - author ': ' 0 . Henry '})
# User - defined Object Meta is cleared each time the
bucket . update_obj ect_meta is called and must be
configured again .
bucket . update_obj ect_meta ('< yourObject Name >', {' x - oss -
meta - price ': ' 100 dollar '})
```

Run the following code to change Object Meta, such as Content-Type:

```
bucket . update_obj ect_meta ('< yourObject Name >', {' x - oss -
meta - author ': ' 0 . Henry '})
# User - defined Object Meta is cleared each time the
bucket . update_obj ect_meta is called and must be
configured again .
bucket . update_obj ect_meta ('< yourObject Name >', {' Content -
Type ': ' text / plain '})
```

Obtain Object Meta

You can use the APIs provided by OSS Python SDK to obtain Object Meta.

Method	Description	Advantage
get_object_meta	Obtains the ETag, Size (object size), and LastModified (the time when the object is last modified) of an object.	Simpler and faster
head_object	Obtains all Object Meta.	None

Run the following code to obtain Object Meta:

```
# -*- coding : utf - 8 -*-
import oss2

# We recommend that you do not log on with the
# AccessKey of an Alibaba Cloud account because the
# Alibaba Cloud account has permissions on all APIs
# in OSS. Instead, we recommend that you log on as
# an authorized RAM user to access APIs or perform
# routine operation and maintenance. To create a RAM
# account, log on to https://ram.console.aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
KeySecret >')
# This example uses the endpoint China East 1 (
# Hangzhou). In actual scenarios, specify your required
# endpoint.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
aliyuncs.com', '< yourBucket Name >')

# Obtain part of Object Meta.
simplified_meta = bucket.get_object_meta("< yourObject Name
>")
print(simplified_meta.headers['Last-Modified'])
print(simplified_meta.headers['Content-Length'])
print(simplified_meta.headers['ETag'])

# Obtain all Object Meta.
objectmeta = bucket.head_object("< yourObject Name >")
print(objectmeta.headers['Content-Type'])
print(objectmeta.headers['Last-Modified'])
print(objectmeta.headers['x-oss-object-type'])
```

3.11.5 List objects

Objects are listed alphabetically. You can use OSS Python SDK to list objects in different methods.

Simple list

Run the following code to list 10 objects in a specified bucket.

```
# -*- coding : utf - 8 -*-
import oss2
from itertools import islice
```

```
# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

for b in islice ( oss2 . ObjectIter ator ( bucket ), 10 ):
    print ( b . key )
```

List objects by a specified prefix

Run the following code to list objects with a specified prefix:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# List objects by a specified prefix 100 objects are
# listed by default .
for obj in oss2 . ObjectIter ator ( bucket , prefix = ' img
-'):
    print ( obj . key )
```

List all objects in a bucket

Run the following code to list all objects in a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
```

```

auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Set the delimiter parameter to a slash (/).
for obj in oss2 . ObjectIter ator ( bucket , delimiter =
'/'):
# Determine whether the obj is a directory using the
is_prefix method .
    if obj . is_prefix (): # directory
        print (' directory : ' + obj . key )
    else : # object
        Print (' file : ' + obj . Key )

```

3.11.6 Delete objects

This topic describes how to delete objects.



Warning:

Delete objects with caution because deleted objects cannot be recovered.

Delete an object

Run the following code to delete a single object:

```

# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . delete_obj ect ('< yourObject Name >')

```

Delete multiple objects

Run the following code to delete multiple objects simultaneously:

```

# Delete 3 objects . You can delete a maximum of 1 ,
000 objects simultaneo usly .
result = bucket . batch_dele te_objects (['< yourObject Name - a
>', '< yourObject Name - b >', '< yourObject Name - c >'])
# Print the names of deleted objects .

```

```
print ('\n '.join ( result . deleted_keys ))
```

3.11.7 Copy objects

You can copy an object from a bucket (source bucket) to another bucket (target bucket) in the same region.

You must note the following limits before copy an object:

- You must have the read and write permission on the source object.
- You cannot copy an object from a region to another region. For example, you cannot copy an object from a bucket in Hangzhou region to a bucket in Qingdao region.

Simple copy

You can use simple copy for objects smaller than 1 GB. Run the following code for simple copy:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourDestin ationBucke tName >')

bucket . copy_objec t ('< yourSource BucketName >', '< yourSource
ObjectName >', '< yourDestin ationObjec tName >')
```

Copy a large-sized object

To copy an object larger than 1 GB, you need to use UploadPartCopy. To enable UploadPartCopy, perform the following steps:

1. Use bucket.init_multipart_upload to initiate an UploadPartCopy task.
2. Start UploadPartCopy with bucket.upload_part_copy. Aside from the last part, all parts must be larger than 100 KB.
3. Submit the UploadPartCopy task with bucket.complete_multipart_upload.

Run the following code for UploadPartCopy task

```
# -*- coding : utf - 8 -*-
import oss2
from oss2.models import PartInfo
from oss2 import determine_part_size

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

src_object = '< yourSource ObjectName >'
dst_object = '< yourDestin ationObjec tName >'

total_size = bucket . head_objec t ( src_object ) . content_le
ngth
part_size = determine_ part_size ( total_size , preferred_ size
= 100 * 1024 )

# Initiate an UploadPart Copy task
upload_id = bucket . init_multi part_uploa d ( dst_object ) .
upload_id
parts = []

# Copy parts one by one .
part_numbe r = 1
offset = 0
while offset < total_size :
    num_to_upl oad = min ( part_size , total_size - offset )
    byte_range = ( offset , offset + num_to_upl oad - 1 )

    result = bucket . upload_par t_copy ( bucket . bucket_nam e
, src_object , byte_range , dst_object , upload_id , part_numbe
r )
    parts . append ( PartInfo ( part_numbe r , result . etag ))

    offset += num_to_upl oad
    part_numbe r += 1

# Complete the UploadPart Copy task .
bucket . complete_m ultipart_u pload ( dst_object , upload_id ,
parts )
```

For more information about UploadPartCopy, see [UploadPartCopy](#)

3.11.8 Restore an archive object

This topic describes how to restore an archive object.

You must restore an archive object before you read it. Do not call `restoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

1. An archive object is in the frozen state.
2. After you submit it for restoration, the server restores the object. The object is in the restoring state.
3. You can read the object after it is restored. The restored state of the object lasts one day by default. You can prolong this period to a maximum of seven days. Once this period expires, the object returns to the frozen state.

Run the following code to restore an object:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')
objectName = '< yourObject Name >'

bucket . restore_ob ject ( objectName )
```

For more information about the archive storage classes, see [Introduction to storage classes](#).

3.11.9 Manage a symbolic link

This topic describes how to manage a symbolic link.

Create a symbolic link

A symbolic link is a special object that maps to an object similar to a shortcut used in Windows.

Run the following code to create a symbolic link:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

objectName = '< yourObject Name >'
symlink = "< yourSymlin k >";

bucket . put_symlin k ( objectName , symlink )
```

For more information about symbolic links, see [PutSymlink](#).

Obtain object content for a symbolic link

To obtain a symbolic link, you must have the read permission on it. Run the following code to obtain the object content for a symbolic link:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

symlink = "< yourSymlin k >";

bucket . get_symlin k ( symlink )
```

For more information about symbolic links, see [GetSymlink](#).

3.12 Authorized access

This topic describes how to authorize access to users.

Sign a URL to authorize temporary access

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time for a URL to restrict the period of access from visitors.

Run the following code to download objects with a signed URL:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all APIs in OSS . We recommend that
# you log on as a RAM user to access APIs or
# perform routine operations and maintenanc e . To create
# a RAM account , log on to https :// ram . console . aliyun
# . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
# the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Set the expiration time of the URL to one hour .
print ( bucket . sign_url (' GET ', '< yourObject Name >', 60 ))
```

Use STS for temporary access authorization

For the complete code of using STS, see [GitHub](#).

You can use Security Token Service (STS) for temporary access authorization.

For more information about STS, see [Introduction](#) in the access control API (STS) reference. For more information about accounts and authorization, see [STS temporary access authorization](#) in best practices.

You must install the official Python STS client first:

```
pip install aliyun - python - sdk - sts
```

Please use version 2.0.6 and later. Run the following code to upload files with STS temporary authorization:

```
# -*- coding : utf - 8 -*-
from aliyunsdkc ore import client
```

```

from aliyunsdks ts . request . v20150401 import AssumeRole
Request
import json
import oss2

# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
endpoint = ' oss - cn - hangzhou . aliyuncs . com '
# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM account , log on to https :// ram . console . aliyun
. com .
access_key _id = '< yourAccess KeyId >'
access_key _secret = '< yourAccess KeySecret >'
bucket_nam e = '< yourBucket Name >'
# role_arn is the resource name of the role .
role_arn = '< yourRoleAr n >'

clt = client . AcsClient ( access_key _id , access_key _secret ,
' cn - hangzhou ')
req = AssumeRole Request . AssumeRole Request ()

# Set the format of returned values to JSON .
req . set_accept _format ( ' json ' )
req . set_RoleAr n ( role_arn )
req . set_RoleSe ssionName ( ' session - name ' )
body = clt . do_action ( req )

# Use the AccessKeyI d and AccessKeyS ecret to apply
for a temporary token from STS .
token = json . loads ( body )

# Use the authentica tion informatio n in the temporary
token to initialize the StsAuth instance .
auth = oss2 . StsAuth ( token [ ' Credential s ' ] [ ' AccessKeyI d
' ],
                    token [ ' Credential s ' ] [ ' AccessKeyS ecret ' ],
                    token [ ' Credential s ' ] [ ' SecurityTo ken ' ])

# Use the StsAuth instance to initialize the bucket .
bucket = oss2 . Bucket ( auth , endpoint , bucket_nam e )

# Upload a string .
bucket . put_object ( ' object - name . txt ' , b ' hello world ' )

```

3.13 Enable logging in Python SDK

OSS Python SDK provides a logging function to easily track problems. This function is disabled by default.

With this function, you can locate and collect log information about OSS operations and save the information as log files in local disks.

**Note:**

The logging function is only available in OSS Python SDK 2.6.x and later.

The format of a log file is as follows:

```
< time >< name >< level >< threadId >< message >
```

Log levels can be arranged in the following order of severity: CRITICAL > ERROR > WARNING > INFO > DEBUG > NOTSET.

**Note:**

After you specify a log level, only log information of levels that are equal to or higher than the specified level is recorded in local log files. For example, if you specify the log level as INFO, log information of levels CRITICAL, ERROR, WARNING, and INFO, is recorded in log files.

Enable logging in Python SDK

To enable the logging function in the OSS Python SDK, run the following code:

```
# -*- coding : utf - 8 -*-

import os
import logging
import oss2
from itertools import islice

# Initialize information such as AccessKeyId ,
AccessKeySecret , and Endpoint .
# Replace < AccessKeyId > , < AccessKeySecret > , < Bucket > , and
< Endpoint > with actual values .
access_key_id = '< AccessKeyId >'
access_key_secret = '< AccessKeySecret >'
bucket_name = '< Bucket >'
endpoint = '< Endpoint >'

log_file_path = "log.log"
# Enable the logging function .
oss2.set_file_logger ( log_file_path , 'oss2' , logging .
INFO )

# Create a bucket .
bucket = oss2 . Bucket ( oss2 . Auth ( access_key_id ,
access_key_secret ) , endpoint , bucket_name )

# Traverse all objects in the bucket .
for b in islice ( oss2 . ObjectIterator ( bucket ) , 10 ) :
    print ( b . key )
# Obtain the object meta .
```

```
object_meta = bucket.get_object_meta('object')
```

The recorded log information is as follows:

```
2018-11-20 16:37:46,325 oss2.api [INFO] 26716 :
Init oss bucket, endpoint: oss-cn-shenzhen.aliyuncs.
com, isCname: False, connect_timeout: None, app_name: ,
enabled_crc: True
2018-11-20 16:37:46,326 oss2.api [INFO] 26716
: Start to List objects, bucket: funrily-hello, prefix
: , delimiter: , marker: , max_keys: 100
2018-11-20 16:37:46,430 oss2.api [INFO] 26716 :
List objects done, req_id: 5BF3C7DA23 6B3A201CE6 45F7,
status_code: 200
2018-11-20 16:37:46,430 oss2.api [INFO] 26716 :
Start to get object metadata, bucket: bucket-hello,
key: object
2018-11-20 16:37:46,437 oss2.api [ERROR] 26716
: Exception: {
  'status': 404,
  'x-oss-request-id': '5BF3C7DA23 6B3A201CE6 4679',
  'details': {
    'HostId': 'bucket-hello.oss-cn-shenzhen.aliyuncs.
com',
    'Message': 'The specified key does not exist.',
    'Code': 'NoSuchKey',
    'RequestId': '5BF3C7DA23 6B3A201CE6 4679',
    'Key': 'object'
  }
}
```

According to the preceding log information, all objects in a bucket are traversed when the bucket is initialized. If a requested object does not exist, a `NoSuchKey` error occurs.

3.14 Data encryption

3.14.1 Client-side encryption

Client-side encryption is performed locally on data before the data is uploaded to OSS.



Notice:

- Only objects smaller than 5 GB can be encrypted at the client side.
- After uploading an object encrypted at the client side, you are not allowed to call the `update_object_meta` or [#unique_210](#) interface to modify the metadata of the object.

For the complete code of client-side encryption, see [GitHub](#).

You can use the following two methods to manage keys:

- Manually manage keys (RSA).
- Use Key Management Service (KMS) to manage keys.

Using the two methods, you can prevent the keys from being leaked and therefore ensure the security of data on the client. Additionally, others cannot decrypt the data to obtain the original data even the data is leaked.

For detailed information about client-side encryption, see [Introduction to client-side encryption SDK](#) in the OSS Developer Guide.

Encryption meta information

Parameter	Description	Required or not
x-oss-meta-oss-crypto-key	Specifies the encrypted key, which is a string encoded with Base64 after being encrypted by the RSA algorithm.	Required
x-oss-meta-oss-crypto-start	Specifies the initial value of the encrypted data, which is generated randomly. The initial value is a string encoded with Base64 after being encrypted by the RSA algorithm.	Required
x-oss-meta-oss-cek-alg	Specifies the encryption algorithm with the following available values: AES, GCM, and NoPadding.	Required
x-oss-meta-oss-wrap-alg	Specifies the encryption algorithm with the following available values: rsa and kms.	Required
x-oss-meta-oss-matdesc	Specifies the description of the Content Encryption Key (CEK) in the JSON format. This parameter does not take effect currently.	Optional

Parameter	Description	Required or not
x-oss-meta-unencrypted-content-length	Specifies the length of data before encryption. This parameter is not generated if the content-length is not specified.	Optional
x-oss-meta-unencrypted-content-md5	Specifies the MD5 of the data before encryption. If the MD5 is not specified, this parameter is not generated.	Optional

Upload and download objects with the keys manually managed

You can upload and download objects with the keys manually managed. The code is as follows:

```
# -*- coding : utf - 8 -*-
import os
import oss2
from oss2.crypto import LocalRsaProvider
# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permission s on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenanc e . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# Create a bucket and use the RSA method to
# encrypted the data . This method
bucket = oss2 . CryptoBucket ( auth , '< yourEndpoi nt >', '<
yourBucket Name >', crypto_pro vider = LocalRsaPr ovider ())
key = ' motto . txt '
content = b ' a ' * 1024 * 1024
filename = ' download . txt '
# Upload an object .
bucket . put_object ( key , content , headers ={' content - length
': str ( 1024 * 1024 )})
# Download an object .
result = bucket . get_object ( key )
# Perform verificati on .
content_go t = b ''
for chunk in result :
    content_go t += chunk
assert content_go t == content
# Download an object to a local file .
result = bucket . get_object _to_file ( key , filename )
# Perform verificati on .
with open ( filename , ' rb ' ) as fileobj :
    assert fileobj . read () == content
os . remove ( filename )
```

Upload and download objects with the keys managed by KMS

· Prerequisites

1. You have activated the Alibaba Cloud Key Management Service.
2. The key ID for your region has been generated.
3. You have created the custom authorization policies as follows: Log on to the RAM console, select > RAM > Policies > Custom Policy > Create Authorization Policy, and associate corresponding users with the policy.

Refer to the following format to create a custom policy:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey",
        "kms:GenerateDataKey",
        "kms:ListKeys",
        "kms:Encrypt",
        "kms:Decrypt"
      ],
      "Resource": [
        "acs:kms:*:yourRegion:*:yourloginUserId:key/*"
      ]
    }
  ]
}
```

· Sample code

```
# -*- coding : utf - 8 -*-
import os
import oss2
from oss2.crypto import AliKMSProvider
# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS. We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenance. To
# create a RAM account, log on to https://ram.
# console.aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
KeySecret >')
# Create a bucket and use the KMS method to
# encrypt the data. This method only applies to
# scenarios where objects are uploaded or downloaded
# entirely.
bucket = oss2.CryptoBucket(auth, '< yourEndpoint >',
'liu5iman123456', crypto_provider = AliKMSProvider('<
yourAccessKeyId >', '< yourAccessKeySecret >', '< yourRegion
>', '< yourCMK >', '1234'))
key = 'motto.txt'
```



```

content = b ' a ' * 1024 * 1024
filename = ' download . txt '
# Upload an object .
bucket . put_object ( key , content , headers ={' content -
length ': str ( 1024 * 1024 )})
# Download an object .
result = bucket . get_object ( key )
# Perform verificati on .
content_go t = b ''
for chunk in result :
    content_go t += chunk
assert content_go t == content
# Download an object to a local file .
result = bucket . get_object _to_file ( key , filename )
# Perform verificati on .
with open ( filename , ' rb ' ) as fileobj :
    assert fileobj . read () == content
os . remove ( filename )

```

3.14.2 Server-side encryption

OSS supports server-side encryption for uploaded data. This means that when user data is uploaded, OSS encrypts the data and permanently stores the data. Then, when the data is downloaded by a user, OSS automatically decrypts the data, returns the original data to the user, and declares in the header of the response that the data has been encrypted on the server.

The following server-side encryption methods are available for different application scenarios:

- Server-side encryption that uses CMKs managed by KMS for encryption and decryption (SSE-KMS)

When uploading an object, you can use a specified CMK ID or the default CMK managed by KMS to encrypt and decrypt a large amount of data. This method is cost-effective because you do not need to send user data to the KMS server through networks for encryption and decryption.



Notice:

API calling fees are incurred if you use CMKs to encrypt or decrypt data.

- Server-side encryption fully managed by OSS (SSE-OSS)

When you upload an object, OSS encrypts the object on the server side by using AES256 fully managed by OSS. In this method, OSS uses AES256 to encrypt each object with an individual key. Furthermore, the individual keys are encrypted by a CMK that is updated periodically for higher security. This method applies to encrypt or decrypt bulk data.

**Notice:**

- Only one server-side encryption method can be used for an object at one time.
- If you configure server-side encryption for a bucket, you can still configure the encryption method for a single object when uploading or copying it. In this case, the encryption method configured for the object takes precedence. For more information, see [PutObject](#).
- For more information about server-side encryption, see [Server-side encryption](#).

Configure server-side encryption for a bucket

You can run the following code to configure the default encryption method for a bucket. After the method is configured, all objects that are uploaded to the bucket without their encryption methods configured are encrypted in the configured default encryption method.

```
# -*- coding : utf - 8 -*-
import oss2
from oss2 . models import ServerSide Encryption Rule

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Configures the encryption method for the bucket . In
this example , AE256 is specified as an example .
rule = ServerSide Encryption Rule ()
rule . sse_algori thm = oss2 . SERVER_SID E_ENCRYPTI ON_AES256
# Configures the CMK ID when the encryption method is
set to KMS . To encrypt data by using a specified
CMK , input the specified CMK ID . If you use the
CMK managed by OSS to encrypt data , set this
parameter to null . If you use the AES256 method to
encrypt data , this parameter must be set to null .
rule . kms_master _keyid = ""

# Configures server - side encryption for the bucket .
result = bucket . put_bucket _encryptio n ( rule )
# Checks the returned HTTP status code .
```

```
print (' http   response   code :', result . status )
```

Obtain the server-side encryption settings of a bucket

You can run the following code to obtain the server-side encryption settings of a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess   KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket   Name >')

# Obtains the server - side encryption settings of the
bucket .
result = bucket . get_bucket _encryptio n ()
# Prints the obtained server - side encryption settings .
print (' sse_algori thm :', result . sse_algori thm )
print (' kms_master _keyid :', result . kms_master _keyid ) # If
the bucket encryption method is set to AES256 , the
value of kms_master _keyid is None .
```

Delete the server-side encryption settings of a bucket

You can run the following code to delete the server-side encryption settings of a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with the AccessKey
of an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM user account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess   KeyId >', '< yourAccess
KeySecret >')
auth = oss2 . Auth ('< yourAccess   KeyId >', '< yourAccess
KeySecret >')
# This example uses the China East 1 ( Hangzhou )
endpoint . Specify the actual endpoint based on your
requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket   Name >')
```

```
# Deletes the server-side encryption settings of the
bucket .
result = bucket . delete_bucket_encryption ()
# Checks the returned HTTP status code .
print (' http status :', result . status )
```

3.15 Static website hosting

You can set your bucket configuration to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

For more information about static website hosting, see [Static website hosting](#).

Configure static website hosting

Run the following code to configure static website hosting:

```
# -*- coding : utf - 8 -*-
import oss2
from oss2 . models import BucketWebsite

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Enable the static website hosting mode . Set the
index page to index . html and the error page to
error . html .
bucket . put_bucket_website ( BucketWebsite (' index . html ', '
error . html '))
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
```

```

auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

try :
# If the static website hosting mode is not enabled
, a NoSuchWebs ite exception is thrown by get_bucket
_website .
    website = bucket . get_bucket _website ()
    print (' Index file is { 0 }, error file is { 1 }'.
format ( website . index_file , website . error_file ))
except oss2 . exceptions . NoSuchWebs ite as e :
    print (' Website is not configured , request_id ={ 0 }'.
format ( e . request_id ))

```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```

# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . delete_buc ket_websit e ()

```

3.16 Manage lifecycle rules

You can use OSS to configure lifecycle rules to reduce storage costs. The lifecycle management rules can be used for automatic deletion of expired objects and fragments, or storage class conversion to Infrequent Access for objects that will expire soon. Each rule includes:

- **Rule ID:** It identifies a rule. Ensure that each rule ID in a bucket is unique.

- **Policy:** You can configure a policy by using the following configuration methods. Only one method can be configured for a bucket.
 - **Configure by Prefix:** You can create multiple rules with this method. Ensure that each prefix is unique.
 - **Configure for Entire Bucket:** You can configure only one rule with this method.
- **Expired:** You can configure the following configuration methods:
 - **Set by Number of Days:** It specifies the number of days from the last modification time of objects.
 - **Set by Date:** It specifies the date prior to which objects are created. If objects are created prior to the date, these objects are deleted.
- **Status:** This lifecycle rule takes effect or not.

The parts uploaded by `upload_part` method also support lifecycle rules. The last modification time of an object is the time the multipart upload event is initiated.

For more information about lifecycle management, see [Manage object lifecycle](#).

Configure lifecycle rules

Run the following code to configure lifecycle rules:

```
# -*- coding : utf - 8 -*-
import oss2
from oss2.models import LifecycleE xpiration , LifecycleR
ule , BucketLife cycle , AbortMulti partUpload
import datetime

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This case uses the Hangzhou endpoint as an example
. Fill in the region endpoint name according to the
actual circumstan ces .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# Set the expiration duration to 3 days after last
modified date .
rule1 = LifecycleR ule (' rule1 ', ' tests /',
                        status = LifecycleR ule . ENABLED ,
                        expiration = LifecycleE xpiration ( days = 3
))

# Configure expiration date to delete objects that are
created before the date .
```

```

rule2 = LifecycleRule('rule2', 'logging-',
                      status=LifecycleRule.DISABLED,
                      expiration=LifecycleExpiration(created_before_date=datetime.
date(2018, 12, 12)))

# Set the expiration duration to 3 days for parts
in an object.
rule3 = LifecycleRule('rule3', 'tests1/',
                      status=LifecycleRule.ENABLED,
                      abort_multipart_upload=AbortMultipartUpload(
days=3))

# Configure expiration date to delete parts that are
created before the date.
rule4 = LifecycleRule('rule4', 'logging1-',
                      status=LifecycleRule.DISABLED,
                      abort_multipart_upload=AbortMultipart
partUpload(created_before_date=datetime.date(2018, 12,
12)))

lifecycle = BucketLifecycle([rule1, rule2, rule3, rule4])
bucket.put_bucket_lifecycle(lifecycle)

```

View lifecycle rules

Run the following code to view lifecycle rules:

```

# -*- coding: utf-8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance. To
create a RAM account, log on to https://ram.console
.aliyun.com.
auth = oss2.Auth('< yourAccessKeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China (Hangzhou). Specify
the actual endpoint based on your requirements.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
aliyuncs.com', '< yourBucketName >')

lifecycle = bucket.get_bucket_lifecycle()

for rule in lifecycle.rules:
    if rule.abort_multipart_upload is None:
        print('id={0}, prefix={1}, status={2}, days={
3}, created_before_date={4}'
              .format(rule.id, rule.prefix, rule.status,
rule.expiration.days,
rule.expiration.created_before_date))
    else:
        print('id={0}, prefix={1}, status={2}, days={
3}, created_before_date={4}'
              .format(rule.id, rule.prefix, rule.status,
rule.abort_multipart_upload.days,

```

```
rule . abort_mult ipart_uplo ad . created_be  
fore_date ))
```

Clear lifecycle rules

Run the following code to clear lifecycle rules:

```
# -*- coding : utf - 8 -*-  
import oss2  
  
# It is highly risky to log on with AccessKey of  
an Alibaba Cloud account because the account has  
permissions on all the APIs in OSS . We recommend  
that you log on as a RAM user to access APIs  
or perform routine operations and maintenanc e . To  
create a RAM account , log on to https :// ram . console  
. aliyun . com .  
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess  
KeySecret >')  
# This example uses endpoint China ( Hangzhou ) . Specify  
the actual endpoint based on your requiremen ts .  
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .  
aliyuncs . com ', '< yourBucket Name >')  
  
bucket . delete_buc ket_lifecy cle ()  
  
# An exception occurs when you view lifecycle rules  
again .  
try :  
    lifecycle = bucket . get_bucket _lifecycle ()  
except oss2 . exceptions . NoSuchLife cycle :  
    print (' lifecycle is not configured ')
```

3.17 CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) and [PutBucketcors](#) in the OSS Developer Guide.

Configure CORS rules

Run the following code to configure CORS rules for the specified bucket:

```
# -*- coding : utf - 8 -*-  
import oss2  
from oss2 . models import BucketCors , CorsRule  
  
# It is highly risky to log on with AccessKey of  
an Alibaba Cloud account because the account has  
permissions on all the APIs in OSS . We recommend  
that you log on as a RAM user to access APIs  
or perform routine operations and maintenanc e . To
```



```

create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

rule = CorsRule ( allowed_or_igins =['*'],
                  allowed_me thods =[' GET ', ' HEAD '],
                  allowed_he aders =['*'],
                  max_age_se conds = 1000 )

# The existing rules will be replaced .
bucket . put_bucket _cors ( BucketCors ([ rule ]))

```

Obtain CORS rules

Run the following code to obtain CORS rules:

```

# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

try :
    cors = bucket . get_bucket _cors ()
except oss2 . exceptions . NoSuchCors :
    print (' cors is not set ')
else :
    for rule in cors . rules :
        print (' AllowedOri gins ={ 0 }'. format ( rule .
allowed_or_igins ))
        print (' AllowedMet hods ={ 0 }'. format ( rule .
allowed_me thods ))
        print (' AllowedHea ders ={ 0 }'. format ( rule .
allowed_he aders ))
        print (' ExposeHead ers ={ 0 }'. format ( rule . expose_he
aders ))

```

```
print (' MaxAgeSeco nds ={ 0 }'. format ( rule . max_age_se
conds ))
```

Delete CORS rules

Run the following code to delete all CORS rules for the specified bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requiremen
ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

bucket . delete_buc ket_cors ()
```

3.18 Set logging

You can enable access logs to record bucket access to log files, which are stored in a specified bucket.

The log file format is as follows:

```
< TargetPref ix >< SourceBuck et >- YYYY - mm - DD - HH - MM - SS -
UniqueStri ng
```

For more information about access log files, see [Set access logging](#).

Enable access logging

Run the following code to enable bucket access logging:

```
# -*- coding : utf - 8 -*-
import oss2
from oss2 . models import BucketLogg ing

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
```

```
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >' )

# Enable access logging . Store log files in the
current bucket and set the log file storage directory
to ' logging /'.
logging = bucket . put_bucket _logging ( BucketLogg ing ( bucket
. bucket_nam e , ' logging /' ))
if logging . status == 200 :
    print ( " Enable access logging " )
else :
    print ( " request_id :", logging . request_id )
    print ( " resp :", logging . resp . response )
```

View access logging configurations

Run the following code to view the access logging configurations for a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ( '< yourAccess KeyId >', '< yourAccess
KeySecret >' )
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >' )

logging = bucket . get_bucket _logging ()
print ( ' TargetBuck et={ 0 }, TargetPref ix={ 1 }'. format (
logging . target_buc ket , logging . target_pre fix ))
```

Disable access logging

Run the following code to disable access logging for a bucket:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ( '< yourAccess KeyId >', '< yourAccess
KeySecret >' )
# This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >' )
```

```
logging = bucket.delete_bucket_logging()
if logging.status == 204:
    print("Disable access logging")
else:
    print("request_id :", logging.request_id)
    print("resp :", logging.resp.response)
```

3.19 Anti-leech

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leeching, see [Anti-leeching settings](#) in the OSS Developer Guide.

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```
# -*- coding : utf - 8 -*-
import oss2
from oss2.models import BucketReferer

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance. To
create a RAM account, log on to https://ram.console
.aliyun.com.
auth = oss2.Auth('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 (Hangzhou).
Specify the actual endpoint based on your requiremen
ts.
bucket = oss2.Bucket(auth, 'http://oss-cn-hangzhou.
aliyuncs.com', '< yourBucket Name >')

# Configure to allow empty Referers (True indicates
that an empty Referer is allowed, and False
indicates that an empty Referer is not allowed), and
configure the referer whitelist.
```

```
bucket . put_bucket _referer ( BucketReferer ( True , [' http ://
aliyun . com ', ' http ://*. aliyuncs . com ']))
```

Obtain a referer whitelist

Run the following code to obtain a referer whitelist:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

config = bucket . get_bucket _referer ()
print (' allow empty referer ={ 0 }, referers ={ 1 }'. format (
config . allow_empty_referer , config . referers ))
```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
Specify the actual endpoint based on your requirements .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

# You cannot clear a referer whitelist directly . To
clear a referer whitelist , you need to create the
rule that allows an empty referer field and replace
the original rule with the new rule .
```

```
bucket . put_bucket _referer ( BucketReferer ( True , []))
```

3.20 Image processing

Image Processing (IMG) is a massive, secure, cost-effective, and highly reliable image processing service. After source images are uploaded to OSS, you can process images on any Internet device at any anytime and from anywhere through simple RESTful APIs.

For more information about IMG, see [Image Processing](#).

For the complete code of IMG, see [GitHub](#).

Basic features

OSS provides the following IMG features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Image processing](#)
- [Image processing access rules](#): calls multiple IMG functions.

Usage

IMG uses standard HTTP GET requests. You can configure IMG parameters in the QueryString included in the URL of an image.

If the ACL for an image is private read and write, only authorized users are allowed can access the image.

- Anonymous access

You can use a level-3 domain name in the following format to access a processed image:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >?  
>? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Specifies the name of a bucket.
endpoint	Specifies the endpoint used to access the region that the bucket is created.
object	Specifies the name of an image object.
image	Specifies the reserved identifier for IMG .
action	Specifies the operation performed on an image, such as scaling, cropping, and rotating.

Parameter	Description
param	Specifies the parameter that indicates the operation performed on an image.

- Basic operations

For example, scale an image to a width of 100 px and adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customize image styles

You can use a level-3 domain name in the following format to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

■ **style**: specifies the reserved identifier of a customized image style.

■ **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image in sequence.

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Access through HTTPS

IMG allows you to access images through the HTTPS protocol. For example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

Authorized access allows you to customize an image style, access images through the HTTPS protocol, and perform cascade operations.

Run the following code to generate a signed URL for IMG:

```
# -*- coding : utf - 8 -*-
import oss2

# This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your requirements .
endpoint = ' http://oss-cn-hangzhou.aliyuncs.com '
# It is highly risky to log on with AccessKey of an Alibaba Cloud account because the account has permissions on all APIs in OSS . We strongly recommend that you create a RAM user and use it for API access and daily operations and maintenance . Log on to https://ram.console.aliyun.com to create a RAM user .
access_key_id = '< yourAccess KeyId >'
access_key_secret = '< yourAccess KeySecret >'
bucket_name = '< yourBucket Name >'
key = ' example . jpg '

# Creates a bucket instance . All object - related methods must be called by the bucket instance .
bucket = oss2 . Bucket ( oss2 . Auth ( access_key_id , access_key_secret ), endpoint , bucket_name )

# Uploads a sample image .
bucket . put_object _from_file ( key , ' example . jpg ' )

# Generates a signed URL and sets the expiration time to 10 minutes . The unit of expiration time is second .
style = ' image / resize , m_fixed , w_100 , h_100 / rotate , 90 '
url = bucket . sign_url ( ' GET ' , key , 10 * 60 , params = { ' x - oss - process ' : style } )
```

```
print ( url )
```

- Access image objects with SDK

You can use OSS SDK to access and process an image object.

SDK allows you to customize image styles, access images through the HTTPS protocol, and perform cascade operations.

- Basic operations

The `get_object` and `get_object_to_file` methods support image processing.

Use the following code to perform basic operations on an image:

```
# -*- coding : utf - 8 -*-
import os
import oss2

# This example uses endpoint China East 1 ( Hangzhou
# ). Specify the actual endpoint based on your
# requiremen ts .
endpoint = ' http :// oss - cn - hangzhou . aliyuncs . com '
# It is highly risky to log on with AccessKey
# of an Alibaba Cloud account because the account
# has permission s on all APIs in OSS . We strongly
# recommend that you create a RAM user and use
# it for API access and daily operations and
# maintenanc e . Log on to https :// ram . console . aliyun
# . com to create a RAM user .
access_key _id = '< yourAccess KeyId >'
access_key _secret = '< yourAccess KeySecret >'
bucket_nam e = '< yourBucket Name >'
key = ' example . jpg '
new_pic = ' example - new . jpg '

# Creates a bucket instance . All object - related
# methods must be called by the bucket instance .
bucket = oss2 . Bucket ( oss2 . Auth ( access_key _id ,
# access_key _secret ), endpoint , bucket_nam e )

# Uploads a sample image .
bucket . put_object _from_file ( key , ' example . jpg ' )

# Scales the image .
style = ' image / resize , m_fixed , w_100 , h_100 '
bucket . get_object _to_file ( key , new_pic , process = style
# )

# Crops the image .
style = ' image / crop , w_100 , h_100 , x_100 , y_100 , r_1 '
bucket . get_object _to_file ( key , new_pic , process = style
# )

# Rotates the image .
style = ' image / rotate , 90 '
bucket . get_object _to_file ( key , new_pic , process = style
# )

# Sharpens the image .
style = ' image / sharpen , 100 '
```

```

bucket . get_object _to_file ( key , new_pic , process = style
)

# Adds watermarks .
style = ' image / watermark , text_SGVsb G8g5Zu - 54mH5pyN5Y
qhIQ '
bucket . get_object _to_file ( key , new_pic , process = style
)

# Converts the image format .
style = ' image / format , png '
bucket . get_object _to_file ( key , new_pic , process = style
)

# Deletes the sample image .
bucket . delete_obj ect ( key )
# Clears local files .
os . remove ( new_pic )

```

- Customize an image style

You can run the following code to customize an image style:

```

# -*- coding : utf - 8 -*-
import os
import oss2

# This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
endpoint = ' http :// oss - cn - hangzhou . aliyuncs . com '
# It is highly risky to log on with AccessKey
of an Alibaba Cloud account because the account
has permission s on all APIs in OSS . We strongly
recommend that you create a RAM user and use
it for API access and daily operations and
maintenanc e . Log on to https :// ram . console . aliyun
. com to create a RAM account .
access_key _id = '< yourAccess KeyId >'
access_key _secret = '< yourAccess KeySecret >'
bucket_nam e = '< yourBucket Name >'
key = ' example . jpg '
new_pic = ' example - new . jpg '

# Creates a bucket instance . All object - related
methods must be called by the bucket instance .
bucket = oss2 . Bucket ( oss2 . Auth ( access_key _id ,
access_key _secret ), endpoint , bucket_nam e )

# Uploads a sample image .
bucket . put_object _from_file ( key , ' example . jpg ' )

# Customizes the image style .
style = ' style / oss - pic - style - w - 100 '

# Processes the image .
bucket . get_object _to_file ( key , new_pic , process = style
)

# Deletes the sample image .
bucket . delete_obj ect ( key )
# Clears local files .

```

```
os . remove ( new_pic )
```

- Cascade operations

Run the following code to perform cascade operations on an image:

```
# -*- coding : utf - 8 -*-
import os
import oss2

# This example uses endpoint China East 1 ( Hangzhou ). Specify the actual endpoint based on your
# requirements .
endpoint = ' http :// oss - cn - hangzhou . aliyuncs . com '
# It is highly risky to log on with AccessKey
# of an Alibaba Cloud account because the account
# has permissions on all APIs in OSS . We strongly
# recommend that you create a RAM user and use
# it for API access and daily operations and
# maintenance . Log on to https :// ram . console . aliyun
# . com to create a RAM user .
access_key _id = '< yourAccess KeyId >'
access_key _secret = '< yourAccess KeySecret >'
bucket_name = '< yourBucket Name >'
key = ' example . jpg '
new_pic = ' example - new . jpg '

# Creates a bucket instance . All object - related
# methods must be called by the bucket instance .
bucket = oss2 . Bucket ( oss2 . Auth ( access_key _id ,
# access_key _secret ), endpoint , bucket_name )

# Uploads a sample image .
bucket . put_object _from_file ( key , ' example . jpg ' )

# Performs cascade operations .
style = ' image / resize , m_fixed , w_100 , h_100 / rotate , 90
'

# Processes the image .
bucket . get_object _to_file ( key , new_pic , process = style
)

# Deletes the sample image .
bucket . delete_obj ect ( key )
# Clears local files .
os . remove ( new_pic )
```

Image processing persistence

OSS provides the “saveas” operation for data processing. With this feature, you can save the processing result to a specified bucket as resources and assign the result with a specified key. After the resources are saved, you can access the resources directly by specifying the bucket to speed up resource download. This feature applies to ultra-large image cropping or other high-latency operations.

Run the following code for image processing persistence:

```
# -*- coding : utf - 8 -*-
import os
import base64
import oss2

# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requirements .
endpoint = ' http :// oss - cn - hangzhou . aliyuncs . com '
# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all APIs in OSS . Instead , we
# strongly recommend that you create a RAM user and
# use it for API access and daily operations and
# maintenance . Log on to https :// ram . console . aliyun .
# com to create a RAM account .
# source_bucket_name and target_bucket_name are the same
# bucket .
access_key_id = '< yourAccess KeyId >'
access_key_secret = '< yourAccess KeySecret >'
source_bucket_name = '< yourBucket Name >'
target_bucket_name = '< yourTarget BucketName >'
source_image_name = ' example . jpg '

# Creates a bucket instance . All object - related methods
# must be called by the bucket instance .
bucket = oss2 . Bucket ( oss2 . Auth ( access_key_id ,
access_key_secret ) , endpoint , source_bucket_name )

# Scales the image .
style = ' image / resize , m_fixed , w_100 , h_100 '
target_image_name = ' example - resize . jpg '
process = "{ 0 }| sys / saveas , o_ { 1 } , b_ { 2 }". format ( style
,
oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_image_name ) ) ) ,
oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_bucket_name ) ) ) )
result = bucket . process_object ( source_image_name , process
)
print ( result )

# Crops the image .
style = ' image / crop , w_100 , h_100 , x_100 , y_100 , r_1 '
target_image_name = ' example - crop . jpg '
process = "{ 0 }| sys / saveas , o_ { 1 } , b_ { 2 }". format ( style
,
oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_image_name ) ) ) ,
oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_bucket_name ) ) ) )
result = bucket . process_object ( source_image_name , process
)
print ( result )

# Rotates the image .
style = ' image / rotate , 90 '
target_image_name = ' example - rotate . jpg '
process = "{ 0 }| sys / saveas , o_ { 1 } , b_ { 2 }". format ( style
,
```

```

        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_image_name ))) ,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_bucket_name )))
result = bucket . process_object ( source_image_name , process
)
print ( result )

# Sharpens the image .
style = ' image / sharpen , 100 '
target_image_name = ' example - sharpen . jpg '
process = "{ 0 } | sys / saveas , o_ { 1 } , b_ { 2 } ". format ( style
,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_image_name ))) ,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_bucket_name )))
result = bucket . process_object ( source_image_name , process
)
print ( result )

# Adds watermarks .
style = ' image / watermark , text_SGVsb G8g5Zu - 54mH5pyN5Y qhIQ
,
target_image_name = ' example - watermark . jpg '
process = "{ 0 } | sys / saveas , o_ { 1 } , b_ { 2 } ". format ( style
,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_image_name ))) ,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_bucket_name )))
result = bucket . process_object ( source_image_name , process
)
print ( result )

# Converts the image format .
style = ' image / format , png '
target_image_name = ' example - formatconvert . png '
process = "{ 0 } | sys / saveas , o_ { 1 } , b_ { 2 } ". format ( style
,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_image_name ))) ,
        oss2 . compat . to_string ( base64 . urlsafe_b64encode ( oss2 .
compat . to_bytes ( target_bucket_name )))
result = bucket . process_object ( source_image_name , process
)

```

```
print ( result )
```

3.21 Chinese character and time

This topic describes the Chinese characters and time used by OSS Python SDK.

Chinese character

Errors may occur if Chinese characters are used in the Python code. Therefore, you must declare the Chinese character encoding at the beginning of the code. For example:

```
# -*- coding : utf - 8 -*-
```

· Data type

Python 2.x supports the following two data types:

Data type	Description
str	Character strings, corresponding to the bytes type in Python 3.x
unicode	Unicode streams, of which the length is measured in characters. For example, the length of <code>u '中文'</code> is 2.

Python 3.x supports the following two data types:

Data type	Description
str	Character strings, corresponding to the unicode type in Python 2.x
bytes	Byte streams, of which the length is measured in bytes. For example, the length of <code>b '中文'</code> depends on the encoding, which is 6 if it is encoded with UTF-8.

- Input and output conventions

The following table describes the conventions of input parameter types for OSS Python SDK.

Input	Data type	Remarks
OSS object name	str	If the input type is bytes, it must be encoded with UTF-8.
Local file name	str, unicode	If the input type is bytes , it must be encoded with UTF-8, such as the yourLocalFile parameter in bucket.get_object_to_file.
Input data stream	bytes	For example, the data parameter in bucket.put_object.

The following table describes the conventions of output parameter types for OSS Python SDK.

Output	Data type	Remarks
Results parsed from the XML	str	For example, a string in the result obtained by bucket.list_object.
Downloaded content	bytes	Python SDK regards that the bytes type is encoded with UTF-8 by default. Therefore, make sure that the Python source file is also encoded with UTF-8.

- Type conversion functions

OSS Python SDK provides three functions for type conversion.

Function	Description
to_bytes	- Converts the unicode type to the str type in Python 2.x. Other types are returned without changes. - Converts the str type to the bytes type in Python 3.x. Other types are returned without changes.
to_unicode	- Converts the str type to the unicode type in Python 2.x. Other types are returned without changes. - Converts the bytes type to the str type in Python 3.x. Other types are returned without changes.
to_string	Acts as to_bytes in Python 2.x. Acts as to_unicode in Python 3.x.

Time

OSS Python SDK converts all timestamp strings retrieved from the server to [Unix Time](#), that is, the number of seconds since UTC 00:00 on January 1, 1970. For example, the last_modified time in the results returned by bucket.get_object is the Unix Time of the int type.

You can use the datetime.datetime.fromtimestamp() method to convert the time, obtaining the timestamp string.

3.22 Exception handling

OSS Python SDK has three types of exceptions: ClientError, RequestError, and ServerError, which are defined in the oss2.exceptions sub-module.

The following table lists the member variables of exceptions.

Variable	Type	Description
status	int	· ServerError: HTTP status code · ClientError and RequestError: fixed values
request_id	str	· ServerError: request ID returned by the OSS server · ClientError and RequestError: empty string
Code and message	str	Correspond to text text in the Code and Message XML Tags in OSS error response .

Example for handling exceptions

The following code demonstrates how to handle exceptions when downloading an object that does not exist, and how to print the HTTP status code and request ID of the error messages.

```
# -*- coding : utf - 8 -*-
import oss2

# It is highly risky to log on with AccessKey of
# an Alibaba Cloud account because the account has
# permissions on all the APIs in OSS . We recommend
# that you log on as a RAM user to access APIs
# or perform routine operations and maintenance . To
# create a RAM account , log on to https :// ram . console
# . aliyun . com .
auth = oss2 . Auth ('< yourAccess KeyId >', '< yourAccess
KeySecret >')
# This example uses endpoint China East 1 ( Hangzhou ).
# Specify the actual endpoint based on your requiremen
# ts .
bucket = oss2 . Bucket ( auth , ' http :// oss - cn - hangzhou .
aliyuncs . com ', '< yourBucket Name >')

try :
    stream = bucket . get_object (' random - key . txt ')
except oss2 . exceptions . NoSuchKey as e :
```

```
print (' status ={ 0 }, request_id ={ 1 }'.format ( e . status
, e . request_id ))
```

ClientError

ClientError is caused by input error on the client side. For example, a ClientError is thrown when an empty object name list is received when the bucket.batch_delete_objects method is used. The status value of ClientError objects is oss2.exceptions.OSS_CLIENT_ERROR_STATUS.

RequestError

When an exception is thrown by the underlying HTTP library, the Python SDK will convert the exception to RequestError. The status value of ClientError objects is fixed to oss2.exceptions.OSS_CLIENT_ERROR_STATUS.

ServerError

When an HTTP error code is returned by the OSS server, Python SDK converts it into a ServerError. ServerError is divided into multiple subclasses based on the HTTP status code and OSS error code. The NotFound subclass corresponds to all 404 exceptions, and the Conflict subclass corresponds to all 409 exceptions.

The following table lists common error codes.

Exception class	HTTP status code	OSS error code	Description
NotModified	304	Null	It is not modified.
AccessDenied	403	AccessDenied	The access is denied.
NoSuchBucket	404	NoSuchBucket	The bucket does not exist.
NoSuchKey	404	NoSuchKey	The file does not exist.
NoSuchUpload	404	NoSuchUpload	The multipart upload does not exist.
NoSuchWebsite	404	NoSuchWebsiteConfiguration	The static website hosting is not configured.
NoSuchLifecycle	404	NoSuchLifecycle	The lifecycle management is not configured.

Exception class	HTTP status code	OSS error code	Description
NoSuchCors	404	NoSuchCORS Configuration	CORS is not configured.
BucketNotEmpty	409	BucketNotEmpty	The bucket is not empty.
PositionNotEqualToLength	409	PositionNotEqualToLength	The Append position is not equal to the object length.
ObjectNotAppendable	409	ObjectNotAppendable	The object is not appendable.

4 Go

4.1 Preface

This topic describes how to use code samples.

SDK source code and API documents

For the source code of OSS Go SDK, see [GitHub](#). For more information, see [OSS Go SDK API documents](#).

Code samples

OSS Go SDK provides a variety of code samples for your reference or use. The following table describes the content of code samples:

Sample file	Content
new_bucket.go	Initialize the client
put_object.go	Upload objects, including simple upload and resumable upload .
append_object.go	Append upload
get_object.go	Download objects, including streaming download , conditional download , download object into compressed objects
delete_object.go	Delete objects
copy_object.go	Copy objects within a bucket , Copy objects across buckets and conditional copy
list_objects.go	List objects , including listing objects with a specified prefix and list a specified number of objects.
object_meta.go	Configure and read Object Meta
object_acl.go	Set and read ACL for an object
sign_url.go	Generate a signed URL
cname_sample.go	Bind custom domain (CNAME)
create_bucket.go	Create a bucket

Sample file	Content
list_buckets.go	List buckets with default or specified parameters
bucket_acl.go	Set ACL for buckets
bucket_referer.go	Configure, view, and delete the anti-leech settings of a bucket
bucket_logging.go	Configure, view, and delete the access logs of a bucket
bucket_lifecycle.go	Configure, view, and delete the lifecycle settings of a bucket
bucket_cors.go	Configure, view, and delete the CORS settings of a bucket

4.2 Installation

Preparation

- Environment requirements

You must use Golang 1.6 and later versions.

Refer to [Installing Go from source](#) to download and install the compiling and running environment of Go. After Go is installed, set the GOPATH to the path of your code. For more information about GOPATH, run `go help gopath`.

- Check the version of Go

Run `go version` to check the version of Go.

Download SDK

- [Download from GitHub](#)
- [Download previous versions](#)

Install SDK

Run the following command to install OSS Go SDK:

```
go get github.com / aliyun / aliyun - oss - go - sdk / oss
```



Note:

No messages are printed to the interface during installation. Please wait. In case of installation time-out, run the command again.

Check SDK version

Run the following code to check the version of OSS Go SDK:

```
package main

import (
    "fmt"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    fmt.Println("OSS Go SDK Version :", oss.Version)
}
```

4.3 Quick start

This topic describes how to use OSS Go SDK to perform routine operations such as bucket creation, object uploads, and object downloads.

Create a bucket

Run the following code to create a bucket:

```
package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func handleError (err error) {
    fmt.Println("Error :", err)
    os.Exit(-1)
}
func main() {
    // This example uses endpoint China (Hangzhou).
    Specify the actual endpoint based on your requirements.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    // It is highly risky to log on with AccessKey
    of an Alibaba Cloud account because the account has
    permissions on all the APIs in OSS. We recommend
    that you log on as a RAM user to access APIs
    or perform routine operations and maintenance. To
    create a RAM account, log on to https://ram.console
    .aliyun.com.
    accessKeyId := "<yourAccessKeyId>"
    accessKeySecret := "<yourAccessKeySecret>"
    bucketName := "<yourBucketName>"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId,
    accessKeySecret)
    if err != nil {
        handleError(err)
    }
    // Create a bucket.
    err = client.CreateBucket(bucketName)
    if err != nil {
```

```

        handleError r ( err )
    }
}

```

For more information about endpoints, see [Regions and endpoints](#).

Upload objects

Run the following code to upload a file to OSS:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func handleError r (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // This example uses endpoint China (Hangzhou).
    // Specify the actual endpoint based on your requirements.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    // It is highly risky to log on with AccessKey
    // of an Alibaba Cloud account because the account has
    // permissions on all the APIs in OSS. We recommend
    // that you log on as a RAM user to access APIs
    // or perform routine operations and maintenance. To
    // create a RAM account, log on to https://ram.console
    // .aliyun.com.
    accessKeyId := "< yourAccessKeyId >"
    accessKeySecret := "< yourAccessKeySecret >"
    bucketName := "< yourBucketName >"
    objectName := "< yourObjectName >"
    localFileName := "< yourLocalFileName >"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId,
        accessKeySecret)
    if err != nil {
        handleError r (err)
    }
    // Obtain the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        handleError r (err)
    }
    // Upload a file.
    err = bucket.PutObjectFromFile(objectName, localFileName)
    if err != nil {
        handleError r (err)
    }
}

```

Download objects

Run the following code to download an object to local:

```

package main

```



```

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func handleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // This example uses endpoint China (Hangzhou).
    Specify the actual endpoint based on your requirements.
    endpoint := "http://oss-cn-hangzhou.aliyuncs.com"
    // It is highly risky to log on with AccessKey
    of an Alibaba Cloud account because the account has
    permissions on all the APIs in OSS. We recommend
    that you log on as a RAM user to access APIs
    or perform routine operations and maintenance. To
    create a RAM account, log on to https://ram.console
    .aliyun.com.
    accessKeyId := "< yourAccess KeyId >"
    accessKeySecret := "< yourAccess KeySecret >"
    bucketName := "< yourBucket Name >"
    objectName := "< yourObjectName >"
    downloadedFileName := "< yourDownloadedFileName >"
    // Create an OSSClient instance.
    client, err := oss.New(endpoint, accessKeyId,
    accessKeySecret)
    if err != nil {
        handleError(err)
    }
    // Obtain the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        handleError(err)
    }
    // Download the object.
    err = bucket.GetObjectToFile(objectName, downloaded
    FileName)
    if err != nil {
        handleError(err)
    }
}

```

List objects

Run the following code to list objects in a specified bucket. You can list a maximum of 100 objects by default.

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func HandleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.

```

```

    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        HandleErro r ( err )
    }
    // Obtain the bucket .
    bucketName := "< yourBucket Name >"
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        HandleErro r ( err )
    }
    // List objects .
    marker := ""
    for {
        lsRes , err := bucket . ListObject s ( oss . Marker (
marker ))
        if err != nil {
            HandleErro r ( err )
        }
        // Print the listed objects . A maximum of 100
objects are listed by default .
        for _ , object := range lsRes . Objects {
            fmt . Println (" Bucket : ", object . Key )
        }
        if lsRes . IsTruncate d {
            marker = lsRes . NextMarker
        } else {
            break
        }
    }
}

```

Delete objects

Run the following code to delete a specified object:

```

package main
import (
    "fmt "
    "os "
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)
func handleErro r ( err error ) {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
func main () {
    // This example uses endpoint China ( Hangzhou ).
    Specify the actual endpoint based on your requiremen
ts .
    endpoint := " http :// oss - cn - hangzhou . aliyuncs . com "
    // It is highly risky to log on with AccessKey
of an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
    accessKeyI d := "< yourAccess KeyId >"
    accessKeyS ecret := "< yourAccess KeySecret >"
    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
}

```

```

// Create an OSSClient instance .
client , err := oss . New ( endpoint , accessKeyId ,
accessKeySecret )
if err != nil {
    handleError ( err )
}
// Obtain the bucket .
bucket , err := client . Bucket ( bucketName )
if err != nil {
    handleError ( err )
}
// Delete the object .
err = bucket . DeleteObject ( objectName )
if err != nil {
    handleError ( err )
}
}

```

4.4 Initialization

OSSClient serves as the OSS Java client to manage OSS resources such as buckets and objects.

To create an OSSClient instance, you need to specify an endpoint. For more information about endpoints, see [Regions and endpoints](#) and [Bind a custom domain name](#).

The following table describes the parameters that you can configure:

Parameter	Description	Default value
UseCname	Specifies whether the CNAME is used.	No
Timeout	Specifies the request timeout time in seconds, including the connection timeout and socket read-write timeout.	30 seconds for the connection timeout and 60 seconds for the read-write timeout
SecurityToken	Specifies the securitytoken for temporary users	Null
EnableMD5	Specifies whether the MD5 verification is enabled . CRC verification is recommended because it is more efficient than MD5 verification.	No
EnableCRC	Specifies whether CRC verification is enabled.	Yes

Parameter	Description	Default value
Proxy	Specifies the proxy server, such as <code>http : // 8 . 8 . 8 : 3128</code>	Null
AuthProxy	Specifies a proxy server that requires an account and a password to login.	Null

Use an OSS domain to create an OSSClient instance

Run the following code to create an OSSClient instance with a domain assigned by OSS:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

Use a custom domain (CNAME) to create an OSSClient instance

For the complete code of custom domain, see [GitHub](#).

Run the following code to create an OSSClient instance with CNAME:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

// oss.UseCname (true) indicates that the CNAME is enabled. CNAME indicates a custom domain bound to a bucket.
client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >", oss.UseCname(true))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```



Notice:

If the CNAME is enabled, you cannot list buckets because the CNAME is bound to a certain bucket.

Use STS to create an OSSClient instance

Run the following code to create an OSSClient instance with Security Token Service (STS):

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >", oss.SecurityToken("< yourSecurityToken >"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

For more information, see [What is RAM and STS](#) and [Authorized access](#).

Set timeout

Run the following code to set connection timeout:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

// oss.Timeout(10, 120) indicates that the HTTP
// connection timeout is set to 10 seconds (the default
// value is 30 seconds) and the HTTP read-write
// timeout is set to 120 seconds (the default value
// is 60 seconds). The value 0 indicates that timeout
// never occurs, which is not recommended.
client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >", oss.Timeout(10, 120))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

CRC verification

CRC verification is enabled by default during uploads and downloads to ensure data integrity. Run the following code to disable CRC verification:



Warning:

We recommended that you do not disable CRC verification. If you disable CRC verification, the data integrity during uploads and downloads is not guaranteed.

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >", oss.EnableCRC(false))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

```
}
```

4.5 Bucket

A bucket serves as a container that stores objects. Objects belong to a bucket.

Create a bucket

For the complete code of creating a bucket, see [GitHub](#).

Run the following code to create a bucket:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Create a bucket. The default storage class and ACL of the created bucket are set to standard and private read respectively.
    err = client.CreateBucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

For more information about bucket naming rules, see naming conventions in [Basic concepts](#). You can specify the ACL and the storage class when you create a bucket, as shown in [Bucket-level permissions](#) and [Storage class](#).

You can run the following code to create a bucket with specified permission:

```
// Create a bucket and set the bucket ACL to public read (which is private by default).
err = client.CreateBucket("< yourBucketName >", oss.ACLPublicRead)
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
```

```
}
```

You can run the following code to creates a bucket of the archive class:

```
err = client . CreateBucket (< yourBucket Name >, oss .
StorageClass ( oss . StorageArchive ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
```

You can run the followign code to create a bucket of the Infrequent Access class:

```
err = client . CreateBucket (< yourBucket Name >, oss .
StorageClass ( oss . StorageIA ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
```

List buckets

For the complete code of listing buckets, see [GitHub](#).

- List all buckets

Buckets are displayed in alphabetical order. You can list all buckets or specify a bucket that meets certain conditions.

Run the following code to list all buckets:

```
package main

import (
    "fmt"
    "os"
    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Create an OSSClient instance .
    client, err := oss . New (< yourEndpoint >, "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // List all buckets .
    marker := ""
    for {
        lsRes, err := client . ListBuckets ( oss . Marker ( marker
))
        if err != nil {
            fmt . Println (" Error :", err )
            os . Exit (- 1 )
        }

        // 100 buckets are listed by default .
```

```

    for _, bucket := range lsRes.Buckets {
        fmt.Println(" Bucket : ", bucket.Name)
    }

    if lsRes.IsTruncated {
        marker = lsRes.NextMarker
    } else {
        break
    }
}
}
}

```

- List buckets with a specified prefix

Run the following code to list buckets with a specified prefix:

```

// List buckets by a specified prefix .
lsRes, err = client.ListBuckets(oss.Prefix("<
yourBucket Prefix >"))
if err != nil {
    fmt.Println(" Error :", err)
    os.Exit(-1)
}
// Print the bucket list .
fmt.Println(" Buckets with prefix : ", lsRes.Buckets
)
for _, bucket := range lsRes.Buckets {
    fmt.Println(" Bucket with prefix : ", bucket.Name
)
}

```

- List buckets by marker

The value of the marker parameter indicates the name of a bucket. Run the following code to list buckets after the specified bucket (the value of marker):

```

// List buckets after the specified marker .
lsRes, err = client.ListBuckets(oss.Marker("<
yourBucket Marker >"))
if err != nil {
    fmt.Println(" Error :", err)
    os.Exit(-1)
}

// Print the bucket list .
fmt.Println(" My buckets with marker :", lsRes.
Buckets)
for _, bucket := range lsRes.Buckets {
    fmt.Println(" Bucket with marker : ", bucket.Name)
}

```

- List a specified number of buckets

Run the following code to list a specified number (maxKeys) of buckets:

```

// Set the number of buckets that can be listed
to 500 . The default value is 100 . The maximum
value is 1,000 .
lsRes, err = client.ListBuckets(oss.MaxKeys(500))
if err != nil {

```



```

    fmt.Println(" Error :", err )
    os.Exit(-1)
}

// Print the bucket list .
fmt.Println(" My buckets max num :", lsRes.Buckets )
for _, bucket := range lsRes.Buckets {
    fmt.Println(" Bucket with maxKeys :", bucket.Name )
}

```

Determine whether a bucket exists

Run the following code to determine whether a specified bucket exists:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }

    // Determine whether a bucket exists .
    isExist, err := client.IsBucketExist("< yourBucket Name >")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
    fmt.Println(" IsBucketExist result : ", isExist )
}

```

Configure an ACL for a bucket

The ACL of a bucket includes the following permissions.

Permission	Description	Value
Private	The bucket owner and the authorized users can read and write objects in the bucket. Other users cannot perform any operation on the objects.	oss.ACLPrivate

Permission	Description	Value
Public read	The bucket owner and the authorized users can read and write objects in the bucket. Other users can only read the objects in the bucket. Authorize this permission with caution.	oss.ACLPublicRead
Public read-write	All users can read and write objects in the bucket. Authorize this permission with caution.	oss.ACLPublicReadWrite

For more information about ACL, see [access control](#) in the developer guide. For the complete code of configure ACL for a bucket, see [GitHub](#).

Use the following code to configure the ACL for a bucket:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Set the ACL for the bucket to public read.
    err = client.SetBucketACL("< yourBucketName >", oss.ACLPublicRead)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Obtain the ACL for a bucket

Use the following code to obtain the ACL for a bucket:

```
package main

import (
    "fmt"
    "os"
)
```

```

" github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Obtain the ACL for the bucket
aclRes , err := client . GetBucketA CL ("< yourBucket Name >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}
fmt . Println (" Bucket ACL :", aclRes . ACL )
}

```

Obtain the region of a bucket

Run the following code to obtain the region (or location) of a bucket:

```

package main

import (
" fmt "
" os "
" github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Obtain the region of the bucket :
loc , err := client . GetBucketL ocation ("< yourBucket Name
>")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}
fmt . Println (" Bucket Location :", loc )
}

```

Obtain bucket information

Use the following code to obtain bucket information:

```

package main

import (
" fmt "
" os "

```

```

" github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Bucket informatio n includes regions ( region or
location ), creation dates ( CreationDa te ), permission s (
ACL ), owners ( Owner ), and storage class ( StorageCla ss
).
res , err := client . GetBucketI nfo ("< yourBucket Name >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}
fmt . Println (" BucketInfo . Location : ", res . BucketInfo .
Location )
fmt . Println (" BucketInfo . CreationDa te : ", res . BucketInfo
. CreationDa te )
fmt . Println (" BucketInfo . ACL : ", res . BucketInfo . ACL )
fmt . Println (" BucketInfo . Owner : ", res . BucketInfo . Owner )
fmt . Println (" BucketInfo . StorageCla ss : ", res . BucketInfo
. StorageCla ss )
fmt . Println (" BucketInfo . ExtranetEn dpoint : ", res .
BucketInfo . ExtranetEn dpoint )
fmt . Println (" BucketInfo . IntranetEn dpoint : ", res .
BucketInfo . IntranetEn dpoint )
}

```

Delete a bucket

Before a bucket is deleted, ensure that all objects in the bucket, and fragments that are generated from multipart upload are deleted.



Note:

To delete the fragments that are generated from multipart upload, use `Bucket.ListMultipartUploads` to list all fragments, and then use `Bucket.AbortMultipartUpload` to delete the fragments. For more information, see [Multipart upload](#).

Run the following code to delete a bucket:

```

package main

import (
    "fmt"
    "os"
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {

```

```
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}

// Delete the bucket .
err = client . DeleteBuck et ("< yourBucket Name >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
}
```

4.6 Single-connection bandwidth throttling

This topic describes how to ensure other applications have sufficient bandwidth when you perform upload and download operations on OSS. You can include the `x-oss-traffic-limit` parameter in your requests and set bandwidth limits.



Note:

For more information about the scenarios and precautions of the single-connection bandwidth throttling feature, see [#unique_47](#) in the OSS Developer Guide.

Configure bandwidth throttling for common uploads and downloads

Use the following code to configure bandwidth throttling when you upload and download objects to and from OSS:

```
package main

import (
    "fmt"
    "os"
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    //
    // You can use the single - connection bandwidth
    // throttling feature when you call the following
    // operations :
    // PutObject / AppendObje ct / PostObject / CopyObject /
    // UploadPart / UploadPart Copy
    // The following code provides examples of how to
    // configure bandwidth throttling when you call these
    // operations .
    //

    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
    KeyId >", "< yourAccess KeySecret >")
    if err != nil {
```

```

    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}

// Obtain the bucket .
bucket, err := client.Bucket("< yourBucket Name >")
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}
// Specify the object to be uploaded .
fd, err := os.Open("< yourLocalFile >")
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}
defer fd.Close()

// Set the upload bandwidth limit . The value of the
parameter must be a number . The value must be
expressed in bit / s .
// 41943040 indicates that the bandwidth limit for
the request is 40 Mbit / s , or 5 MB / s .
var traffic int64 = 41943040
// Example 1 : Configure bandwidth throttling for the
object to be uploaded .
err = bucket.PutObject("< youObjectName >", fd, oss.
TrafficLimitHeader(traffic))
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}

// Example 2 : Configure bandwidth throttling for the
object to be downloaded . newFile indicates the name
of the downloaded object .
err = bucket.GetObjectToFile("< youObjectName >", "
newFile", oss.TrafficLimitHeader(traffic))
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}

// Example 3 : Configure bandwidth throttling for
append upload . In this case , after localFile0 ne is
uploaded , you can append localFileT wo .
localFile0 ne := "< localFile0 ne >"
localFileT wo := "< localFileT wo >"
fd1, err := os.Open(localFile0 ne)
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}
defer fd1.Close()
fd2, err := os.Open(localFileT wo)
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit(- 1 )
}
defer fd2.Close()

var nextPos int64
nextPos, err = bucket.AppendObject("< yourObject Name
>", fd1, nextPos, oss.TrafficLimitHeader(traffic))

```

```

    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
    nextPos, err = bucket.AppendObject("< yourObject Name
>", fd2, nextPos, oss.TrafficLimitHeader(traffic))
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }

    // Example 4 : Configure bandwidth throttling for
    multipart upload of a large object .
    // You can split an object into multiple parts
    based on the object size .
    chunks, err := oss.SplitFileByPartNum(" localFile ", 3 )
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
    // Open the object .
    fd, err = os.Open(" fileName ")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
    defer fd.Close()
    // Initialize the multipart upload and configure
    bandwidth throttling .
    imur, err := bucket.InitiateMultipartUpload("<
yourObject Name >")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
    // Upload parts and set the bandwidth limit .
    var parts [] oss.UploadPart
    for _, chunk := range chunks {
        fd.Seek(chunk.Offset, os.SEEK_SET)
        part, err := bucket.UploadPart(imur, fd, chunk.Size
, chunk.Number, oss.TrafficLimitHeader(traffic))
        if err != nil {
            fmt.Println(" Error :", err )
            os.Exit(-1)
        }
        parts = append(parts, part)
    }
    // Complete the multipart upload .
    _, err = bucket.CompleteMultipartUpload(imur, parts)
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
}

```

Configure bandwidth throttling for uploads and downloads that use signed URLs

Use the following code to configure bandwidth throttling when you upload and download objects from and to OSS by using signed URLs.

```
package main
```

```

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-sdk-go/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Example 1: Use the signed URL to upload an object.
    // Open the object.
    fd, err := os.Open("< localfile >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer fd.Close()

    // Set the upload bandwidth limit. The value of the
    // parameter must be a number. The value must be
    // expressed in bit/s.
    // 41943040 indicates that the bandwidth limit for
    // the request is 40 Mbit/s, or 5 MB/s.
    var traffic int64 = 41943040
    // Obtain the signed URL for uploading an object.
    strURL, err := bucket.SignURL("< yourObjectName >", oss.HTTPPut, 60, oss.TrafficLimitParam(traffic))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Upload the object.
    err = bucket.PutObjectWithURL(strURL, fd)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Example 2: Use the signed URL to download an object.
    // Obtain the signed URL for downloading an object.
    strURL, err = bucket.SignURL("< yourObjectName >", oss.HTTPGet, 60, oss.TrafficLimitParam(traffic))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```



```
// Download the object . newFile indicates the name of
// the downloaded object .
err = bucket . GetObjectT oFileWithU RL ( strURL , " newFile ")
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
}
```

4.7 Bucket tagging

By using the bucket tagging function, you can classify your buckets to manage them. For example, you can specify that only buckets with specified tags are returned in ListBucket operations.

- Only the bucket owner and authorized RAM users can add tags to a bucket. Otherwise, the 403 Forbidden error is returned with the error code: AccessDenied.
- You can add a maximum of 20 tags (key-value pairs) to a bucket.
- The maximum size of a key is 64 bytes. The key of a tag cannot be null and cannot be prefixed with `http ://`, `https ://`, or `Aliyun` .
- The maximum size of a tag value is 128 bytes. The value of a tag can be null.
- The key and value of a tag must be UTF-8 encoded.
- If you use PutBucketTagging to add tags to a bucket, the original tags added to the bucket are completely overwritten.

Add tags to a bucket

You can run the following code to add tags to a bucket:

```
package    main

import (
    " fmt "
    " os "

    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ( "< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Initialize s the tags .
    tag1 := oss . Tag {
        Key : " key1 ",
        Value : " value1 ",
    }
```

```

    }
    tag2 := oss.Tag {
        Key : " key2 ",
        Value : " value2 ",
    }
    tagging := oss.Tagging {
        Tags : [] oss.Tag { tag1 , tag2 },
    }
    // Add the tags to the bucket .
    err = client.SetBucketTagging (" yourBucket Name ", tagging )
    if err != nil {
        fmt.Println (" Error :", err )
        os.Exit (- 1 )
    }
}

```

Obtain the tags added to a bucket

You can run the following code to obtain the tags added to a bucket:

```

package main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main () {
    // Creates an OSSClient instance .
    client, err := oss.New ("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println (" Error :", err )
        os.Exit (- 1 )
    }

    // Obtains the tags added to the bucket .
    ret, err := client.GetBucketTagging (" yourBucket Name ")
    if err != nil {
        fmt.Println (" Error :", err )
        os.Exit (- 1 )
    }
    // Prints the number of tags added to the bucket .
    fmt.Println (" Tag length : ", len ( ret.Tags ))
}

```

List buckets with a specified tag

Run the following code to list buckets with a specified tag:

```

package main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

```

```

func main () {
    // Creates an OSSClient .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Searches for the bucket based on TagKey .
    ret , err := client . ListBucket s ( oss . TagKey ("
yourTaggin gKey "))
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Prints the bucket informatio n .
    for _ , bucket := range ret . Buckets {
        fmt . Println (" bucket :", bucket )
    }
    // Searches for the bucket based on TagKey and
    TagValue .
    // TagValue must be used with TagKey . You can leave
    the value of TagValue as null . In this case ,
    TagValue is not used as a filter to search for
    the bucket .
    res , err := client . ListBucket s ( oss . TagKey ("
yourTaggin gKey "), oss . TagValue (" yourTaggin gValue "))
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Prints the bucket informatio n .
    for _ , b := range res . Buckets {
        fmt . Println (" bucket :", b )
    }
}

```

Delete the tags added to a bucket

You can run the following code to delete the tags added to a bucket:

```

package main

import (
    "fmt "
    "os "

    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Deletes the tags added to the bucket .
    err = client . DeleteBuck etTagging (" yourBucket Name ")
}

```

```

    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }
}

```

4.8 Object tagging

4.8.1 Configure object tagging

This topic describes how to configure object tagging.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).



Note:

- You can add tags to an object when uploading it or add tags for an uploaded object. If add tags to an object that already has tags, the original tags are overwritten. For more information about adding tags, see [#unique_52](#).
- To add tags to an object, you must have the permission to call PutObjectTagging.
- The Last-Modified value of an object is not updated when its tags are changed.
- A tag can contain letters, numbers, spaces, and the following symbols: + - = . _ : /

Add tags to an object when uploading it

- Add tags to an object when uploading it by calling PutObject.

Run the following code to add tags to an object when uploading it by calling PutObject:

```

package    main

import (
    "fmt"
    "os"
    "strings"

    "github.com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt.Println (" Error :", err )
        os . Exit (- 1 )
    }
}

```

```
// Obtains the bucket that stores the object .
bucket , err := client . Bucket ("< yourBucket Name >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}

// Configures the tags to be added to the object
and the tagging rule .
tag1 := oss . Tag {
    Key : " key1 ",
    Value : " value1 ",
}
tag2 := oss . Tag {
    Key : " key2 ",
    Value : " value2 ",
}
tagging := oss . Tagging {
    Tags : [] oss . Tag { tag1 , tag2 },
}

// Adds tags to the object when uploading it .
err = bucket . PutObject ("< yourObject Key >", strings .
NewReader ("< yourObject Value >"), oss . SetTagging ( tagging ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
}
```

- Add tags to an object when uploading it in the multipart upload method.

Run the following code to add tags to an object when uploading it in the multipart upload method:

```
package main

import (
    "fmt"
    "os"

    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains the bucket that stores the object .
    bucket , err := client . Bucket ("< yourBucket Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Configures the tags to be added to the
    object and the tagging rule .
```

```

tag1 := oss . Tag {
    Key : " key1 ",
    Value : " value1 ",
}
tag2 := oss . Tag {
    Key : " key2 ",
    Value : " value2 ",
}
tagging := oss . Tagging {
    Tags : [] oss . Tag { tag1 , tag2 },
}

// Obtains the name and path of the local file
to be uploaded .
fileName := "< youFileNam eAndPath >"

// Divides the file into 3 parts and uploads the
parts . The number of parts are calculated based
on the file size .
chunks , err := oss . SplitFileB yPartNum ( fileName , 3 )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

// Opens the file .
fd , err := os . Open ( fileName )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
defer fd . Close ()

// Initialize s the multipart upload task and sets
the tags to be added .
imur , err := bucket . InitiateMu ltipartUpl oad ( "<
yourObject Name >", oss . SetTagging ( tagging ) )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

// Uploads the object part by part .
var parts [] oss . UploadPart
for _ , chunk := range chunks {
    fd . Seek ( chunk . Offset , os . SEEK_SET )
    part , err := bucket . UploadPart ( imur , fd , chunk .
Size , chunk . Number )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
    parts = append ( parts , part )
}
_ , err = bucket . CompleteMu ltipartUpl oad ( imur , parts )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

```

```
}
```

- Add tags to an object when uploading it in the append upload method.

Run the following code to add tags to an object when uploading it in the append upload method:

```
package main

import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtains the bucket that stores the object.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Configures the tags to be added to the
    object and the tagging rule.
    tag1 := oss.Tag {
        Key: "key1",
        Value: "value1",
    }
    tag2 := oss.Tag {
        Key: "key2",
        Value: "value2",
    }
    tagging := oss.Tagging {
        Tags: [] oss.Tag { tag1, tag2 },
    }

    var nextPos int64
    // Uploads the file to the appendable object for
    the first time. Only the tags set when AppendObje
    ct is called for the first time.
    nextPos, err = bucket.AppendObject("< yourObjectName
>", strings.NewReader("< yourObjectValue >"), nextPos, oss
. SetTagging(tagging))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Uploads the file for the second time.
    nextPos, err = bucket.AppendObject("< yourObjectName
>", strings.NewReader("< yourObjectAppendValue >"),
nextPos)
    if err != nil {
```

```

        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
}

```

- Add tags to an object when uploading it in the resumable upload method.

Run the following code to add tags to an object when uploading it in the resumable upload method:

```

package    main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Obtains the bucket that stores the object .
    bucket , err := client . Bucket ( "< yourBucket Name >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Configures the tags to be added to the
    object and the tagging rule .
    tag1 := oss . Tag {
        Key : " key1 ",
        Value : " value1 ",
    }
    tag2 := oss . Tag {
        Key : " key2 ",
        Value : " value2 ",
    }
    tagging := oss . Tagging {
        Tags : [] oss . Tag { tag1 , tag2 },
    }

    // Divides the object into multiple parts of 100
    KB and uses three threads to upload the parts
    simultaneously . Tags are added to the object when
    it is being uploaded .
    err = bucket . UploadFile ( "< yourObjectName >", "<
yourNeedUploadingFile >", 100 * 1024 , oss . Routines ( 3 ) , oss
. SetTagging ( tagging ) )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
}

```



```
}
```

Add tags to an uploaded object or modify the tags added to an object

Run the following code to add tags to an uploaded object or modify the tags added to an object:

```
package    main

import (
    "fmt"
    "os"
    "strings"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtains the bucket that stores the object.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Configures the tags to be added to the object
    and the tagging rule.
    tag1 := oss.Tag {
        Key: "key1",
        Value: "value1",
    }
    tag2 := oss.Tag {
        Key: "key2",
        Value: "value2",
    }
    tagging := oss.Tagging {
        Tags: [] oss.Tag { tag1, tag2 },
    }
    // Adds the tags to the object.
    err = bucket.PutObjectTagging("< yourObjectName >",
tagging)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Add tags to an object when copying it

You can specify one of the following tagging rules when copying an object:

- **Copy (default):** Copy the tags of the source object to the destination object.

- **Replace:** Add the tags specified in the request to the destination object.

Examples of adding tags to objects of different sizes when copying them are described as follows:

- Run the following code to add tags to an object smaller than 1 GB when directly copying it:

```
package main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtains the bucket that stores the object.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Configures the tags to be added to the object and the tagging rule.
    tag1 := oss.Tag {
        Key: "key1",
        Value: "value1",
    }
    tag2 := oss.Tag {
        Key: "key2",
        Value: "value2",
    }
    tagging := oss.Tagging {
        Tags: [] oss.Tag { tag1, tag2 },
    }

    // If you only set the tagging parameter when copying the object, the tags are not added.
    _, err = bucket.CopyObject("< yourSourceObjectName >", "< yourDirectObjectName >", oss.SetTagging(tagging))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // The tags are added only when the TaggingReplace and tagging parameters are both set.
}
```

```

- , err = bucket.CopyObject("< yourSource  ObjectName >",
"< yourDirect  ObjectName >", oss.SetTagging(tagging), oss.
TaggingDirActive(oss.TaggingReplace))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
}
}

```

- Run the following code to add tags to an object larger than 1 GB when copying it in the multipart copy method:

```

package main

import (
    "fmt"
    "os"
    "strings"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtains the bucket that stores the object.
    bucket, err := client.Bucket("< yourBucket Name >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Configures the tags to be added to the
    object and the tagging rule.
    tag1 := oss.Tag{
        Key: "key1",
        Value: "value1",
    }
    tag2 := oss.Tag{
        Key: "key2",
        Value: "value2",
    }
    tagging := oss.Tagging{
        Tags: []oss.Tag{tag1, tag2},
    }

    // Uploads an object for copying.
    content := "this your object value "
    err = bucket.PutObject("< yourObjectName >", strings.
NewReader(content))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

```
// Initialize s an object named < desObjectName > as
the destination object and sets the tags to be
added to the object .
imur, err := bucket . InitiateMultipartUpload ("<
desObjectName >", oss . SetTagging ( tagging ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
// The yourSourceObject indicates the source object .
The source object is uploaded as one part .
part, err := bucket . UploadPart Copy ( imur , "<
yourBucket Name >", "< yourSourceObject >", 0 , int64 ( len (
content )), 1 )
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
parts := [] oss . UploadPart { part }
_, err = bucket . CompleteMultipartUpload ( imur , parts
)
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
}
```

Add tags to a symbolic link object

Run the following code to add tags to a symbolic link object:

```
package main

import (
    "fmt"
    "os"
    "strings"

    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Creates an OSSClient instance .
    client, err := oss . New ("< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains the bucket that stores the object .
    bucket, err := client . Bucket ("< yourBucket Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Configures the tags to be added to the object
and the tagging rule .
    tag1 := oss . Tag {
        Key : " key1 ",
        Value : " value1 ",
    }
```

```

    }
    tag2 := oss . Tag {
        Key : " key2 ",
        Value : " value2 ",
    }
    tagging := oss . Tagging {
        Tags : [] oss . Tag { tag1 , tag2 },
    }

    err = bucket . PutObject ("< yourObject Name >", strings .
NewReader ("< yourObject Value >"))
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    err = bucket . PutSymlink ("< yourObject Name >", "< yourObject
SymlinkName >", oss . SetTagging ( tagging ))
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}
}

```

4.8.2 Obtain the tags added to an object

This topic describes how to obtain the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to obtain the tags added to an object:

```

package main

import (
    " fmt "
    " os "

    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains the bucket that stores the object .
    bucket , err := client . Bucket ("< yourBucket Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains the tags added to the object .
    taggingResult , err := bucket . GetObjectT agging ("<
yourObject Name >")
    if err != nil {

```

```
    fmt.Println("Error :", err)
    os.Exit(-1)
}
fmt.Printf("Object Tagging : %v \n", taggingResult)
}
```

For more information about how to obtain the tags added to an object, see [#unique_54](#).

4.8.3 Delete the tags added to an object

This topic describes how to delete the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to delete the tags added to an object:

```
package main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }

    // Obtains the bucket that stores the object.
    bucket, err := client.Bucket("< yourBucket Name >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }

    // Deletes the tags added to the object.
    err = bucket.DeleteObjectTagging("< yourObjectName >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }
}
```

For more information about how to delete the tags added to an object, see [#unique_56](#).

4.8.4 Object tagging and lifecycle rules

Lifecycle rules can take effect on objects with specified prefixes or tags. You can also specify prefixes and tags as the condition of a lifecycle rule at the same time.



Note:

If you set a specified tag as the condition of a lifecycle rule, the rule applies to an object only when the key and value in the tag of the object both match the specified tag. If you specify a prefix and multiple tags as the condition of a lifecycle rule, the rule applies to an object only when the prefix and tags of the object match the prefix and all tags specified in the rule.

Set tags as the matching condition of a lifecycle rule

Run the following code to set tags as the matching condition of a lifecycle rule:

```
package    main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Sets the lifecycle rule ID to "rule2" and
    // applies it to objects prefixed with "one".
    // The storage class of Objects prefixed with "one"
    // is converted to IA after 3 days and is converted
    // to Archive after 30 days.
    Days := 3,
    StorageClasses := oss.StorageIA,
}
    transitionArch := oss.LifecycleTransition {
        Days := 30,
        StorageClasses := oss.StorageArchive,
    }

    // Configures the tags used as the matching condition
    tag1 := oss.Tag {
        Key : "key1",
        Value : "value1",
    }
    tag2 := oss.Tag {
        Key : "key2",
        Value : "value2",
    }
```

```

    }

    // Sets the bucket lifecycle rule , which includes the
    // tags set as the matching condition .
    rule1 := oss . LifecycleRule {
        ID : " rule1 ",
        Prefix : " one ",
        Status : " Enabled ",
        Transition s : [] oss . LifecycleTransition { transition IA
, transition Arch },
        Tags :[] oss . Tag { tag1 , tag2 },
    }
    rules := [] oss . LifecycleRule { rule }
    err = client . SetBucketLifecycle ("< yourBucket Name >",
rules )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}

```

View the tags set as the matching condition of a lifecycle rule

Run the following code to view the tags set as the matching condition of a lifecycle rule:

```

package main

import (
    "fmt"
    "os"

    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains lifecycle rule for the bucket that stores
    // the object .
    lc , err := client . GetBucketLifecycle ("< yourBucket Name
>")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Prints the lifecycle rule that includes the tags
    // set as the matching condition .
    for i , v := range lc . Rules {
        fmt . Printf (" Bucket %d Lifecycle : %v ", i , v )
        if v . Expiration != nil {
            fmt . Printf (" , Expiration :%v ", * v . Expiration )
        }
        fmt . Printf ("\n ")
    }
}

```



```
}
```

4.9 Server-side encryption

OSS supports server-side encryption for uploaded data. This means that when user data is uploaded, OSS encrypts the data and permanently stores the data. Then, when the data is downloaded by a user, OSS automatically decrypts the data, returns the original data to the user, and declares in the header of the response that the data has been encrypted on the server.

The following server-side encryption methods are available for different application scenarios:

- Server-side encryption that uses CMKs managed by KMS for encryption and decryption (SSE-KMS)

When uploading an object, you can use a specified CMK ID or the default CMK managed by KMS to encrypt and decrypt a large amount of data. This method is cost-effective because you do not need to send user data to the KMS server through networks for encryption and decryption.



Notice:

API calling fees are incurred if you use CMKs to encrypt or decrypt data.

- Server-side encryption fully managed by OSS (SSE-OSS)

When you upload an object, OSS encrypts the object on the server side by using AES256 fully managed by OSS. In this method, OSS uses AES256 to encrypt each object with an individual key. Furthermore, the individual keys are encrypted by a CMK that is updated periodically for higher security. This method applies to encrypt or decrypt bulk data.



Notice:

- Only one server-side encryption method can be used for an object at one time.
- If you configure server-side encryption for a bucket, you can still configure the encryption method for a single object when uploading or copying it. In this case, the encryption method configured for the object takes precedence. For more information, see [PutObject](#).
- For more information about server-side encryption, see [Server-side encryption](#).

Configure server-side encryption for a bucket

You can run the following code to configure the default encryption method for a bucket. After the method is configured, all objects that are uploaded to the bucket without their encryption methods configured are encrypted in the configured default encryption method.

```
package main

import (
    "fmt"
    "os"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Initialize s an encryption rule. In this example,
    the encryption method is set to AES256.
    config := oss.ServerEncryptionRule{SSEDefault: oss.SSEDefaultRule{SSEAlgorithm: "AES256"}}
    err = client.SetBucketEncryption("yourBucketName", config)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Obtain the server-side encryption settings of a bucket

You can run the following code to obtain the server-side encryption settings of a bucket:

```
package main

import (
    "fmt"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

```
// Obtains the server-side encryption rule for the
bucket .
ret , err := client . GetBucketE ncryption (" yourBucket Name
")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
// Prints the obtained encryption rule .
fmt . Println (" Encrypt rule ", ret . SSEDefault . SSEAlgorit
hm )
}
```

Delete the server-side encryption settings of a bucket

You can run the following code to delete the server-side encryption settings of a bucket:

```
package main

import (
    "fmt"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Deletes the encryption rule .
    err = client . DeleteBuck etEncrypti on (" yourBucket Name ")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}
```

4.10 Upload objects

4.10.1 Overview

In OSS, an object is the basic unit for data operations. OSS Go SDK provides the following file upload methods:

- **Simple upload:** supports the upload of a file up to 5 GB in size.
- **Append upload:** supports the upload of a file up to 5 GB in size.

- **Resumable upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files. It supports concurrent upload and you can define the size of each part.
- **Multipart upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files.

**Note:**

For more information about the usage scenarios of each upload method, see “Upload files” in OSS Developer Guide.

During the file upload, you can set [Object Meta](#), and view the upload progress in the [upload progress bar](#).

4.10.2 Simple upload

For the complete code of simple upload, see [GitHub](#).

Upload a string

You can run the following code to upload a string:

```
package    main

import (
    "fmt"
    "os"
    "strings"
    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Upload a string.
    err = bucket.PutObject("< yourObjectName >", strings.NewReader("yourObject Value"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

```
}
```

Upload a byte array

You can run the following code to upload a byte array:

```
package    main

import (
    "fmt"
    "os"
    "bytes"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Upload a byte array.
    err = bucket.PutObject("< yourObjectName >", bytes.NewReader([]byte("yourObjectValueByteArray")))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

Upload a file stream

You can run the following code to upload a file stream:

```
package    main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

```
// Obtain the bucket .
bucket , err := client . Bucket ("< yourBucket  Name >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}

// Read the local file .
fd , err := os . Open ("< yourLocalF ile >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
defer fd . Close ()

// Upload a file stream .
err = bucket . PutObject ("< yourObject  Name >", fd )
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
}
```

Upload a local file

You can run the following code to upload a local file:

```
package main

import (
    "fmt"
    "os"
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtain the bucket .
    bucket , err := client . Bucket ("< yourBucket  Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Upload a local file .
    err = bucket . PutObjectF romFile ("< yourObject  Name >", "<
yourLocalF ile >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}
```

```
}
```

4.10.3 Append upload

You cannot perform `copyObject` for appended objects. When you call the

`AppendObject`, if the target object does not exist, a new appendable object is created. If the target object exists, the uploaded content is appended to the end of the object.

Run the following code to upload a file with append upload:

```
package main

import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    var nextPos int64 = 0
    // If the object is appended for the first time,
    // the append position is 0, and the returned value
    // is the position for the next append. The position
    // for the next append is the current length of the
    // object.
    nextPos, err = bucket.AppendObject("< yourObjectName >",
        strings.NewReader("YourObject AppendValue1"), nextPos)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Start the second append.
    nextPos, err = bucket.AppendObject("< yourObjectName >",
        strings.NewReader("YourObject AppendValue2"), nextPos)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // You can append content to an object for multiple
    // times.
```

```
}
```

You can specify the Object Meta only when you append an object for the first time (that is, the append position is 0).

```
// Specify the Object Meta in the first append
upload .
nextPos , err = bucket . AppendObject ("< yourObject Name >",
strings . NewReader (" YourObject Value "), 0 , oss . Meta ("
MyProp ", " MyPropVal "))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
```

4.10.4 Resumable upload

You can use resumable upload to split a file you want to upload into several parts and upload them simultaneously. After you have uploaded all parts, you can combine these parts into a complete object. Thus, you can complete uploading the entire object.

Current upload progress is recorded into a checkpoint file during the upload. If a part fails to be uploaded during the process, the object is uploaded from the part recorded in the checkpoint file when the upload restarts, that is, the resumable upload. After the upload is complete, the checkpoint is deleted.



Note:

- The SDK records the upload progress information in the checkpoint file. Therefore , you must ensure that the program has the write permission to the checkpoint file . Do not modify the checkpoint file because it includes verification information. If the checkpoint file is corrupted, all parts will be upload again.
- If the local file is changed during file upload, all the parts will be uploaded again.

You can use `Bucket . UploadFile` to achieve resumable upload. You can configure the following parameters.

Parameter	Description
objectKey	Specifies the name of the object uploaded to OSS.
filePath	Specifies the path of the local file.
partSize	Specifies the upload part size, ranging from 100 KB to 5 GB.

Parameter	Description
options	Specifies the options, including: • Routines: Specifies the number of parts that are uploaded at the same time. The default value is 1, which means that parts are not uploaded concurrently. • Checkpoint: Specifies whether the resumable upload is enabled and configures the checkpoint file. Resumable upload is disabled by default. For example, <code>oss . Checkpoint (true , "")</code> indicates that the resumable upload is enabled, and the checkpoint file is the <code>file . cp</code> stored in the directory that the local file needs to be uploaded is stored. The <code>file</code> is the name of the local file. You can also use <code>oss . Checkpoint (true , " your - cp - file . cp ")</code> to specify the checkpoint file. • Other metadata information: See Manage Object Meta.

Run the following code for resumable upload:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client, err := oss.New("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }

    // Obtain the bucket .
    bucket, err := client.Bucket("< yourBucket Name >")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit(-1)
    }
}
```

```

}

// The part size is 100 KB . Parts are uploaded
// through three coroutines concurrently . The resumable
// upload is enabled .
// < yourObjectName > is the objectKey , " LocalFile " is
// the filePath , and 100 * 1024 is the partSize .
err = bucket . UploadFile ( "< yourObjectName >", " LocalFile ",
100 * 1024 , oss . Routines ( 3 ), oss . Checkpoint ( true , ""))
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
}
}

```

4.10.5 Multipart upload

This topic describes how to upload an object in multiple parts

To enable multipart upload, perform the following steps:

1. Initiate a multipart upload event.

You can call `Bucket.InitiateMultipartUpload` to return the globally unique `uploadId` created in OSS.

2. Start the multipart upload task.

You can call `Bucket.UploadPart` to upload data in multiple parts.



Notice:

- Parts with a same `uploadId` are sequenced by their part numbers. If you have uploaded a part and use the same part number to upload another part, the later part will replace the former part.
- The MD5 values of a data part is included in the `ETag` header and is returned to the user.
- OSS SDK automatically configures Content-MD5. OSS calculates the MD5 value of the uploaded data and compares it with the MD5 value calculated by SDK. If the two values are different, the `InvalidDigest` error code is returned.

3. Complete the multipart upload task.

After you upload all parts, call `Bucket.CompleteMultipartUpload` to combine these parts into a complete object.

The following code is used as a complete example that describes the process of multipart upload:

```
package main
```

```

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    bucketName := "< yourBucketName >"
    objectName := "< yourObjectName >"
    localFilename := "< yourLocalFilename >"
    // Obtains a bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    chunks, err := oss.SplitFileByPartNum(localFilename,
3)
    fd, err := os.Open(localFilename)
    defer fd.Close()
    // Step 1: Initiates a multipart upload event.
    imur, err := bucket.InitiateMultipartUpload(
objectName)
    // Step 2: Uploads parts.
    var parts []oss.UploadPart
    for _, chunk := range chunks {
        fd.Seek(chunk.Offset, os.SEEK_SET)
        // Calls UploadPart to upload each part.
        part, err := bucket.UploadPart(imur, fd, chunk.
Size, chunk.Number)
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
        parts = append(parts, part)
    }
    // Step 3: Completes the multipart upload task.
    cmur, err := bucket.CompleteMultipartUpload(imur,
parts)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("cmur:", cmur)
}

```

Cancel a multipart upload event

Run the following code to cancel a multipart upload event:

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

```

```

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Obtains a bucket .
    bucket , err := client . Bucket ("< yourBucket Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Initiates a multipart upload event .
    imur , err := bucket . InitiateMu ltipartUpl oad ("<
yourObject Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Cancels the multipart upload event .
    err = bucket . AbortMulti partUpload ( imur )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}

```

List uploaded parts

Run the following code to list parts uploaded in a multipart upload event:

```

package main
import (
    "fmt "
    "os "
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)
func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
    locaFilena me := "< yourLocalF ilename >"
    uploadID := "< yourUpload ID >"
    // Obtains a bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // Splits the local file into multiple parts ( 3
parts in this example ).
    chunks , err := oss . SplitFileB yPartNum ( locaFilena me ,
3 )
    fd , err := os . Open ( locaFilena me )

```

```

    defer fd . Close ()
    // Initiates a multipart upload event .
    imur , err := bucket . InitiateMu ltipartUpl oad (
objectName )
    uploadID = imur . UploadID
    fmt . Println ( " InitiateMu ltipartUpl oad UploadID : ",
uploadID )
    if err != nil {
        fmt . Println ( " Error : ", err )
        os . Exit ( - 1 )
    }
    // Uploads parts .
    var parts [] oss . UploadPart
    for _ , chunk := range chunks {
        fd . Seek ( chunk . Offset , os . SEEK_SET )
        // Calls UploadPart to upload each part .
        part , err := bucket . UploadPart ( imur , fd , chunk .
Size , chunk . Number )
        if err != nil {
            fmt . Println ( " Error : ", err )
            os . Exit ( - 1 )
        }
        fmt . Println ( " UploadPart Number : ", part . PartNumber ,
", ETag : ", part . ETag )
        parts = append ( parts , part )
    }
    // Lists uploaded parts based on InitiateMu
ltipartUpl oadResult
    lsRes , err := bucket . ListUpload edParts ( imur )
    if err != nil {
        fmt . Println ( " Error : ", err )
        os . Exit ( - 1 )
    }
    // Prints the uploaded parts .
    fmt . Println ( "\ nParts : ", lsRes . UploadedPa rts )
    for _ , upload := range lsRes . UploadedPa rts {
        fmt . Println ( " List PartNumber : ", upload . PartNumber
, " , ETag : " , upload . ETag , " , LastModifi ed : " , upload .
LastModifi ed )
    }
    // Generates InitiateMu ltipartUpl oadResult based on
objectName and UploadID , and then list all uploaded
parts , which applies to scenarios where you know the
objectName and UploadID .
    var imur_with_ uploadid oss . InitiateMu ltipartUpl
oadResult
    imur_with_ uploadid . Key = objectName
    imur_with_ uploadid . UploadID = uploadID
    // Lists uploaded parts .
    lsRes , err = bucket . ListUpload edParts ( imur_with_
uploadid )
    if err != nil {
        fmt . Println ( " Error : ", err )
        os . Exit ( - 1 )
    }
    // Prints the uploaded parts .
    fmt . Println ( "\ nListUploa dedParts by UploadID : ",
uploadID )
    for _ , upload := range lsRes . UploadedPa rts {
        fmt . Println ( " List PartNumber : ", upload . PartNumber
, " , ETag : " , upload . ETag , " , LastModifi ed : " , upload .
LastModifi ed )
    }
    // Completes the multipart upload task .

```

```

    cmur , err := bucket . CompleteMultipartUpload ( imur ,
    parts )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
    fmt . Println ( " cmur :", cmur )
}

```

List all part upload events for a bucket

Call `Bucket . ListMultipartUploads` to list all ongoing multipart upload events (events that are initiated but not completed or events that are canceled). You can configure the following parameters:

Parameter	Description
Delimiter	Specifies a delimiter of a forward slash (/) used to group object names. All objects whose names contain the specified prefix and between which the delimiter occurs for the first time form a group of elements.
MaxUploads	Specifies the maximum number of part upload events. The maximum value (also the default value) you can set is 1,000.
KeyMarker	Lists all multipart upload events in which object names start with the letter that follows the KeyMarker value in alphabetical order. This parameter can be used together with UploadIDMarker to specify the start point of the returned result.
Prefix	Specifies the prefix that must be included in the returned object name. If you use a prefix for query, the returned object name will contain the prefix.

Parameter	Description
UploadIDMarker	UploadIDMarker is used together with the KeyMarker parameter to specify the start point of the returned result. • This parameter is ignored if KeyMarker is not configured. • If KeyMarker is configured, the query result contains: - Multipart upload events in which all object names start with the letter that follows the KeyMarker value in alphabetical order. - Multipart upload events in which the object names equal to the value of KeyMarker but the upload IDs of the objects are larger than the value of UploadIDMarker.

- Use default parameters

```

package main
import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
func main() {
    // Creates an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Obtains a bucket.
    bucketName := "< yourBucketName >"
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    // Lists all multipart upload events.
    keyMarker := ""
    uploadIdMarker := ""
    for {
        // 1000 records are listed at a time by
        default.
        lsRes, err := bucket.ListMultipartUploads(oss.KeyMarker(keyMarker), oss.UploadIDMarker(uploadIdMarker))
        if err != nil {
            fmt.Println("Error:", err)
            os.Exit(-1)
        }
    }
}

```

```

    }
    // Prints multipart upload events .
    for _, upload := range lsRes . Uploads {
        fmt . Println (" Upload : ", upload . Key , ",
UploadID : ", upload . UploadID )
    }
    if lsRes . IsTruncated {
        keyMarker = lsRes . NextKeyMarker
        uploadIdMarker = lsRes . NextUploadIDMarker
    } else {
        break
    }
}
}

```

- Run the following code to list multipart upload events with a specified prefix:

```

lsRes , err := bucket . ListMultipartUploads ( oss .
Prefix ("< yourObject NamePrefix >"))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
fmt . Println (" Uploads :", lsRes . Uploads )

```

- Run the following code to list 100 multipart upload events in maximum at a time:

```

lsRes , err := bucket . ListMultipartUploads ( oss .
MaxUploads ( 100 ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
fmt . Println (" Uploads :", lsRes . Uploads )

```

- Run the following code to list multipart upload events with a specified prefix and a maximum number:

```

lsRes , err := bucket . ListMultipartUploads ( oss .
Prefix ("< yourObject NamePrefix >"), oss . MaxUploads ( 100 ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
fmt . Println (" Uploads :", lsRes . Uploads )

```

4.10.6 Progress bars

You can use progress bars to indicate the upload or download progress.

The following code is used as an example to describe how to view progress information with `Bucket.PutObjectFromFile`:

```

package main

import (
    "fmt"
    "os"
    "github.com / aliyun / aliyun - oss - go - sdk / oss"

```



```

)

// Define a progress bar listener .
type OssProgressListener struct {
}

// Define the progress event handler .
func ( listener * OssProgressListener ) ProgressChanged (
event * oss . ProgressEvent ) {
    switch event . EventType {
        case oss . TransferStartedEvent :
            fmt . Printf ( " Transfer Started , ConsumedBytes : % d ,
TotalBytes % d .\ n " ,
                event . ConsumedBytes , event . TotalBytes )
        case oss . TransferDataEvent :
            fmt . Printf ( "\ rTransfer Data , ConsumedBytes : % d ,
TotalBytes % d , % d %%. " ,
                event . ConsumedBytes , event . TotalBytes , event .
ConsumedBytes * 100 / event . TotalBytes )
        case oss . TransferCompletedEvent :
            fmt . Printf ( "\ nTransfer Completed , ConsumedBytes : % d ,
TotalBytes % d .\ n " ,
                event . ConsumedBytes , event . TotalBytes )
        case oss . TransferFailedEvent :
            fmt . Printf ( "\ nTransfer Failed , ConsumedBytes : % d ,
TotalBytes % d .\ n " ,
                event . ConsumedBytes , event . TotalBytes )
        default :
    }
}

func main () {
// Create an OSSClient instance .
client , err := oss . New ( "< yourEndpoint >" , "< yourAccess
KeyId >" , "< yourAccess KeySecret >" )
if err != nil {
    fmt . Println ( " Error : " , err )
    os . Exit ( - 1 )
}

bucketName := "< yourBucket Name >"
objectName := "< yourObject Name >"
localFile := "< yourLocalFile >"

// Obtain a bucket .
bucket , err := client . Bucket ( bucketName )
if err != nil {
    fmt . Println ( " Error : " , err )
    os . Exit ( - 1 )
}

// Upload with a progress bar displayed .
err = bucket . PutObjectFromFile ( objectName , localFile ,
oss . Progress ( & OssProgressListener {} ) )
if err != nil {
    fmt . Println ( " Error : " , err )
    os . Exit ( - 1 )
}
}

```

```
}
```

4.11 Download objects

4.11.1 Overview

For the complete code of object download, see [GitHub](#).

OSS Go SDK provides the following download methods:

- [Streaming download](#)
- [Download to your local file](#)
- [Resumable download](#)
- [Conditional download](#)
- [Download a compressed object](#)

You can view the download progress shown in the [download progress bar](#).

4.11.2 Streaming download

Download an object to a stream.

Run the following code to download a specified object to a stream:

```
package main

import (
    "fmt"
    "os"
    "io/ioutil"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Download an object to a stream.
    body, err := bucket.GetObject("< yourObjectName >")
    if err != nil {
        fmt.Println("Error:", err)
    }
}
```

```

    os . Exit ( - 1 )
}
// If you do not close the reader after the data
// is read, connection leaks may occur. Consequently, no
// connection is available and the program cannot run
// properly.
defer body . Close ()

data , err := ioutil . ReadAll ( body )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
fmt . Println ( " data :", string ( data ))
}

```

Download an object to the local cache

Run the following code to download a specified object to the local cache:

```

package    main

import (
    " fmt "
    " os "
    " io "
    " bytes "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "< yourAccess
    KeyId >", "< yourAccess KeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Obtain the bucket .
    bucket , err := client . Bucket ( "< yourBucket Name >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Download the object to the local cache .
    body , err := bucket . GetObject ( "< yourObject Name >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
    defer body . Close ()

    buf := new ( bytes . Buffer )
    io . Copy ( buf , body )
    fmt . Println ( " buf :", buf )
}

```

```
}
```

Download an object to a local file stream

Run the following code to download a specified object to a local file stream:

```
package    main

import (
    "fmt"
    "os"
    "io"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Download an object to a local file stream.
    body, err := bucket.GetObject("< yourObjectName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer body.Close()

    fd, err := os.OpenFile("LocalFile", os.O_WRONLY | os.O_CREATE, 0660)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    defer fd.Close()

    io.Copy(fd, body)
}
```

4.11.3 Range download

If you want to download a part of data in an object, you can specify a download range to download only the data that you want.

Run the following code to download data within a specified range:

```
package    main

import (
```

```

" fmt "
" os "
" io / ioutil "
" github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Creates an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains the bucket .
    bucket , err := client . Bucket ("< yourBucket Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtains data between the starting byte value of
    15 and the maximum byte value of 35 inclusive ( that
    is , a total of 21 bytes ).
    // If the specified value range is invalid , for
    example , the specified range includes a negative number
    , or the specified value is larger than the object
    size , the entire object is downloaded .
    body , err := bucket . GetObject ("< yourObject Name >", oss .
Range ( 15 , 35 ))
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    // If you do not close the reader after the data
    is read , connection leaks may occur . Consequent ly ,
    no connection s are available and an exception occurs
    .
    defer body . Close ()

    data , err := ioutil . ReadAll ( body )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    fmt . Println (" data :", string ( data ))
}

```

4.11.4 Download to your local file

This topic describes how to download an object from OSS to a local file.

Run the following code to download a specified object from OSSClient to your local file:

```

package main

import (
    " fmt "
    " os "

```

```

" github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Obtain the bucket .
bucket , err := client . Bucket ("< yourBucket Name >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Download an object to your local file .
err = bucket . GetObjectT oFile ("< yourObject Name >", "
LocalFile ")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}
}
}

```

4.11.5 Resumable download

Large-sized objects may fail to be downloaded because the network is unstable or the program exits abnormally and the entire object needs to be downloaded again. However, the object may still fail to be downloaded after multiple attempts. Therefore, OSS provides resumable download.

To enable resumable download, you need to split the object you want to download into multiple parts and download them separately. After you have downloaded all parts, these parts will be combined into a complete object.

The progress information of the current download is recorded in a checkpoint file . If a part fails to be download, the next download starts from the part based on the progress information recorded in the checkpoint file, achieving the resumable download. After the object is downloaded, the checkpoint file is deleted.



Note:

- OSS SDK records download progress information in a checkpoint file. Therefore , make sure that the program has the write permission to the checkpoint file. Do not modify the checkpoint file because it includes verification information. If the checkpoint file for an object is corrupted, the entire object is downloaded again.

- If the object being downloaded is modified (the etag of the object changes) during the download, or part files are missing or modified, the entire object is download again.

You can use `Bucket . DownloadFile` to enable resumable download.

`DownloadFileRequest` contains the following parameters:

Parameter	Description
<code>objectKey</code>	Specifies the name of the object to be downloaded.
<code>filePath</code>	Specifies the path of the local file that you download the object to.
<code>partSize</code>	Specifies the size of a part. The value can be between 1 byte to 5 GB.
<code>options</code>	Specifies download options, including: • Routines: Specifies the number of parts you need to download simultaneously. The default value is 1. • Checkpoint: Specifies whether the resumable upload is enabled and configures the checkpoint file. The resumable upload is disabled by default. For example, <code>oss . Checkpoint (true , "")</code> indicates that the resumable download is enabled, and the checkpoint file is <code>file . cp</code>, in which <code>file</code> is the name of the local file. The checkpoint file is located in the directory where the local file is located. You can use <code>oss . Checkpoint (true , " your - cp - file . cp ")</code> to specify the checkpoint file. • If you want to set conditions for the download, see Conditional download.

Run the following code for resumable download:

```
package    main

import (
    "fmt"
    "os"
```

```

" github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Obtain the bucket .
bucket , err := client . Bucket ("< yourBucket Name >")
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}

// Start the resumable download . Three parts are
downloaded simultaneo usly .
// < yourObject Name >" is the objectKey , " LocalFile " is
the filePath , and 100 * 1024 is the partSize .
err = bucket . DownloadFi le ("< yourObject Name >", " LocalFile
", 100 * 1024 , oss . Routines ( 3 ), oss . Checkpoint ( true ,
""))
if err != nil {
fmt . Println (" Error :", err )
os . Exit (- 1 )
}
}
}

```

4.11.6 Conditional download

You can configure one or more conditions for downloads. If the specified conditions are met, the object can be downloaded. If the specified conditions are not met, an error code is returned and the object cannot be downloaded. You can configure the following conditions: You can configure the following conditions:

Parameter	Description	Configuration method
IfModifiedSince	If the specified time is earlier than the time the object is modified, the object can be downloaded . Otherwise, an error (304 Not modified) is returned.	oss.IfModifiedSince

Parameter	Description	Configuration method
IfUnmodifiedSince	If the specified time is later than or equal to the time the object is modified, the object can be downloaded . Otherwise, an error (412 Precondition failed) is returned.	oss.IfUnmodifiedSince
IfMatch	If the specified ETag matches that of an object, the object can be downloaded. Otherwise, an error (412 Precondition failed) is returned.	oss.IfMatch
IfNoneMatch	If the specified ETag does not match that of an object, the object can be downloaded. Otherwise, an error (304 Not modified) is returned.	oss.IfNoneMatch

Both If-Modified-Since and If-Unmodified-Since can exist at the same time as object download conditions. Both If-Match and If-None-Match can exist at the same time as object download conditions.

You can use `Bucket.GetObjectDetailedMeta` to Obtain the ETag.

```
package    main

import (
    "fmt"
    "os"
    "time"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }

    // Obtain the bucket .
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }
}
```

```

}

// Configure time conditions .
date := time . Date ( 2015 , time . November , 10 , 23 , 0 ,
0 , 0 , time . UTC )

// The condition is not met , and the object will
not be downloaded .
err = bucket . GetObjectT oFile ("< yourObject Name >", "
LocalFile ", oss . IfModified Since ( date ))
if err == nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}

// The conditions are met , and the object will be
downloaded .
err = bucket . GetObjectT oFile ("< yourObject Name >", "
LocalFile ", oss . IfUnmodifi edSince ( date ))
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
}

```

4.11.7 Download a compressed object

This topic describes how to download a compressed object.

You can download a compressed object. Currently, GZIP is supported for object compression. `Bucket.GetObject` and `Bucket.GetObjectToFile` support compression in downloads.

```

package main

import (
    "fmt"
    "os"
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtain the bucket .
    bucket , err := client . Bucket ("< yourBucket Name >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Download an compressed object .
    err = bucket . GetObjectT oFile ("< yourObject Name >", "
LocalFile . gzip ", oss . AcceptEnco ding (" gzip "))
}

```

```

    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit (- 1 )
    }
}

```

4.11.8 Progress bars

You can use progress bars to indicate the upload or download progress.

The following code is used as an example to describe how to view progress information with `Bucket.GetObjectToFile`:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

// Define a progress bar listener .
type OssProgressListener struct {
}

// Define the progress event handler .
func (listener *OssProgressListener) ProgressChanged (
    event *oss.ProgressEvent) {
    switch event.EventType {
        case oss.TransferStartedEvent :
            fmt.Printf(" Transfer Started , ConsumedBytes : %d ,
TotalBytes %d .\n ",
                event.ConsumedBytes , event.TotalBytes )
        case oss.TransferDataEvent :
            fmt.Printf("\rTransfer Data , ConsumedBytes : %d ,
TotalBytes %d , %d %%.",
                event.ConsumedBytes , event.TotalBytes , event .
ConsumedBytes * 100 / event.TotalBytes )
        case oss.TransferCompletedEvent :
            fmt.Printf("\nTransfer Completed , ConsumedBytes : %d ,
TotalBytes %d .\n ",
                event.ConsumedBytes , event.TotalBytes )
        case oss.TransferFailedEvent :
            fmt.Printf("\nTransfer Failed , ConsumedBytes : %d ,
TotalBytes %d .\n ",
                event.ConsumedBytes , event.TotalBytes )
        default :
    }
}

func main () {
    // Create an OSSClient instance .
    client, err := oss.New("< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println(" Error :", err )
        os.Exit (- 1 )
    }

    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"

```

```

    localFile := "< yourLocalFile >"

    // Obtain the bucket .
    bucket, err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Download with a progress bar displayed .
    err = bucket . GetObjectToFile ( objectName , localFile , oss
    . Progress (& OssProgressListener {}))
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
    fmt . Println ( " Transfer Completed .")
}

```

4.12 Manage objects

4.12.1 Overview

You can manage objects stored in buckets with APIs, including the following operations:

- [Determine whether an object exists](#)
- [Manage ACL for an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Delete objects](#)
- [Copy objects](#)
- [Restore an archive object](#)
- [Manage a symbolic link](#)

4.12.2 Determine whether a specified object exists

This topic describes how to determine whether a specified object exists

Run the following code to determine whether a specified object exists:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {

```

```
// Create an OSSClient instance .
client , err := oss . New ("< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess  KeySecret >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}

// Obtain the bucket .
bucket , err := client . Bucket ("< yourBucket  Name >")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}

// Determine whether the specified object exists .
isExist , err := bucket . IsObjectEx ist ("< yourObject  Name
>")
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
fmt . Println (" Exist :", isExist )
}
```

4.12.3 Manage ACL for an object

For the complete code demo of ACL, see [GitHub](#).

The following table describes the permissions included in the Access Control List (ACL) for an object.

Permission	Description	Value
default	The ACL of an object is the same with that of its bucket.	oss.ACLDefault
Private	Only the object owner and authorized users can read and write the object.	oss.ACLPrivate
Public read	Only the object owner and authorized users can read and write the object. Other users can only read the object. Authorize this permission with caution.	oss.ACLPublicRead
Public read-write	All users can read and write the object. Authorize this permission with caution.	oss.PublicReadWrite

The ACL privileges of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the bucket.
    bucket, err := client.Bucket("< yourBucketName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Configure an ACL for an object.
    err = bucket.SetObjectACL("< yourObjectName >", oss.ACLPublicReadWrite)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Obtain the ACL for an object.
    aclRes, err := bucket.GetObjectACL("< yourObjectName >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Object ACL:", aclRes.ACL)
}
```

4.12.4 Manage Object Meta

For more information about Object Meta, see [Object Meta](#) in OSS Developer Guide.

For the complete code of Object Meta, see [GitHub](#).

Configure Object Meta

You can configure the Object Meta when uploading an object. The following table lists the Object Meta you can configure:

Parameter	Description
CacheControl	Specifies the cache action of a webpage when an object is downloaded.
ContentDisposition	// Specify the name of the object when it is downloaded.
ContentEncoding	Specifies the encoding format when an object is downloaded.
Expires	Specifies the cache expiration time (GMT is used).
ServerSideEncryption	Specifies the server-side encryption algorithm when the OSS creates an object . Valid value: AES256.
ObjectACL	Specifies the ACL for an object created by OSS.
Meta	Specifies the custom Object Meta, with X - Oss - Meta - as the prefix.

Run the following code to configure Object Meta:

```
package main

import (
    "fmt"
    "os"
    "time"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }

    // Obtain the bucket .
    bucket, err := client.Bucket("< yourBucket Name >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }

    // Configure Object Meta : The expiration time is
    2049 - 01 - 10 , 23 : 00 : 00 in the GMT format . The ACL
    is public read . The custom Object Meta is MyProp (
    the value is MyPropVal ).
    expires := time.Date(2049, time.January, 10, 23,
    0, 0, 0, time.UTC)
```

```

options := [] oss . Option {
    oss . Expires ( expires ),
    oss . ObjectACL ( oss . ACLPublicRead ),
    oss . Meta ( " MyProp ", " MyPropVal " ),
}

// Use the data stream to upload .
err = bucket . PutObject ( "< yourObject Name >", strings .
NewReader ( " MyObjectValue " ), options ... )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

// Obtain Object Meta .
props , err := bucket . GetObjectDetailedMeta ( "<
yourObject Name >" )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
fmt . Println ( " Object Meta :", props )
}

```

Modify Object Meta

Run the following code to modify one or more Object Meta at a time:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Modify one Object Meta at a time .
    err = bucket . SetObjectMeta ( objectName , oss . Meta ( "
MyMeta ", " MyMetaValue1 " ) )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
}

```



```

    }

    // Modify multiple Object Metas at one time .
    options := [] oss . Option {
        oss . Meta (" MyMeta ", " MyMetaValue2 "),
        oss . Meta (" MyObjectLocation ", " HangZhou "),
    }
    err = bucket . SetObjectMeta ( objectName , options ...)
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtain Object Meta .
    props , err := bucket . GetObjectDetailedMeta ( objectName
)
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    fmt . Println (" Object Meta :", props )
}

```

Obtain Object Meta

Run the following code to obtain Object Meta:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    bucketName := "< yourBucketName >"
    objectName := "< yourObjectName >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // Obtain Object Meta .
    props , err := bucket . GetObjectDetailedMeta ( objectName
)
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}

```

```
    fmt.Println(" Object Meta :", props )
}
```

4.12.5 List objects

Objects are listed alphabetically. You can use `Bucket . ListObject s` to list objects in a bucket. The following parameters can be used:

Parameter	Description
delimiter	Specifies a delimiter used to group objects. The substring between the specified prefix and the first occurrence of the delimiter of a forward slash (/) is <code>commonPrefixes</code> .
prefix	Specifies the prefix you configure to list required objects.
maxKeys	Specifies the maximum number of objects that can be listed. The default value is 100. The maximum value is 1,000.
marker	Specifies the initial object in the list.

For the complete code of `ListObjects`, see [GitHub](#).

Simple list

You can use the default parameters to obtain the object list of a bucket. 100 objects can be listed by default. Run the following code to list objects in a bucket:

```
package main

import (
    "fmt"
    "os"
    "github.com / aliyun / aliyun - oss - go - sdk / oss "
)

func HandleError ( err error ) {
    fmt.Println(" Error :", err )
    os.Exit(-1)
}

func main () {
    // Create an OSSClient instance .
    client , err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError ( err )
    }
}
```

```

// Obtain the bucket .
bucketName := "< yourBucket Name >"
bucket, err := client . Bucket ( bucketName )
if err != nil {
    HandleError ( err )
}

// List all objects .
marker := ""
for {
    lsRes, err := bucket . ListObject s ( oss . Marker (
marker ))
    if err != nil {
        HandleError ( err )
    }

    // Prints the object list . 100 records are
    returned by default .
    for _, object := range lsRes . Objects {
        fmt . Println ( " Bucket : ", object . Key )
    }

    if lsRes . IsTruncate d {
        marker = lsRes . NextMarker
    } else {
        break
    }
}
}

```

List a specified number of objects

Run the following code to list a specified number of objects:

```

package main

import (
    "fmt"
    "os"
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client, err := oss . New ( "< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    bucketName := "< yourBucket Name >"

    // Obtain the bucket .
    bucket, err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
}

```

```
// Configure the maximum number of the listed
object and list objects .
lsRes , err := bucket . ListObject s ( oss . MaxKeys ( 200 ))
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

// Print the results .
fmt . Println ( " Objects :", lsRes . Objects )
for _ , object := range lsRes . Objects {
    fmt . Println ( " Object :", object . Key )
}
}
```

List objects by a specified prefix

Run the following code to list objects with a specified prefix:

```
package main

import (
    "fmt"
    "os"
    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    bucketName := "< yourBucket Name >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // List objects with a specified prefix . 100
objects are listed by default .
    lsRes , err := bucket . ListObject s ( oss . Prefix ( " my -
object -"))
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // Print the results .
    fmt . Println ( " Objects :", lsRes . Objects )
    for _ , object := range lsRes . Objects {
        fmt . Println ( " Object :", object . Key )
    }
}
```

List objects by marker

The marker parameter indicates the name of an object from which the listing begins. Run the following code to specify an object (specified with marker) after which the listing begins:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"

    // Obtain the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // List objects placed after the object (specified
    with marker). 100 objects are listed by default.
    lsRes, err := bucket.ListObjects(oss.Marker("my-
object-xx"))
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Print the results.
    fmt.Println("Objects:", lsRes.Objects)
    for _, object := range lsRes.Objects {
        fmt.Println("Object:", object.Key)
    }
}
```

List all objects on one or more pages

Run the following code to list all objects on one or more pages. The maximum number of objects listed on each page is specified with maxKeys.

```
package main

import (
```

```

    " fmt "
    " os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    bucketName := "< yourBucket Name >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // List all objects on multiple pages . 200 objects
    are listed in one page .
    marker := oss . Marker ("")
    for {
        lsRes , err := bucket . ListObject s ( oss . MaxKeys (
200 ), marker )
        if err != nil {
            fmt . Println (" Error :", err )
            os . Exit (- 1 )
        }
        marker = oss . Marker ( lsRes . NextMarker )

        fmt . Println (" Objects :", lsRes . Objects )

        if ! lsRes . IsTruncate d {
            break
        }
    }
}

```

List objects by marker on one or more pages

Run the following code to specify an object (specified with marker) after which the listing begins on one or more pages: The maximum number of objects listed on each page is specified with maxKeys.

```

package main

import (
    " fmt "
    " os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .

```

```

    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    bucketName := "< yourBucket Name >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // List objects placed after the object ( specified
with marker ) on multiple pages . 50 objects are listed
in one page .
    marker = oss . Marker (" my - object - xx ")
    for {
        lsRes , err := bucket . ListObject s ( oss . MaxKeys ( 50
), marker )
        if err != nil {
            fmt . Println (" Error :", err )
            os . Exit (- 1 )
        }

        marker = oss . Marker ( lsRes . NextMarker )

        fmt . Println (" Objects :", lsRes . Objects )

        if ! lsRes . IsTruncate d {
            break
        }
    }
}

```

List objects by a specified prefix on one or more pages

Run the following code to list objects by a specified prefix on one or more pages: The maximum number of objects listed on each page is specified with `maxKeys`.

```

package main

import (
    "fmt "
    "os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    bucketName := "< yourBucket Name >"

```

```

// Obtain the bucket .
bucket , err := client . Bucket ( bucketName )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

// List objects with a specified prefix on multiple
pages . 80 objects are listed in one page .
prefix := oss . Prefix ( " my - object -" )
marker := oss . Marker ( "" )
for {
    lsRes , err := bucket . ListObject s ( oss . MaxKeys ( 80
), marker , prefix )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    prefix = oss . Prefix ( lsRes . Prefix )
    marker = oss . Marker ( lsRes . NextMarker )

    fmt . Println ( " Objects :", lsRes . Objects )

    if ! lsRes . IsTruncate d {
        break
    }
}
}

```

4.12.6 Delete objects



Warning:

Delete objects with caution because deleted objects cannot be recovered.

For the complete code of deleting objects, see [GitHub](#).

Delete an object

Run the following code to delete a single object:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess KeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
}

```



```

bucketName := "< yourBucket Name >"
objectName := "< yourObject Name >"

// Obtain the bucket .
bucket, err := client . Bucket ( bucketName )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

// Delete an object .
err = bucket . DeleteObject ( objectName )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
}

```

Delete multiple objects

You can use `Bucket . DeleteObjects` to delete multiple objects and specify the `DeleteObjectsQuiet` parameter to determine whether the result of the delete operation is returned. The results of the deleted objects are returned by default.

Run the following code to delete multiple objects:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client, err := oss . New ( "< yourEndpoint >", "< yourAccessKeySecret >" )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    bucketName := "< yourBucket Name >"

    // Obtain the bucket .
    bucket, err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }

    // The results of deleted objects are returned .
    delRes, err := bucket . DeleteObjects ([] string { " my - object - 1 ", " my - object - 2 " })
    if err != nil {
        fmt . Println ( " Error :", err )
        os . Exit ( - 1 )
    }
}

```

```

    fmt.Println(" Deleted   Objects :", delRes.DeletedObje  cts )

// The deletion results are not returned .
_, err = bucket.DeleteObje  cts ([] string {" my - object - 3
", " my - object - 4 "},
oss.DeleteObje  ctsQuiet ( true ))
if err != nil {
    fmt.Println(" Error :", err )
    os.Exit (- 1 )
}
}
}

```

4.12.7 Copy objects

For the complete code of copying objects, see [GitHub](#).

Copy objects within the same bucket

Run the following code to copy objects within the same bucket:

```

package    main

import (
    " fmt "
    " os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi  nt >", "< yourAccess
KeyId >", "< yourAccess  KeySecret >")
    if err != nil {
        fmt.Println (" Error :", err )
        os.Exit (- 1 )
    }

    bucketName  := "< yourBucket  Name >"
    objectName  := "< yourObject  Name >"
    destObject  Name := "< yourDestOb  jectName >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt.Println (" Error :", err )
        os.Exit (- 1 )
    }

    // Copy an object to another object in the same
    bucket .
    _, err = bucket . CopyObject ( objectName , destObject  Name )
    if err != nil {
        fmt.Println (" Error :", err )
        os.Exit (- 1 )
    }
}

```

```
}
```

Copy objects to another bucket

You can copy an object from the current bucket to another bucket or from other buckets to the current bucket. Run the following code to copy objects to another bucket:

```
package    main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"
    srcBucketName := "< yourSrcBucketName >"
    dstBucketName := "< yourDstBucketName >"
    srcObjectName := "< yourSrcObjectName >"
    dstObjectName := "< yourDstObjectName >"

    // Obtain the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Copy the source object (srcObjectName) from other buckets (srcBucketName) to the current bucket.
    bucket.CopyObjectFrom(srcBucketName, srcObjectName, dstObjectName)
    if err != nil {
        fmt.Println("CopyObjectFrom Error:", err)
        os.Exit(-1)
    }

    // Copy the source object (srcObjectName) from the current bucket to another bucket (dstBucketName).
    bucket.CopyObjectTo(dstBucketName, dstObjectName, srcObjectName)
    if err != nil {
        fmt.Println("CopyObjectTo Error:", err)
        os.Exit(-1)
    }
}
```

```
}
```

- Process object metadata during copy

You can process the object metadata when copying the object by specifying the

`MetadataDirective` parameter:

- `oss.MetaCopy`: The default value, indicating that the metadata of the target object is the same with that of the source object, that is, copying the source object metadata.
- `oss.MetaReplace`: Replace the source object metadata with the new metadata.



Notice:

If you specify the `MetadataDirective` parameter to `oss.MetaReplace`, the metadata of the source object is overwritten and cannot be recovered. Therefore, perform this operation with caution.

The following metadata can be specified if the value of `MetadataDirective` is `oss.MetaReplace`:

Metadata	Description
CacheControl	Specifies the cache action of the web page when the new object is downloaded.
ContentDisposition	Specifies the name of the new object when it is downloaded.
ContentEncoding	Specifies the content encoding format when a new object is downloaded.
Expires	Specify the expiration time (in GMT) of the new object.
ServerSideEncryption	Specifies the server-side encryption algorithm used when the OSS creates an object. Valid value: AES256
ObjectACL	Specifies ACL of an OSS object when it is created.
Meta	Custom parameter prefixed with <code>x-oss-meta-</code> .

Run the following code to process object metadata during copy:

```
package main
```

```

import (
    "fmt"
    "os"
    "time"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
    destObjectName := "< yourDestObjectName >"

    // Obtain the bucket.
    bucket, err := client.Bucket(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Specify the metadata of the target object.
    expires := time.Date(2049, time.January, 10, 23, 0, 0, 0, time.UTC)
    options := []oss.Option{
        oss.MetadataDirective(oss.MetaReplace),
        oss.Expires(expires),
        oss.ObjectACL(oss.ACLPublicRead),
        oss.Meta("MyMeta", "MyMetaValue")}

    // Overwrites the metadata of the source object with the specified metadata.
    _, err = bucket.CopyObject(objectName, destObjectName, options...)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

Conditional copy

You can set conditions for object copy. The object is copied only when the conditions are satisfied. Otherwise, an error is reported and the object does not get copied.

Available conditions include:

Parameter	Description
CopySourceIfMatch	If the source object's ETag value is the same as the ETag provided by the user, the copy operation is executed. Otherwise, an error is returned.

Parameter	Description
CopySourceIfNoneMatch	If the source object' s ETag value is not the same as the ETag provided by the user, the copy operation is actioned. Otherwise, an error is returned.
CopySourceIfModifiedSince	If the specified time in the input parameter is the same as or later than the actual modification time of the source object, the object is copied normally. Otherwise, an error is returned.
CopySourceIfUnmodifiedSince	If the source object has been modified after the time specified by the user, the object is copied. Otherwise, an error is returned.

CopySourceIfMatch and **CopySourceIfNoneMatch** can be used as conditions together . **CopySourceIfModifiedSince** and **CopySourceIfUnmodifiedSince** can be used as conditions together.

Run the following code for conditional copy:

```
package    main

import (
    "fmt"
    "os"
    "time"
    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
    KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    // bucketName := "< yourBucket Name >"
    // objectName := "< yourObject Name >"
    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
    destMatchO bjectName := "< yourMatchD estObjectN ame >"
    destUnmatc hObjectName e := "< yourUnmatc hDestObjec tName >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
}
```

```

    date := time . Date ( 2011 , time . November , 10 , 23 , 0 ,
0 , 0 , time . UTC )

// The conditions are satisfied , copy the object .
_, err = bucket . CopyObject ( objectName , destMatchOb
bjectName , oss . CopySource IfModified Since ( date ))
if err != nil {
    fmt . Println ( " CopyObject   CopySource   IfModified   Since   Error
:" , err )
}

// The conditions are not met , the object is not
copied .
_, err = bucket . CopyObject ( objectName , destUnMatc
hObjectName , oss . CopySource IfUnmodifi edSince ( date ))
if err != nil {
    fmt . Println ( " CopyObject   CopySource   IfUnmodifi edSince
Error : " , err )
}
}
}

```

4.12.8 Restore an archive object

You must restore an archive object before you read it. Do not call `restoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

- An archive object is in the frozen state.
- After you submit it for restoration, the server restores the object. The object is in the restoring state.
- One minute later, the object enters the restored state and can be read. The restored state lasts for one day by default, and can be prolonged to a maximum of seven days. Once this period ends, the object returns to the frozen state.

For detailed information about archive storage class, see [Introduction to storage classes](#). For detailed information about related status code, see [RestoreObject](#). For the complete code of restoring an archive object, see [GitHub](#).

Run the following code to restore an object:

```

package    main

import (
    "fmt"
    "os"
    "time"
    "strings"
    "github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func    HandleError ( err    error ) {
    fmt . Println ( " Error : " , err )
}

```

```

    os . Exit ( - 1 )
}

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess KeySecret >" )
    if err != nil {
        HandleError ( err )
    }

    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
    localFilename := "< yourLocalFilename >"

    // Create a bucket .
    err = client . CreateBucket ( bucketName , oss . StorageClass
( oss . StorageArchive ))
    if err != nil {
        HandleError ( err )
    }

    // Obtain the bucket .
    archiveBucket , err := client . Bucket ( bucketName )
    if err != nil {
        HandleError ( err )
    }

    // Upload the archived object .
    var val = " More than just cloud ."
    err = archiveBucket . PutObject ( objectName , strings .
.NewReader ( val ))
    if err != nil {
        HandleError ( err )
    }

    // Check whether the object is an archive object .
    meta , err := archiveBucket . GetObjectDetailedMeta (
objectName )
    if err != nil {
        HandleError ( err )
    }

    fmt . Println ( " X - Oss - Storage - Class : ", meta . Get ( " X -
Oss - Storage - Class " ))
    if meta . Get ( " X - Oss - Storage - Class " ) == string ( oss .
StorageArchive ) {
        // Restore the archive object for read .
        err = archiveBucket . RestoreObject ( objectName )
        if err != nil {
            HandleError ( err )
        }

        // Wait until the object is restored .
        meta , err = archiveBucket . GetObjectDetailedMeta (
objectName )
        for meta . Get ( " X - Oss - Restore " ) == " ongoing - request ="
true {
            fmt . Println ( " x - oss - restore : " + meta . Get ( " X - Oss -
Restore " ))
            time . Sleep ( 1000 * time . Second )
            meta , err = archiveBucket . GetObjectDetailedMeta (
objectName )
        }
    }
}

```



```

}

// Download the restored object .
err = archiveBucket.GetObjectToFile ( objectName ,
localFilename )
if err != nil {
    HandleError ( err )
}

// Restore the object again .
err = archiveBucket.RestoreObject ( objectName )

fmt.Println ( " ArchiveSample completed " )
}

```

4.12.9 Manage a symbolic link

A symbolic link is a special object that maps to an object similar to a shortcut used in Windows. You can configure user-defined Object Meta for symbolic links.

Run the following code to create a symbolic link:

```

package main

import (
    "fmt"
    "os"
    "strings"
    "github.com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss.New ("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println ( " Error :", err )
        os.Exit (- 1 )
    }

    bucketName := "< yourBucketName >"
    // The symbolic link .
    objectName := "< yourSymLinkName >"
    // The target object that the symbolic link directs
    to
    targetObjectName := "< yourSymLinkTargetName >"

    // Obtain the bucket .
    bucket , err := client.Bucket ( bucketName )
    if err != nil {
        fmt.Println ( " Error :", err )
        os.Exit (- 1 )
    }

    // Create a symbolic link .
    err = bucket.PutSymlink ( objectName , targetObjectName )
    if err != nil {
        fmt.Println ( " Error :", err )
        os.Exit (- 1 )
    }
}

```

```
// Upload a file to the object that the symbolic
link directs to .
err = bucket . PutObject ( targetObjectName , strings .
.NewReader ( " target " ))
if err != nil {
    fmt . Println ( " Error : ", err )
    os . Exit ( - 1 )
}

err = bucket . PutSymlink ( objectName , targetObjectName )
if err != nil {
    fmt . Println ( " Error : ", err )
    os . Exit ( - 1 )
}

// Obtain the content of the object that the
symbolic link directs to .
meta , err := bucket . GetSymlink ( objectName )
if err != nil {
    fmt . Println ( " Error : ", err )
    os . Exit ( - 1 )
}
fmt . Println ( meta . Get ( oss . HTTPHeader OssSymlink Target
))
}
```

For more information about creating symbolic links, see [PutSymlink](#).

Obtain object content for a symbolic link

To obtain a symbolic link, you must have the read permission on it. Run the following code to obtain the object content for a symbolic link:

```
package main

import (
    "fmt"
    "os"
    "github.com / aliyun / aliyun - oss - go - sdk / oss"
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >" )
    if err != nil {
        fmt . Println ( " Error : ", err )
        os . Exit ( - 1 )
    }

    bucketName := "< yourBucketName >"
    // Symbolic Link name .
    objectName := "< yourSymlinkObjectName >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        fmt . Println ( " Error : ", err )
        os . Exit ( - 1 )
    }
}
```

```
// Obtain the content of the object that the
symbolic link directs to .
meta , err := bucket . GetSymlink ( objectName )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
fmt . Println ( meta . Get ( oss . HTTPHeader OssSymlink Target
))
}
```

For more information about obtaining symbolic links, see [GetSymlink](#).

4.13 Authorized access

Sign a URL to authorize temporary access

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time for a URL to restrict the period of access from visitors.

For the complete code of signing a URL to authorize temporary access, see [GitHub](#).

- Use a signed URL to upload a file

Run the following code to upload a file with a signed URL:

```
package    main

import (
    "fmt"
    "os"
    "strings"

    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError (err)
    }

    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
    localFilename := "< yourLocalFilename >"

    // Obtain the bucket .
    bucket, err := client . Bucket ( bucketName )
    if err != nil {
```

```

    HandleError r ( err )
}

// Directly upload the file after the the URL is
signed .
signedURL , err := bucket . SignURL ( objectName , oss .
HTTPPut , 60 )
if err != nil {
    HandleError r ( err )
}

var val = " Choose Alibaba Cloud "
err = bucket . PutObjectW ithURL ( signedURL , strings .
.NewReader ( val ))
if err != nil {
    HandleError r ( err )
}

// Directly upload the file after the the URL
is signed , in which you can configure optional
parameters .
options := [] oss . Option {
    oss . Meta ( " myprop " , " mypropval " ) ,
    oss . ContentType ( " image / tiff " ) ,
}

signedURL , err = bucket . SignURL ( objectName , oss .
HTTPPut , 60 , options ... )
if err != nil {
    HandleError r ( err )
}

err = bucket . PutObjectF romFileWit hURL ( signedURL ,
localFileN ame , options ... )
if err != nil {
    HandleError r ( err )
}
}

```

**Note:**

For more information about the optional parameters, see [Manage Object Meta](#) in [Manage objects](#).

- Use a signed URL to download an object

Run the following code to use a signed URL to download an object:

```

package main

import (
    "fmt"
    "os"
    "io / ioutil "

    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func HandleError r ( err error ) {
    fmt . Println ( " Error : " , err )
    os . Exit ( - 1 )
}

```

```

}

func main () {
    client , err := oss . New ("< yourEndpoi nt >", "< yourAccess
    KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        HandleError ( err )
    }

    bucketName := "< yourBucket Name >"
    objectName := "< yourObject Name >"
    localDownl oadedFilena me := "< yourDownlo adedFilena me >"

    // Obtain the bucket .
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        HandleError ( err )
    }

    // Directly download the object to the stream after
    the URL is signed .
    signedURL , err := bucket . SignURL ( objectName , oss .
    HTTPGet , 60 )
    if err != nil {
        HandleError ( err )
    }

    body , err := bucket . GetObjectW ithURL ( signedURL )
    if err != nil {
        HandleError ( err )
    }
    // Read the object content .
    data , err := ioutil . ReadAll ( body )
    body . Close ()
    data = data // use data

    // Directly download the object to a local file
    after the URL is signed .
    err = bucket . GetObjectT oFileWithU RL ( signedURL ,
    localDownl oadedFilena me )
    if err != nil {
        HandleError ( err )
    }
}

```

Use STS for temporary access authorization

OSS supports Alibaba Cloud Security Token Service (STS) for temporary access authorization. STS is a web service that provides a temporary access token to a cloud computing user. Through the STS, you can assign a third-party application or a RAM user (you can manage the user ID) an access credential with a custom validity period and permissions. For more information about STS, see [STS introduction](#).

STS advantages:

- Your long-term key (AccessKey) is not exposed to a third-party application. You only need to generate an access token and send the access token to the third-party application. You can customize access permissions and the validity of this token.
- You do not need to keep track of permission revocation issues. The access token automatically becomes invalid when it expires.

For more information about the process of access to OSS with STS, see [RAM and STS scenario practices](#) in OSS Developer Guide.

Run the following code to create a signature request with STS:

```
import "github.com/aliyun/aliyun-oss-go-sdk/oss"

// After a user obtains a temporary STS credential,
// the OSSClient is generated with the security token
// and temporary access key (AccessKeyId and AccessKeySecret).
// Create an OSSClient instance.
client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >", oss.SecurityToken("< yourSecurityToken >"))
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

// Perform operations on OSS.
```

4.14 Manage lifecycle rules

You can use OSS to configure lifecycle rules to reduce storage costs. The lifecycle management rules can be used for automatic deletion of expired objects and fragments, or storage class conversion to Infrequent Access for objects that will expire soon. Each rule includes:

- Rule ID: It identifies a rule. Ensure that each rule ID in a bucket is unique.
- Policy: You can configure a policy by using the following configuration methods. Only one method can be configured for a bucket.
 - Configure by Prefix: You can create multiple rules with this method. Ensure that each prefix is unique.
 - Configure for Entire Bucket: You can configure only one rule with this method.

- **Expired:** You can configure the following configuration methods:
 - **Set by Number of Days:** It specifies the number of days from the last modification time of objects.
 - **Set by Date:** It specifies the date prior to which objects are created. If objects are created prior to the date, these objects are deleted.
- **Status:** This lifecycle rule takes effect or not.

The parts uploaded by `uploadPart` method also support lifecycle rules. The last modification time of an object is the time the multipart upload event is initiated.

For more information about lifecycle management, see [Manage object lifecycle](#). For the complete code of lifecycle management, see [GitHub](#).

Configure lifecycle rules

Run the following code to configure lifecycle rules:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    // Lifecycle rule 1 (id: "rule1", enable: true, prefix: "foo/", expiry: Days 3), indicates that an object with the prefix foo expires three days after the last modified date.
    rule1 := oss.BuildLifecycleRuleByDays("rule1", "foo/", true, 3)
    rules := []oss.LifecycleRule{rule1}

    // Configure lifecycle rules.
    bucketName := "< yourBucketName >"
    err = client.SetBucketLifecycle(bucketName, rules)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

```
}
```

View lifecycle rules

Run the following code to view lifecycle rules:

```
package    main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"

    // View lifecycle rules.
    lcRes, err := client.GetBucketLifecycle(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Lifecycle Rules:", lcRes.Rules)
}
```

Clear lifecycle rules

Run the following code to clear lifecycle rules:

```
package    main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"

    // Clear lifecycle rules.
    err = client.DeleteBucketLifecycle(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
    }
}
```



```
    os . Exit (- 1 )
}
```

4.15 Set logging

You can enable access logs to record bucket access to log files, which are stored in a specified bucket.

The log file format is as follows:

```
< TargetPref ix >< SourceBucket et >- YYYY - mm - DD - HH - MM - SS -
UniqueStri ng
```

For more information about access log files, see [Set access logging](#). For the complete code of set logging, see [GitHub](#).

Enable access logging

Run the following code to enable bucket access logging:

```
package    main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"
    targetBucketName := "< yourTargetBucketName >"
    targetPrefix := "< yourTargetPrefix >"

    // Enable access logging for a bucket. targetBucketName
    // is the bucket that logs are stored, targetPrefix is the
    // prefix of the stored log files, that is, the path of the
    // log files.
    err = client.SetBucketLogging(bucketName, targetBucketName,
    targetPrefix, true)
    if err != nil {
        fmt.Println("Error :", err)
        os.Exit(-1)
    }
}
```

View access logging configurations

Run the following code to view the access logging configurations for a bucket:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"

    // View the access logging configurations for a
    bucket.
    logRes, err := client.GetBucketLogging(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Target Bucket:", logRes.LoggingEnabled
.TargetBucket)
    fmt.Println("Target Prefix:", logRes.LoggingEnabled
.TargetPrefix)
}
```

Disable access logging

Run the following code to disable access logging for a bucket:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    // client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"
```

```
// Disable access logging .
err = client . DeleteBucketLogging ( bucketName )
if err != nil {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}
}
```

4.16 Static website hosting

You can set your bucket configuration to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

For more information about static website hosting, see [Static website hosting](#).

Configure static website hosting

Run the following code to configure static website hosting:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance .
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println(" Error :", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"

    // Configure static web hosting : The index page is index.html and the error page is error.html .
    err = client.SetBucketWebsite(bucketName, "index.html", "error.html")
    if err != nil {
        fmt.Println(" Error :", err)
        os.Exit(-1)
    }
}
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
package main
```

```

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"

    // View static web hosting configurations.
    wsRes, err := client.GetBucketWebsite(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("indexWebsite:", wsRes.IndexDocument.Suffix)
    fmt.Println("errorWebsite:", wsRes.ErrorDocument.Key)
}

```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"

    // Delete static website hosting configurations.
    err = client.DeleteBucketWebsite(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}

```

```
}
```

4.17 Anti-leech

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leaching, see [Anti-leeching settings](#). For the complete code of configuring anti-leech, see [GitHub](#).

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"

    // Add the referer field. Empty referer is not allowed. The referer field allows question marks (?) and asterisks (*) for wildcard use.
    referers := []string{"http://www.aliyun.com",
                        "http://www.???.aliyuncs.com",
                        "http://www.*.com"}

    err = client.SetBucketReferer(bucketName, referers, false)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
}
```

```
}
```

Obtain a referer whitelist

Run the following code to obtain a referer whitelist:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"

    // Obtain the referer whitelist.
    refRes, err := client.GetBucketReferer(bucketName)
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }
    fmt.Println("Referers:", refRes.RefererList)
    fmt.Println("AllowEmpty Referer:", refRes.AllowEmptyReferer)
}
```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucket Name >"

    // Clear the referer whitelist. // You cannot clear
    // a referer whitelist directly. To clear a referer
```

```

whitelist, you need to create the rule that allows
an empty referer field and replace the original rule
with the new rule.
err = client.SetBucketReferer(bucketName, []string{},
true)
if err != nil {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
}
}

```

4.18 CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) and [PutBucketcors](#) in OSS Developer Guide.

For the complete code of CORS, see [GitHub](#).

Configure CORS rules

Run the following code to configure CORS rules for the specified bucket:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "< yourAccessKeyID >", "< yourAccessKeySecret >")
    if err != nil {
        fmt.Println("Error:", err)
        os.Exit(-1)
    }

    bucketName := "< yourBucketName >"

    rule1 := oss.CORSRule{
        AllowedOrigins: []string{"*"},
        AllowedMethods: []string{"PUT", "GET"},
        AllowedHeaders: []string{},
        ExposeHeaders: []string{},
        MaxAgeSeconds: 200,
    }

    rule2 := oss.CORSRule{
        AllowedOrigins: []string{"http://www.a.com", "http://www.b.com"},
        AllowedMethods: []string{"POST"},
    }

```

```

    AllowedHeader : [] string {" Authorization "},
    ExposeHeader : [] string {" x - oss - test ", " x - oss - test1
"},
    MaxAgeSeconds : 100 ,
}

// Configure CORS rules .
err = client . SetBucketCORS ( bucketName , [] oss . CORSRule {
rule1 , rule2 })
if err != nil {
    fmt . Println (" Error :", err )
    os . Exit (- 1 )
}
}
}

```

Obtain CORS rules

Run the following code to obtain CORS rules:

```

package main

import (
    " fmt "
    " os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoint >", "< yourAccess
KeyId >", "< yourAccess KeySecret >")
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }

    bucketName := "< yourBucket Name >"

    // Obtain the CORS rules .
    corsRes , err := client . GetBucketCORS ( bucketName )
    if err != nil {
        fmt . Println (" Error :", err )
        os . Exit (- 1 )
    }
    fmt . Println (" Bucket CORS :", corsRes . CORSRules )
}

```

Delete CORS rules

Run the following code to delete all CORS rules for the specified bucket:

```

package main

import (
    " fmt "
    " os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func main () {
    // Create an OSSClient instance .

```



```
client, err := oss.New("< yourEndpoint >", "< yourAccess  
KeyId >", "< yourAccess KeySecret >")  
if err != nil {  
    fmt.Println("Error:", err)  
    os.Exit(-1)  
}  
  
bucketName := "< yourBucket Name >"  
  
// Delete CORS rules.  
err = client.DeleteBucketCORS(bucketName)  
if err != nil {  
    fmt.Println("Error:", err)  
    os.Exit(-1)  
}  
}
```

4.19 Image processing

Image Processing (IMG) is a massive, secure, cost-effective and highly reliable image processing service. After source images are uploaded to OSS, you can process images on any Internet device at any anytime, from anywhere through simple RESTful APIs.

For more information about IMG, see [Image Processing](#).

Basic features

OSS offers the following IMG features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Image processing](#)
- [Image processing access rules](#): calls multiple IMG functions.

Usage

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- Anonymous access

You can use the following format in level-3 domains to access a processed image:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Specifies the name of a bucket.
endpoint	Specifies endpoint used to access a region.
object	Specifies the name of an image object.
image	Specifies the reserved identifier for IMG .
action	Specifies the operations on an image, such as scaling, cropping, and rotating.

Parameter	Description
param	Specifies the parameter that indicates the operation on an image.

- Basic operations

For example, scale an image to a width of 100 px. Adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customized image styles

You can use the following format in level-3 domains to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- **style**: specifies a reserved identifier of a customized image style.
- **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image sequentially.

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- HTTPS access

IMG supports access through HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/
example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

Use the following code to generate a signed URL for IMG:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError (err)
    }

    // Obtain the bucket.
    bucketName := "< yourBucket Name >"
    bucket, err := client.Bucket (bucketName)
    if err != nil {
        HandleError (err)
    }

    ossImageName := "< yourImageName >"
    signedURL, err := bucket.SignURL (ossImageName, oss
.HTTPGet, 600, oss.Process ("image/format,png"))
    if err != nil {
        HandleError (err)
    } else {
        fmt.Println (signedURL)
    }
}
```

```
}
```

- Access with SDK

You can use SDK to access and process an image object.

SDK allows you to customize image styles, HTTPS access, and cascade operations.

- Basic operations

Run the following code to perform basic operations on an image:

```
package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError (err)
    }

    // Obtain the bucket.
    bucketName := "< yourBucket Name >"
    bucket, err := client.Bucket (bucketName)
    if err != nil {
        HandleError (err)
    }

    // Scale the image.
    sourceImageName := "example.png"
    targetImageName := "example-resize.png"
    style := "image/resize,m_fixed,w_100,h_100"
    err = bucket.GetObjectToFile (sourceImageName,
targetImageName, oss.Process (style))
    if err != nil {
        HandleError (err)
    }

    // Crop the image.
    targetImageName = "example-crop.png"
    style = "image/crop,w_100,h_100,x_100,y_100,r_1"
    err = bucket.GetObjectToFile (sourceImageName,
targetImageName, oss.Process (style))
    if err != nil {
        HandleError (err)
    }

    // Rotate the image.
    targetImageName = "example-rotate.png"
```

```

        style = " image / rotate , 90 "
        err = bucket . GetObjectT oFile ( sourceImag eName ,
targetImag eName , oss . Process ( style ))
        if err != nil {
            HandleErro r ( err )
        }

        // Sharpen the image .
        targetImag eName = " example - sharpen . png "
        style = " image / sharpen , 100 "
        err = bucket . GetObjectT oFile ( sourceImag eName ,
targetImag eName , oss . Process ( style ))
        if err != nil {
            HandleErro r ( err )
        }

        // Add watermarks .
        targetImag eName = " example - watermark . png "
        style = " image / watermark , text_SGVsb G8g5Zu -
54mH5pyN5Y qhIQ "
        err = bucket . GetObjectT oFile ( sourceImag eName ,
targetImag eName , oss . Process ( style ))
        if err != nil {
            HandleErro r ( err )
        }

        // Convert the image format .
        targetImag eName = " example - formatconv ert . jpg "
        style = " image / format , jpg "
        err = bucket . GetObjectT oFile ( sourceImag eName ,
targetImag eName , oss . Process ( style ))
        if err != nil {
            HandleErro r ( err )
        }
    }
}

```

- Customize an image style

Run the following code to customize an image style:

```

package main

import (
    " fmt "
    " os "
    " github . com / aliyun / aliyun - oss - go - sdk / oss "
)

func HandleErro r ( err error ) {
    fmt . Println ( " Error :", err )
    os . Exit ( - 1 )
}

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ( "< yourEndpoi nt >", "<
yourAccess KeyId >", "< yourAccess KeySecret >" )
    if err != nil {
        HandleErro r ( err )
    }

    // Obtain the bucket .
    bucketName := "< yourBucket Name >"
}

```

```

        bucket , err := client . Bucket ( bucketName )
        if err != nil {
            HandleError r ( err )
        }

        // Customize the image style .
        sourceImageName := " example . png "
        targetImageName := " example - custom . png "
        style := " style /< yourCustom StyleName >"
        err = bucket . GetObjectToFile ( sourceImageName ,
            targetImageName , oss . Process ( style ))
        if err != nil {
            HandleError r ( err )
        }
    }
}

```

- Cascade operations

Run the following code to perform cascade operations on an image:

```

package main

import (
    "fmt"
    "os"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError r ( err error ) {
    fmt.Println (" Error :", err )
    os.Exit (- 1 )
}

func main () {
    // Create an OSSClient instance .
    client , err := oss . New ("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError r ( err )
    }

    // Obtain the bucket .
    bucketName := "< yourBucket Name >"
    bucket , err := client . Bucket ( bucketName )
    if err != nil {
        HandleError r ( err )
    }

    // Customize an image style .
    sourceImageName := " example . png "
    targetImageName := " example - resize . png "
    style := " image / resize , m_fixed , w_100 , h_100 / rotate
, 90 "
    err = bucket . GetObjectToFile ( sourceImageName ,
        targetImageName , oss . Process ( style ))
    if err != nil {
        HandleError r ( err )
    }
}

```

```
}
```

Image processing persistence

Run the following code for image processing persistence:

```
package main

import (
    "fmt"
    "os"
    "encoding/base64"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)

func HandleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}

func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError (err)
    }

    // Obtain the bucket.
    bucketName := "< yourBucketName >"
    bucket, err := client.Bucket (bucketName)
    if err != nil {
        HandleError (err)
    }

    // Image processing persistence: Scale the image.
    The target image is stored in the same bucket by
    default.
    sourceImageName := "example.png"
    targetImageName := "example-resize.png"
    style := "image/resize,m_fixed,w_100,h_100"
    process := fmt.Sprintf("%s | sys/saveas,o_%v",
style, base64.URLEncoding.EncodeToString ([]byte (
targetImageName)))
    result, err := bucket.ProcessObject (sourceImageName
, process)
    if err != nil {
        HandleError (err)
    } else {
        fmt.Println (result)
    }

    // Image processing persistence: Scale the image.
    The target image is stored in the different bucket.
    targetBucketName := "< yourTargetBucketName >"
    targetImageName = "example-crop-different-bucket.png"
    "
    style = "image/resize,m_fixed,w_100,h_100"
    process = fmt.Sprintf ("%s | sys/saveas,o_%v,b_%
v", style, base64.URLEncoding.EncodeToString ([]byte (
```



```

targetImageName)), base64.URLEncoding.EncodeToString([]
byte(targetBucketName)))
    result, err = bucket.ProcessObject(sourceImageName,
process)
    if err != nil {
        HandleError(err)
    } else {
        fmt.Println(result)
    }

    // Image processing persistence: Crop the image.
    targetImageName = "example - crop . png"
    style = "image / crop , w_100 , h_100 , x_100 , y_100 , r_1 "
    process = fmt.Sprintf("%s | sys / saveas , o_ %v ", style
, base64.URLEncoding.EncodeToString([]byte(targetImage
eName)))
    result, err = bucket.ProcessObject(sourceImageName,
process)
    if err != nil {
        HandleError(err)
    } else {
        fmt.Println(result)
    }

    // Image processing persistence: Rotate the image.
    targetImageName = "example - rotate . png"
    style = "image / rotate , 90 "
    process = fmt.Sprintf("%s | sys / saveas , o_ %v ", style
, base64.URLEncoding.EncodeToString([]byte(targetImage
eName)))
    result, err = bucket.ProcessObject(sourceImageName,
process)
    if err != nil {
        HandleError(err)
    } else {
        fmt.Println(result)
    }

    // Image processing persistence: Sharpen the image.
    targetImageName = "example - sharpen . png"
    style = "image / sharpen , 100 "
    process = fmt.Sprintf("%s | sys / saveas , o_ %v ", style
, base64.URLEncoding.EncodeToString([]byte(targetImage
eName)))
    result, err = bucket.ProcessObject(sourceImageName,
process)
    if err != nil {
        HandleError(err)
    } else {
        fmt.Println(result)
    }

    // Image processing persistence: Add watermarks.
    targetImageName = "example - watermark . png"
    style = "image / watermark , text_SGVsb_G8g5Zu - 54mH5pyN5Y
qhIQ "
    process = fmt.Sprintf("%s | sys / saveas , o_ %v ", style
, base64.URLEncoding.EncodeToString([]byte(targetImage
eName)))
    result, err = bucket.ProcessObject(sourceImageName,
process)
    if err != nil {
        HandleError(err)
    } else {

```

```

        fmt.Println ( result )
    }

    // Image processing persistence : Covert the image
    format .
    targetImageName = " example - formatconvert . jpg "
    style = " image / format , jpg "
    process = fmt.Sprintf ("% s | sys / saveas , o_ % v ", style
, base64.URLEncoding.EncodeToString ([] byte ( targetImageName )))
    result , err = bucket.ProcessObject ( sourceImageName ,
process )
    if err != nil {
        HandleError ( err )
    } else {
        fmt.Println ( result )
    }
}

```

4.20 Error handling

Client errors

Call errors in Go SDK are returned through a unified error interface. The error interface is defined as follows:

```

type error interface {
    Error () string
}

```

Other errors are also returned through this error interface. For example, HTTP request errors are returned as follows:

```

// net . Error
type Error interface {
    error
    Timeout () bool // Is the error a timeout
    Temporary () bool // Is the error temporary
}

```

Server errors

If the request errors occur when OSS Go SDK is used, corresponding errors are returned. In case of HTTP request errors and I/O errors, errors preset in Go are returned. In case of request processing errors of the OSS Server, the following error is returned through the error interface:

```

type ServiceError struct {
    Code string // Error code returned by OSS
    to a user
    Message string // Error details provided by
OSS
    RequestId string // UUID that uniquely identifies
the request
}

```

```

    HostId      string    // OSS cluster that identifies
the    access
    StatusCode  int       // HTTP status code
}

```

If the HTTP status code returned by OSS is not as expected, the following error is returned, also through the error interface:

```

type Unexpected StatusCode Error struct {
    allowed [] int // Expected HTTP status code
returned by OSS
    got int // Actual HTTP status code returned
by OSS
}

```

Example of error handling

The following code is used to demonstrate error handling:

```

package main
import (
    "fmt"
    "os"
    "strings"
    "github.com/aliyun/aliyun-oss-go-sdk/oss"
)
// Error handling function
func HandleError (err error) {
    fmt.Println("Error:", err)
    os.Exit(-1)
}
func main() {
    // Create an OSSClient instance.
    client, err := oss.New("< yourEndpoint >", "<
yourAccessKeyId >", "< yourAccessKeySecret >")
    if err != nil {
        HandleError (err)
    }
    bucketName := "< yourBucketName >"
    objectName := "< yourObjectName >"
    // Obtain the bucket.
    bucket, err := client.Bucket (bucketName)
    if err != nil {
        HandleError (err)
    }
    // Simple upload: string upload
    err = bucket.PutObject (objectName, strings.NewReader ("
yourObjectValueContentString"))
    if err != nil {
        HandleError (err)
    }
}

```

OSS error codes

The following table lists the OSS error codes.

Error code	Description	HTTP status code
AccessDenied	The access is denied.	403
BucketAlreadyExists	The bucket already exists.	409
BucketNotEmpty	The bucket is not empty.	409
EntityTooLarge	The entity is too large.	400
EntityTooSmall	The entity is too small.	400
FileGroupTooLarge	The object group is too large.	400
InvalidLinkName	The name of the object link is the same as the linked object.	400
LinkPartNotExist	The object to which the object link is linked does not exist.	400
ObjectLinkTooLarge	The object link has too many objects.	400
FieldItemTooLong	The table field in the post request is too large.	400
FilePartInterity	The object part has been changed.	400
FilePartNotExist	The object part does not exist.	400
FilePartStale	The object part has expired.	400
IncorrectNumberOfFilesInPOSTRequest	The number of objects in the post request is incorrect.	400
InvalidArgument	The parameter format is invalid.	400
InvalidAccessKeyId	The AccessKeyId does not exist.	403
InvalidBucketName	The bucket name is invalid.	400
InvalidDigest	The digest is invalid.	400

Error code	Description	HTTP status code
InvalidEncryptionAlgorithmError	The specified entropy encryption algorithm is invalid.	400
InvalidObjectName	The object name is invalid.	400
InvalidPart	The part is invalid.	400
InvalidPartOrder	The part order is invalid.	400
InvalidPolicyDocument	The policy document is invalid.	400
InvalidTargetBucketForLogging	A target bucket for logging is invalid.	400
InternalError	An error occurs in OSS.	500
MalformedXML	The XML format is invalid.	400
MalformedPOSTRequest	The format of the body in the post request is invalid.	400
MaxPOSTPreDataLengthExceededError	The body, except the object content, uploaded in the post request is too large.	400
MethodNotAllowed	The method is not supported.	405
MissingArgument	A parameter is missing.	411
MissingContentLength	The content length is missing.	411
NoSuchBucket	The bucket does not exist.	404
NoSuchKey	The object does not exist.	404
NoSuchUpload	The multipart upload ID does not exist.	404
NotImplemented	The method cannot be implemented.	501
PreconditionFailed	The preconditioning fails.	412
RequestTimeTooSkewed	Client local time and OSS server time differ by more than 15 minutes.	403
RequestTimeout	The request times out.	400

Error code	Description	HTTP status code
RequestIsNotMultiPartContent	The content type in the post request is invalid.	400
SignatureDoesNotMatch	The signature is incorrect.	403
TooManyBuckets	The number of the user's buckets exceeds the limit.	400
InvalidEncryptionAlgorithmError	The specified entropy encryption algorithm is invalid.	400

**Note:**

In the preceding table, the `Error code` is `OssServiceError.Code`, and the `HTTP status code` is `OssServiceError.StatusCode`.

5 C++

5.1 Overview

This document is written based on C++ SDK V1.2.0.

SDK source code

For the SDK source code, see [GitHub](#).

Sample code

The OSS C++ SDK provides a variety of sample code for your reference or use. The following table describes the content in the sample code.

Sample file	Content
PutObjectFromFile	Upload objects
GetObjectToFile	Download objects
AppendObject	Append upload
MultipartUploadObject	Multipart upload
HeadObject	Manage Object Meta
ListObjects	List objects
CopyObject	Copy an object
DeleteObject	Delete an object
PutObjectProgress	View upload progress
DownloadObjectProcess	View download progress
RestoreArchiveObject	Restore an archive object
BucketSample.cc	Set static website hosting, CORS, lifecycle, access logging, and authorized access for a bucket, as shown in Static website hosting , CORS , Lifecycle , Access logging , and Authorized access .

5.2 Installation

This topic describes how to install the OSS C++ SDK.

Prerequisites

Install a compiler that supports C++ V11 and later.

- Visual Studio V2013 and later
- GCC V4.8 and later
- Clang V3.3 and later

Download the SDK

[Download the SDK directly](#)

[Download from GitHub](#)

Install the SDK

You can install the SDK in Linux, Windows, Android, and Mac.

- **Linux:**

1. Install CMake.

Install CMake V3.1 and later. Open the SDK installation package and use CMake to compile and generate the required files.

The compilation commands are as follows:

```
cd < path / to / aliyun - oss - cpp - sdk >
mkdir build
cd build
```



```
cmake ..
```

2. Install third-party libraries libcurl and openssl.

RedHat/Centos:

```
yum -y install libcurl - devel openssl - devel
```

Fedora:

```
sudo dnf install libcurl - devel openssl - devel
```

3. Install the SDK.

```
make && make install
```



Note:

RTTI is disabled in the C++ SDK by default. Therefore, add `-std=c++11, -fno-rtti`, and `-laliabaclo ud - oss - cpp - sdk` when you use `g++` to compile and run the SDK. Example:

```
g++ test.cpp -std=c++11 -fno-rtti -laliabaclo ud - oss - cpp - sdk -o test.bin
```

• Windows:



Note:

The `alibabaclo ud - oss - cpp - sdk . sln` project file is not included in the downloaded SDK package. You must run the `cmake` command to generate the required project file.

1. Install CMake. Open `cmd` to go to the directory that stores the SDK files, create a folder named `build`, and run `cmake` to generate the required files, as shown in the following figure.

```
D:\Item\C++\aliyun-oss-cpp-sdk\build>cmake ..
-- Selecting Windows SDK version 10.0.17134.0 to target Windows 6.1.7601.
-- Project version: 1.1.0
-- TARGET_OS: WINDOWS
-- Configuring done
-- Generating done
-- Build files have been written to: D:/Item/C++/aliyun-oss-cpp-sdk/build
```



Note:

To construct project files in the x64 architecture, run the `cmake -A x64 ..` command.

2. Run VS developer command prompt as an administrator. Go to the build folder and run the following command to compile and install the SDK:

```
msbuild ALL_BUILD . vcxproj
msbuild INSTALL . vcxproj
```

You can also use Visual Studio to open `alibabacloud-oss-cpp-sdk.sln` to generate a solution.

- **Android:**

Build a project based on the `android-ndk-r16` tool chain in Linux. Install third-party libraries `libcurl` and `libopenssl` in `$ ANDROID_NDK / sysroot` and run the following command to compile and install the SDK:

```
cmake -DCMAKE_TOOLCHAIN_FILE=$ANDROID_NDK/build/cmake/android.toolchain.cmake \
  -DANDROID_NDK=$ANDROID_NDK \
  -DANDROID_ABI=armeabi-v7a \
  -DANDROID_TOOLCHAIN=clang \
  -DANDROID_PLATFORM=android-21 \
  -DANDROID_STL=c++_shared ..
make
```

- **Mac:**

In Mac, you must specify the path where OpenSSL is installed.

For example, if OpenSSL is installed in `/usr/local/Cellar/openssl/1.0.2p`, run the following command to compile and install the SDK:

```
cmake -DOPENSSL_ROOT_DIR=/usr/local/Cellar/openssl/1.0.2p \
  -DOPENSSL_LIBRARIES=/usr/local/Cellar/openssl/1.0.2p/lib \
  -DOPENSSL_INCLUDE_DIR=/usr/local/Cellar/openssl/1.0.2p/include ..
```

```
make
```

5.3 Quick start

This topic describes how to use the OSS C++ SDK to complete routine operations such as bucket creation, file uploads, and object downloads.

Create a bucket

A bucket is a global namespace in OSS. It is similar to a data container that stores files. Use the following code to create a bucket:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Specify the name , storage class , and ACL of the bucket . */
    CreateBucketRequest request ( BucketName , StorageClass :: IA , CannedAccessControlList :: PublicReadWrite );

    /* Create the bucket . */
    auto outcome = client . CreateBucket ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " CreateBucket fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
            :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

For more information about the bucket naming rules, see the "Naming conventions" section in [Basic concepts](#). For more information about how to create a bucket, see [#unique_317](#).

For more information about how to obtain endpoint information, see [Regions and endpoints](#).

Upload a file

Use the following code to upload a file to OSS:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Upload the file . */
    auto outcome = client . PutObject ( BucketName , ObjectName , yourLocalFilename );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " PutObject fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

Download an object

Use the following code to download an object to a local file:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;
```

```

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    /* Set the name of the object you want to
download . */
    std :: string FileNameTo Save = " yourFileNa me ";

    /* Initialize network resources . */
    Initialize Sdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );

    /* Download the object to the local file . */
    GetObjectRequest request ( BucketName , ObjectName );
    request . setResponseStreamFactory ([=]() { return std ::
make_shared < std :: fstream > ( FileNameTo Save , std :: ios_base
:: out | std :: ios_base :: in | std :: ios_base :: trunc | std
:: ios_base :: binary ); });

    auto outcome = client . GetObject ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " GetObjectToFile success , ContentLen
gth is " << outcome . result (). Metadata (). ContentLen gth ()
<< std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GetObjectToFile fail " <<
        " , code : " << outcome . error (). Code () <<
        " , message : " << outcome . error (). Message () <<
        " , requestId : " << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

List objects

Use the following code to list objects in the specified bucket: You can list a maximum of 100 objects by default.

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";

```

```

std::string AccessKeySecret = "yourAccessKeySecret";
std::string Endpoint = "yourEndpoint";
std::string BucketName = "yourBucketName";

/* Initialize network resources. */
InitializeSdk();

ClientConfiguration conf;
OssClient client(Endpoint, AccessKeyId, AccessKeySecret, conf);

/* List the objects in the specified bucket. */
ListObjectsRequest request(BucketName);
auto outcome = client.ListObjects(request);

if (!outcome.IsSuccess()) {
    /* Handle exceptions. */
    std::cout << "ListObjects fail " <<
        ", code:" << outcome.error().Code() <<
        ", message:" << outcome.error().Message() <<
        ", requestId:" << outcome.error().RequestId() << std::endl;
    ShutdownSdk();
    return -1;
}
else {
    for (const auto & object : outcome.result().ObjectSummaries()) {
        std::cout << "object " <<
            ", name:" << object.Key() <<
            ", size:" << object.Size() <<
            ", last modified time:" << object.LastModified() << std::endl;
    }
}

/* Release network resources. */
ShutdownSdk();
return 0;
}

```

Delete an object

Use the following code to delete the specified object:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main(void)
{
    /* Initialize the OSS account information. */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";
    std::string ObjectName = "yourObjectName";

    /* Initialize network resources. */
    InitializeSdk();

    ClientConfiguration conf;

```

```

OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

DeleteObjectRequest request ( BucketName , ObjectName );

/* Delete the object . */
auto outcome = client . DeleteObject ( request );

if ( ! outcome . isSuccess () ) {
    /* Handle exceptions . */
    std :: cout << " DeleteObject fail " <<
        ", code :" << outcome . error () . Code () <<
        ", message :" << outcome . error () . Message () <<
        ", requestId :" << outcome . error () . RequestId () << std
:: endl ;
    ShutdownSdk ();
    return - 1 ;
}

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

For more information, see the "[Delete an object](#)" section in Manage files.

5.4 Initialization

OSSClient is used to manage OSS resources such as buckets and objects. To use the C++ SDK to initiate a request, you must initialize an OSSClient instance and modify the default configuration items of ClientConfiguration as needed.

Configure the OSSClient instance

ClientConfiguration is a configuration class of OSSClient. You can use ClientConfiguration to configure parameters such as user agents, host proxies, connection timeout, and the maximum number of connections. You can configure the following parameters.

Name	Description
userAgent	The user agent (the User-Agent field in the HTTP header). The default value is aliyun-sdk-cpp/1. X.X
maxConnections	The maximum number of connection pools. The default value is 16.
requestTimeoutMs	The request timeout period. The connection is closed if no data is received during the timeout period. The default value is 10,000 ms.

Name	Description
<code>connectTimeoutMs</code>	The timeout period to establish a connection. The default value is 5,000 ms.
<code>retryStrategy</code>	The maximum number of retry attempts in the case of a request error.
<code>proxyScheme</code>	The proxy protocol. The default value is HTTP.
<code>proxyPort</code>	The port for the proxy host.
<code>proxyPassword</code>	The password for proxy host verification.
<code>proxyUserName</code>	The username for proxy host verification.
<code>verifySSL</code>	Indicates whether to enable SSL verification. SSL verification is disabled by default.
<code>enableCrc64</code>	Indicates whether to enable the CRC64 check. The CRC64 check is enabled by default.
<code>enableDateSkewAdjustment</code>	Indicates whether to enable automatic correction of HTTP request time. This function is enabled by default.
<code>sendRateLimiter</code>	The limit to the upload speed, in Kbit/s.
<code>recvRateLimiter</code>	The limit to the download speed, in Kbit/s.

Configure a timeout period

Use the following code to configure a timeout period:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;
int main ( void )
{
    /* Initialize network resources . */
    InitializeSdk ();
    ClientConfiguration conf ;

    /* Set the maximum number of connection pools . The
    default value is 16 . */
    conf . maxConnections = 20 ;
```



```

    /* Configure the request timeout period . If no data
    is received during the period , the connection is
    closed . The default value is 10 , 000 ms . */
    conf . requestTimeoutMs = 8000 ;

    /* Configure the timeout period in milliseconds
    when connections are established . The default value
    is 5 , 000 ms . */
    conf . connectTimeoutMs = 8000 ;

    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Configure speed limits

Use the following code to configure speed limits for uploads and downloads:

```

# include < alibabacloud / oss / OssClient . h >
# include < alibabacloud / oss / client / RateLimiter . h >

using namespace AlibabaCloud :: OSS ;

class UserRateLimiter : public RateLimiter
{
public :
    DefaultRateLimiter () : rate_ ( 0 ) {} ;
    ~DefaultRateLimiter () {} ;
    virtual void setRate ( int rate ) { rate_ = rate ; } ;
    virtual int Rate () const { return rate_ ; } ;
private :
    int rate_ ;
};

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId " ;
    std :: string AccessKeySecret = " yourAccess KeySecret " ;
    std :: string Endpoint = " yourEndpoint " ;
    std :: string BucketName = " yourBucket Name " ;
    std :: string ObjectName = " yourObjectName " ;

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;

    auto sendrateLimiter = std :: make_shared < UserRateLimiter > ();
    auto recvrateLimiter = std :: make_shared < UserRateLimiter > ();
    conf . sendRateLimiter = sendrateLimiter ;
    conf . recvRateLimiter = recvrateLimiter ;

    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
}

```

```

/* Set the speed limit for the download , in Kbit / s . */
recvrateLi miter -> setRate ( 256 );

/* Set the speed limit for the upload , in Kbit / s . */
sendrateLi miter -> setRate ( 256 );

/* Upload the file . */
auto outcome = client . PutObject ( BucketName , ObjectName
," yourLocalF ilename ");

/* Update the speed limit during the upload , in Kbit / s . */
sendrateLi miter -> setRate ( 300 );

/* Release network resources . */
ShutdownSd k ();
return 0 ;
}

```

Configure a retry policy

Use the following code to configure a retry policy:

```

# include < alibabacloud / oss / OssClient . h >
# include < alibabacloud / oss / client / RetryStrategy . h >

using namespace AlibabaCloud :: OSS ;

class UserRetryStrategy : public RetryStrategy
{
public :

    /* maxRetries indicates the maximum number of retry
    attempts . scaleFactor indicates the factor that is
    used to calculate the waiting time of a retry
    attempt . */
    DefaultRetryStrategy ( long maxRetries = 3 , long
scaleFactor = 300 ) :
        m_scaleFactor ( scaleFactor ), m_maxRetries (
maxRetries )
    {}

    /* You can define the shouldRetry function to
    determine whether to retry a request . */
    bool shouldRetry ( const Error & error , long
attemptedRetries ) const ;

    /* You can define the calcDelayTimeMs function to
    calculate the waiting time ( delay ) of a retry attempt
    . */
    long calcDelayTimeMs ( const Error & error , long
attemptedRetries ) const ;

private :
    long m_scaleFactor ;
    long m_maxRetries ;
};

bool DefaultRetryStrategy :: shouldRetry ( const Error &
error , long attemptedRetries ) const

```

```

{
    if ( attemptedRetries >= m_maxRetries )
        return false ;

    long responseCode = error . Status () ;

    // http code
    if (( responseCode == 403 && error . Message (). find ( "
RequestTimeTooSkewed " ) ) ||
        ( responseCode > 499 && responseCode < 599 ) ) {
        return true ;
    }
    else {
        switch ( responseCode )
        {
            // curl error code
            case ( ERROR_CURL_BASE + 7 ) : // CURLE_COULDNT_CONNECT
            case ( ERROR_CURL_BASE + 18 ) : // CURLE_PARTIAL_FILE
            case ( ERROR_CURL_BASE + 23 ) : // CURLE_WRITE_ERROR
            case ( ERROR_CURL_BASE + 28 ) : // CURLE_OPERATION_TIMEOUTED
            case ( ERROR_CURL_BASE + 52 ) : // CURLE_GOT_NOTHING
            case ( ERROR_CURL_BASE + 55 ) : // CURLE_SEND_ERROR
            case ( ERROR_CURL_BASE + 56 ) : // CURLE_RECV_ERROR
                return true ;
            default :
                break ;
        }
    }

    return false ;
}

long DefaultRetryStrategy :: calcDelayTimeMs ( const Error
& error , long attemptedRetries ) const
{
    UNUSED_PARAMETER ( error ) ;
    return ( 1 << attemptedRetries ) * m_scaleFactor ;
}

int main ( void )
{
    /* Initialize network resources . */
    InitializeSdk () ;

    ClientConfiguration conf ;

    /* The maximum number of retry attempts in the
    case of a request error . The default value is 3 .
    */
    auto defaultRetryStrategy = std :: make_shared <
UserRetryStrategy > ( 5 ) ;
    conf . retryStrategy = defaultRetryStrategy ;

    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf ) ;

    /* Release network resources . */
    ShutdownSdk () ;
    return 0 ;
}

```

```
}
```

5.5 Buckets

A bucket serves as a container that stores objects. Objects belong to the bucket they are created in or uploaded to.

Create a bucket

Use the following code to create a bucket:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Specify the name , storage class , and ACL of the bucket . */
    CreateBucketRequest request ( BucketName , StorageClass :: IA , CannedAccessControlList :: PublicReadWrite );

    /* Create the bucket . */
    auto outcome = client . CreateBucket ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " CreateBucket fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

For more information about the bucket naming rules, see the "Naming conventions" section in [Basic concepts](#). You can specify the ACL and the storage class when you create a bucket, as shown in [Bucket ACLs](#) and [Storage classes](#).

List buckets

Use the following code to list all buckets:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* List the buckets . */
    ListBucketsRequest request ;
    auto outcome = client . ListBuckets ( request );

    if ( outcome . isSuccess () ) {
        /* Display the bucket information . */
        std :: cout << " success , and bucket count is " <<
outcome . result (). Buckets (). size () << std :: endl ;
        std :: cout << " Bucket name is " << std :: endl ;
        for ( auto result : outcome . result (). Buckets () )
        {
            std :: cout << result . Name () << std :: endl ;
        }
    }
    else {
        /* Handle exceptions . */
        std :: cout << " ListBuckets fail " <<
        ", code : " << outcome . error (). Code () <<
        ", message : " << outcome . error (). Message () <<
        ", requestId : " << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

Configure the ACL of a bucket

The following table describes the permissions included in the ACL of a bucket.

Permission	Description	Value
Private	Only the owner or authorized users of a bucket have read and write permissions on objects in the bucket.	OSS_ACL_PRIVATE
Public Read	Only the owner or authorized users of a bucket have read and write permissions on objects in the bucket. Other users (including anonymous users) have only read permissions on objects in the bucket. Use caution when using this permission.	OSS_ACL_PUBLIC_READ
Public	All users of a bucket have read and write permissions on all objects in the bucket. Use caution when using this permission.	OSS_ACL_PUBLIC_READ_WRITE

For more information about the ACL, see the "[Access control](#)" section in OSS Developer Guide.

Use the following code to configure the ACL of a bucket:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Configure the ACL of the bucket . */
    SetBucketAclRequest request ( BucketName , CannedAccessControlList :: Private );
    auto outcome = client . SetBucketAcl ( request );
```

```

        if ( outcome . isSuccess () ) {
            std :: cout << " setBucketA cl to private success "
<< std :: endl ;
        }
        else {
            /* Handle exceptions . */
            std :: cout << " SetBucketA cl fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
            ShutdownSd k ();
            return - 1 ;
        }

        /* Release network resources . */
        ShutdownSd k ();
        return 0 ;
    }
}

```

Obtain the ACL of a bucket

Use the following code to obtain the ACL of a bucket:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Obtain the ACL of the bucket . */
    GetBucketAclRequest request ( BucketName );
    auto outcome = client . GetBucketAcl ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " getBucketAcl success , acl : " <<
outcome . result (). Acl () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " getBucketAcl fail " <<
        ", code :" << outcome . error (). Code () <<
        ", message :" << outcome . error (). Message () <<
        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSd k ();
        return - 1 ;
    }
}

```

```

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Obtain the region of a bucket

Use the following code to obtain the region where a bucket resides:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Obtain the region where the bucket resides . */
    GetBucketLocationRequest request ( BucketName );
    auto outcome = client . GetBucketLocation ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " getBucketLocation success , location : " << outcome . result (). Location () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " getBucketLocation fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Determine whether a bucket exists

Use the following code to determine whether the specified bucket exists:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */

```



```

std :: string    AccessKeyId = "yourAccessKeyId";
std :: string    AccessKeySecret = "yourAccessKeySecret";
std :: string    Endpoint = "yourEndpoint";
std :: string    BucketName = "yourBucketName";

/* Initialize network resources . */
InitializeSdk();

ClientConfiguration conf;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

/* Determine whether the bucket exists . */
auto outcome = client . DoesBucketExist ( BucketName );

if ( outcome ) {
    std :: cout << "The Bucket exists " << std :: endl ;
}
else {
    std :: cout << "The Bucket does not exist " << std :: endl ;
}

/* Release network resources . */
ShutdownSdk();
return 0 ;
}

```

Delete a bucket

Before deleting a bucket, you must delete all objects in the bucket and fragments generated by multipart upload.



Note:

To delete the fragments that are generated from multipart upload, use [ListMultipartUploads](#) to list all fragments, and then use [AbortMultipartUpload](#) to delete the fragments.

Use the following code to delete a bucket:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string    AccessKeyId = "yourAccessKeyId";
    std :: string    AccessKeySecret = "yourAccessKeySecret";
    std :: string    Endpoint = "yourEndpoint";
    std :: string    BucketName = "yourBucketName";

    /* Initialize network resources . */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
}

```

```

    /* Specify the name of the bucket you want to
delete . */
    DeleteBucket request ( BucketName );

    /* Delete the bucket . */
    auto outcome = client . DeleteBucket ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " DeleteBucket fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

5.6 Single-connection bandwidth throttling

This topic describes how to ensure other applications have sufficient bandwidth when you perform upload and download operations on OSS. You can include the `x-oss-traffic-limit` parameter in your requests and set bandwidth limits.



Note:

For more information about the scenarios and precautions of the single-connection bandwidth throttling feature, see [#unique_47](#) in the OSS Developer Guide.

Configure bandwidth throttling for common uploads and downloads

Use the following code to configure bandwidth throttling when you upload and download objects to and from OSS:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;

```

```

    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Upload the object . */
    std::shared_ptr< std::iostream > content = std::make_shared< std::fstream >("yourLocalFilename", std::ios::in | std::ios::binary);
    PutObjectRequest request ( BucketName , ObjectName , content );
    /* Set the upload bandwidth limit to 100 KB / s . */
    request.SetTrafficLimit ( 819200 );
    auto putOutcome = client.PutObject ( request );

    /* Download the object to local memory . */
    GetObjectRequest request ( BucketName , ObjectName );
    /* Set the download bandwidth limit to 100 KB / s . */
    request.SetTrafficLimit ( 819200 );
    auto getOutcome = client.GetObject ( request );

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Configure bandwidth throttling for uploads and downloads that use signed URLs

Use the following code to configure bandwidth throttling when you upload and download objects from and to OSS by using signed URLs.

```

#include <alibabacloud_oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";
    std::string PutObjectUrlName = "yourPutObjectUrlName";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Specify the valid period for the URL . */
    std::time_t t = std::time ( nullptr ) + 1200 ;
    /* Generate the signed URL for uploading an object . */
    GeneratePresignedUrlRequest request ( BucketName , PutObjectUrlName , Http::Put );
    request.SetExpires ( expires );
    /* Set the upload bandwidth limit to 100 KB / s . */
    request.SetTrafficLimit ( 819200 );
    auto genOutcome = client.GeneratePresignedUrl ( request );
}

```

```

    std::shared_ptr< std::iostream > content = std::
make_shared< std::stringstream >();
    *content << " test  cpp  sdk ";

    /* Use the signed URL to upload the object. */
    auto outcome = client.PutObjectByUrl ( genOutcome .
result (), content );

    /* Generate the signed URL for downloading an
object. */
    GeneratePresignedUrlRequest getrequest ( BucketName ,
PutObjectUrlName , Http::Get );
    getrequest.setExpires ( expires );
    /* Set the download bandwidth limit to 100 KB / s .
*/
    getrequest.setTrafficLimit ( 819200 );
    genOutcome = client.GeneratePresignedUrl ( getrequest );

    /* Use the signed URL to download the object. */
    outcome = client.GetObjectByUrl ( genOutcome . result () );

    /* Release network resources. */
    ShutdownSdk ();
    return 0 ;
}

```

5.7 Object tagging

5.7.1 Configure object tagging

This topic describes how to configure object tagging.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).



Note:

- You can add tags to an object when uploading it or add tags for an uploaded object. If add tags to an object that already has tags, the original tags are overwritten. For more information about adding tags, see [#unique_52](#).
- To add tags to an object, you must have the permission to call PutObjectTagging.
- The Last-Modified value of an object is not updated when its tags are changed.
- A tag can contain letters, numbers, spaces, and the following symbols: + - = . _ : /

Add tags to an object when uploading it

- Add tags to an object when uploading it directly.

Run the following code to add tags to an object when uploading it directly:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ();
    * content << " test cpp sdk ";
    PutObjectRequest request ( BucketName , ObjectName ,
content );

    /* Sets x - oss - tagging to configure the tags to
    be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 " , " value1 " ));
    tagging . addTag ( Tag ( " key2 " , " value2 " ));
    request . setTagging ( tagging . toQueryParameters ());

    /* Uploads the object . */
    auto outcome = client . PutObject ( request );

    if (! outcome . isSuccess ()) {
        /* Handles exceptions . */
        std :: cout << " PutObject fail " <<
        ", code : " << outcome . error (). Code () <<
        ", message : " << outcome . error (). Message () <<
        ", requestId : " << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Releases resources such as networks . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

- Add tags to an object when uploading it in the multipart upload method.

Run the following code to add tags to an object when uploading it in the multipart upload method:

```
# include < alibabacloud / oss / OssClient . h >

int64_t getFileSize ( const std :: string & file )
{
    std :: fstream f ( file , std :: ios :: in | std :: ios ::
binary );
    f . seekg ( 0 , f . end );
    int64_t size = f . tellg ();
    f . close ();
    return size ;
}

using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );
    InitiateMultipartUploadRequest initUploadRequest (
BucketName , ObjectName );

    /* Sets x - oss - tagging to configure the tags to
be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 " , " value1 " ));
    tagging . addTag ( Tag ( " key2 " , " value2 " ));
    initUploadRequest . setTagging ( tagging . toQueryPar
ameters ());

    /* Initialize s a multipart upload task . */
    auto uploadIdResult = client . InitiateMultipartUplo
ad ( initUploadRequest );
    auto uploadId = uploadIdResult . result (). UploadId ();
    std :: string fileToUpload = " yourLocalFilename ";
    int64_t partSize = 100 * 1024 ;
    PartList partEtagList ;
    auto fileSize = getFileSize ( fileToUpload );
    int partCount = static_cast < int > ( fileSize /
partSize );
    /* Calculates the number of parts . */
    if ( fileSize % partSize != 0 ) {
        partCount ++;
    }
}
```

```

/* Uploads each part . */
for ( int i = 1 ; i <= partCount ; i ++ ) {
    auto skipBytes = partSize * ( i - 1 );
    auto size = ( partSize < fileSize - skipBytes ) ?
partSize : ( fileSize - skipBytes );
    std :: shared_ptr < std :: istream > content = std ::
make_share d < std :: fstream >( fileToUplo ad , std :: ios ::
in | std :: ios :: binary );
    content -> seekg ( skipBytes , std :: ios :: beg );

    UploadPart Request uploadPart Request ( BucketName ,
ObjectName , content );
    uploadPart Request . setContent Length ( size );
    uploadPart Request . setUploadI d ( uploadId );
    uploadPart Request . setPartNum ber ( i );
    auto uploadPart Outcome = client . UploadPart (
uploadPart Request );
    if ( uploadPart Outcome . isSuccess () ) {
        Part part ( i , uploadPart Outcome . result ().
ETag ());
        partETagLi st . push_back ( part );
    }
    else {
        std :: cout << " uploadPart fail " <<
        ", code : " << uploadPart Outcome . error (). Code ()
<<
        ", message : " << uploadPart Outcome . error ().
Message () <<
        ", requestId : " << uploadPart Outcome . error ().
RequestId () << std :: endl ;
    }

}

/* Completes the multipart upload task . */
CompleteMu ltipartUpl oadRequest request ( BucketName ,
ObjectName );
request . setUploadI d ( uploadId );
request . setPartLis t ( partETagLi st );
auto outcome = client . CompleteMu ltipartUpl oad (
request );

if ( ! outcome . isSuccess () ) {
    /* Handles exceptions . */
    std :: cout << " CompleteMu ltipartUpl oad fail " <<
    ", code : " << outcome . error (). Code () <<
    ", message : " << outcome . error (). Message () <<
    ", requestId : " << outcome . error (). RequestId () << std
:: endl ;
    ShutdownSd k ();
    return - 1 ;
}

/* Releases resources such as networks . */
ShutdownSd k ();
return 0 ;

```

```
}
```

- Add tags to an object when uploading it by calling `append_object`.

Run the following code to add tags to an object when uploading it in the `append_object` method:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    auto meta = ObjectMetadata ();
    meta . setContent Type ( " text / plain " );

    /* Appends the file to the object . The default
    position from where the file is appended is 0 . */
    std :: shared_ptr < std :: istream > content1 = std ::
make_shared < std :: stringstream > ();
    * content1 << " Thank you for using Aliyun Object
Storage Service ! ";
    AppendObjectRequest request ( BucketName , ObjectName ,
content1 , meta );

    /* Configures the tags to be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 " , " value1 " ));
    tagging . addTag ( Tag ( " key2 " , " value2 " ));
    request . setTagging ( tagging . toQueryParameters ());

    /* Appends the file to the object . */
    auto result = client . AppendObject ( request );

    if ( ! result . isSuccess ()) {
        /* Handles exceptions . */
        std :: cout << " AppendObject fail " <<
        ", code : " << result . error (). Code () <<
        ", message : " << result . error (). Message () <<
        ", requestId : " << result . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Releases resources such as networks . */
    ShutdownSdk ();
    return 0 ;
}
```



```
}
```

- Add tags to an object when uploading it in the resumable upload method.

Run the following code to add tags to an object when uploading it in the resumable upload method:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    std :: string UploadFilePath = " yourUpload filePath ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Uploads the object in the resumable upload
    method . */
    UploadObjectRequest request ( BucketName , ObjectName ,
    YourUpload FileName );

    /* Configures the tags to be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 " , " value1 " ));
    tagging . addTag ( Tag ( " key2 " , " value2 " ));
    request . setTagging ( tagging . toQueryParameters ());

    auto outcome = client . ResumableUploadObject ( request
    );

    if ( ! outcome . isSuccess ( )) {
        /* Handles exceptions . */
        std :: cout << " ResumableUploadObject fail " <<
        ", code : " << outcome . error ( ) . Code ( ) <<
        ", message : " << outcome . error ( ) . Message ( ) <<
        ", requestId : " << outcome . error ( ) . RequestId ( ) << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Releases resources such as networks . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

Add tags to an uploaded object or modify the tags added to an object

Run the following code to add tags to an uploaded object or modify the tags added to an object:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ();
    * content << " test cpp sdk ";
    PutObjectRequest request ( BucketName , ObjectName ,
content );

    /* Uploads the object . */
    auto outcome = client . PutObject ( request );

    /* Configures the tags to be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 " , " value1 " ));
    tagging . addTag ( Tag ( " key2 " , " value2 " ));
    auto putTagging Outcome = client . SetObjectTagging (
SetObjectTaggingRequest ( BucketName , ObjectName , tagging ));

    if (! putTagging Outcome . isSuccess ()) {
        /* Handles exceptions . */
        std :: cout << " SetObjectTagging fail " <<
        " , code : " << putTagging Outcome . error (). Code () <<
        " , message : " << putTagging Outcome . error (). Message ()
<<
        " , requestId : " << putTagging Outcome . error (). RequestId
() << std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Releases resources such as networks . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

Add tags to an object when copying it

You can specify one of the following tagging rules when copying an object:

- **Copy (default):** Copy the tags of the source object to the destination object.
- **Replace:** Add the tags specified in the request to the destination object.

Examples of adding tags to objects of different sizes when copying them are described as follows:

- Run the following code to add tags to an object smaller than 1 GB when directly copying it:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    std :: string CopyObjectName = " yourCopyOb jectName ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );
    std :: shared_ptr < std :: iostream > content = std ::
make_shared < std :: stringstream > ();
    * content << " test cpp sdk ";
    PutObjectRequest request ( BucketName , ObjectName ,
content );

    /* Sets x - oss - tagging to configure the tags to
    be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 ", " value1 " ));
    tagging . addTag ( Tag ( " key2 ", " value2 " ));
    request . setTagging ( tagging . toQueryPar ameters ( ));

    /* Uploads the object . */
    auto outcome = client . PutObject ( request );

    /* Copies the object and sets the tags to be
    added . */
    CopyObjectRequest cpRequest ( yourBucket Name ,
CopyObject Name );
    cpRequest . setCopySou rce ( yourBucket Name , yourObject
Name );
    Tagging tagging2 ;
    tagging2 . addTag ( Tag ( " key1 - 2 ", " value1 - 2 " ));
```

```

        tagging2 . addTag ( Tag ( " key2 - 2 ", " value2 - 2 "));
        tagging2 . addTag ( Tag ( " key3 - 2 ", " value3 - 2 "));
        cpRequest . setTagging ( tagging2 . toQueryParameters ());

    /* Ignores the tags of the source object . Adds
    the tags specified in the request to the object .
    */
    cpRequest . setTagging Directive ( CopyAction List :: Replace
    );

    /* Copies the object . */
    auto copyoutcome = client . CopyObject ( cpRequest );
    if (! copyoutcome . isSuccess ()) {
        /* Handles exceptions . */
        std :: cout << " CopyObject fail " <<
        ", code :" << copyoutcome . error (). Code () <<
        ", message :" << copyoutcome . error (). Message () <<
        ", requestId :" << copyoutcome . error (). RequestId ()
    << std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Releases resources such as networks . */
    ShutdownSdk ();
    return 0 ;
}

```

- Run the following code to add tags to an object larger than 1 GB when copying it the multipart upload method:

```

# include < alibabacloud / oss / OssClient . h >
# include < sstream >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string SourceBucketName = " yourSource BucketName
";
    std :: string CopyBucketName = " yourCopyBucketName ";
    std :: string SourceObjectName = " yourSource ObjectName
";
    std :: string CopyObjectName = " yourCopyObject Name ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );

    auto getObjectMultipartReq = GetObjectMultipartRequest (
SourceBucketName , SourceObjectName );
    auto getObjectMultipartResult = GetObjectMultipart ( getObjectM
ultipartReq );
    if (! getObjectMultipartResult . isSuccess ()) {
        std :: cout << " GetObjectMultipart fail " <<

```

```

        ", code :" << outcome . error (). Code () <<
        ", message :" << outcome . error (). Message () <<
        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
    /* Obtains the size of the source object . */
    auto objectSize = getObjectMetadataResult . result ().
ContentLength ();

    /* Copies the large object . */
    ObjectMetadata metadata ;
    std :: string stream ssDesc ;
    ssDesc << "/" << SourceBucketName << "/" << SourceObject
Name ;
    std :: string value = ssDesc . str ();
    metadata . addHeader ( " x - oss - copy - source ", value );
    InitiateMultipartUploadRequest initUploadRequest (
BucketName , ObjectName , metadata );

    /* Sets x - oss - tagging to configure the tags to
be added . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 ", " value1 " ));
    tagging . addTag ( Tag ( " key2 ", " value2 " ));
    initUploadRequest . setTagging ( tagging . toQueryPar
ameters ());

    /* Initialize a multipart copy task . */
    auto uploadIdResult = client . InitiateMultipartUp
load ( initUploadRequest );
    auto uploadId = uploadIdResult . result (). UploadId ();
    int64_t partSize = 100 * 1024 ;
    PartList partETagList ;
    int partCount = static_cast < int > ( objectSize /
partSize );
    /* Calculates the number of parts . */
    if ( objectSize % partSize != 0 ) {
        partCount ++;
    }

    /* Copies each part . */
    for ( int i = 1 ; i <= partCount ; i ++ ) {
        auto skipBytes = partSize * ( i - 1 );
        auto size = ( partSize < objectSize - skipBytes ) ?
partSize : ( objectSize - skipBytes );
        auto uploadPartCopyReq = UploadPartCopyRequest (
CopyBucketName , CopyObjectName , SourceBucketName ,
SourceObjectName , uploadId , i + 1 );
        uploadPartCopyReq . setCopySourceRange ( skipBytes ,
skipBytes + size - 1 );
        auto uploadPartOutcome = client . UploadPartCopy (
uploadPartCopyReq );
        if ( uploadPartOutcome . isSuccess () ) {
            Part part ( i , uploadPartOutcome . result ().
ETag ());
            partETagList . push_back ( part );
        }
        else {
            std :: cout << " UploadPart Copy fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<

```

```

        ", requestId :" << outcome . error (). RequestId () <<
std :: endl ;
    }

}

/* Completes the multipart copy task . */
CompleteMu ltipartUpl oadRequest request ( CopyBucket Name
, CopyObject Name , partETagLi st , uploadId );
auto outcome = client . CompleteMu ltipartUpl oad (
request );

if (! outcome . isSuccess ()) {
    /* Handles exceptions . */
    std :: cout << " CompleteMu ltipartUpl oad fail " <<
    ", code :" << outcome . error (). Code () <<
    ", message :" << outcome . error (). Message () <<
    ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
    ShutdownSdk ();
    return - 1 ;
}

/* Releases resources such as networks . */
ShutdownSdk ();
return 0 ;
}

```

Add tags to a symbolic link object

Run the following code to add tags to a symbolic link object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    informatio n . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoi nt ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    std :: string LinkName = " yourLinkNa me ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );
    std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ();
    * content << " test cpp sdk ";
    PutObjectRequest request ( BucketName , ObjectName ,
content );

    /* Uploads the object . */
    auto outcome = client . PutObject ( request );

    /* Creates a symbolic link . */
}

```

```

CreateSymb linkRequest csRequest ( BucketName , ObjectName );
csRequest . SetSymb link Target ( LinkName );

/* Sets x - oss - tagging to configure the tags to
be added . */
Tagging tagging ;
tagging . addTag ( Tag ( " key1 " , " value1 " ));
tagging . addTag ( Tag ( " key2 " , " value2 " ));
csRequest . setTagging ( tagging . toQueryParameters ());

auto csoutcome = client . CreateSymb link ( csRequest );

if ( ! csoutcome . isSuccess ( )) {
    /* Handles exceptions . */
    std :: cout << " CreateSymb link fail " <<
    " , code : " << csoutcome . error ( ) . Code ( ) <<
    " , message : " << csoutcome . error ( ) . Message ( ) <<
    " , requestId : " << csoutcome . error ( ) . RequestId ( ) << std
:: endl ;
    ShutdownSdk ( );
    return - 1 ;
}

/* Releases resources such as networks . */
ShutdownSdk ( );
return 0 ;
}

```

5.7.2 Obtain the tags added to an object

This topic describes how to obtain the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to obtain the tags added to an object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize resources such as networks . */
    InitializeSdk ( );

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ( );
    * content << " test cpp sdk ";
    PutObjectRequest request ( BucketName , ObjectName ,
content );
}

```

```

/* Uploads the object . */
auto outcome = client . PutObject ( request );

/* Configures the tags to be added . */
Tagging tagging ;
tagging . addTag ( Tag ( " key1 ", " value1 " ));
tagging . addTag ( Tag ( " key2 ", " value2 " ));
auto putTagging Outcome = client . SetObjectT agging (
SetObjectT aggingRequ est ( BucketName , ObjectName , tagging ));

/* Obtains the tags added to the object . */
getTagging Outcome = client . GetObjectT agging ( GetObjectT
aggingRequ est ( BucketName , key ));

if (! getTagging Outcome . isSuccess ()) {
/* Handles exceptions . */
std :: cout << " getTagging Outcome fail " <<
", code :" << getTagging Outcome . error (). Code () <<
", message :" << getTagging Outcome . error (). Message ()
<<
", requestId :" << getTagging Outcome . error (). RequestId
() << std :: endl ;
ShutdownSd k ();
return - 1 ;
}
else {
auto taglist = getTagging Outcome . result (). Tagging
();
for ( const auto & tag : taglist )
{
std :: cout <<" GetObjectT agging success , Key :"
<< tag . Key () << " ; Value :" << tag . Value () << std
:: endl ;
}
}

/* Releases resources such as networks . */
ShutdownSd k ();
return 0 ;
}

```

For more information about how to obtain the tags added to an object, see [#unique_54](#).

5.7.3 Delete the tags added to an object

This topic describes how to delete the tags added to an object.

The object tagging function uses a key-value pair to tag an object. For more information about the object tagging function, see [#unique_51](#).

Run the following code to delete the tags added to an object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{

```



```

/* Initialize s the Alibaba Cloud OSS account
information . */
std :: string AccessKeyId = " yourAccess KeyId ";
std :: string AccessKeySecret = " yourAccess KeySecret ";
std :: string Endpoint = " yourEndpoint ";
std :: string BucketName = " yourBucket Name ";
std :: string ObjectName = " yourObject Name ";

/* Initialize s resources such as networks . */
InitializeSdk ();

ClientConfiguration conf ;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ();
* content << " test cpp sdk ";
PutObjectRequest request ( BucketName , ObjectName ,
content );

/* Uploads the object */
auto outcome = client . PutObject ( request );

/* Configures the tags to be added . */
Tagging tagging ;
tagging . addTag ( Tag ( " key1 " , " value1 " ));
tagging . addTag ( Tag ( " key2 " , " value2 " ));
auto putTagging Outcome = client . SetObjectTagging (
SetObjectTaggingRequest ( BucketName , ObjectName , tagging ));

/* Deletes the tags added to the object . */
auto delTagging Outcome = client . DeleteObjectTagging (
DeleteObjectTaggingRequest ( BucketName , ObjectName ));

if (! delTagging Outcome . isSuccess ()) {
/* Handles exceptions . */
std :: cout << " delTagging Outcome fail " <<
", code : " << delTagging Outcome . error (). Code () <<
", message : " << delTagging Outcome . error (). Message ()
<<
", requestId : " << delTagging Outcome . error (). RequestId
() << std :: endl ;
ShutdownSdk ();
return - 1 ;
}

/* Releases resources such as networks . */
ShutdownSdk ();
return 0 ;
}

```

For more information about how to delete the tags added to an object, see [#unique_56](#).

5.7.4 Object tagging and lifecycle rules

Lifecycle rules can take effect on objects with specified prefixes or tags. You can also specify prefixes and tags as the condition of a lifecycle rule at the same time.



Note:

If you set a specified tag as the condition of a lifecycle rule, the rule applies to an object only when the key and value in the tag of the object both match the specified tag. If you specify a prefix and multiple tags as the condition of a lifecycle rule, the rule applies to an object only when the prefix and tags of the object match the prefix and all tags specified in the rule.

Set tags as the matching condition of a lifecycle rule

Run the following code to set tags as the matching condition of a lifecycle rule:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    SetBucketLifecycleRequest request ( BucketName );

    /* Configures the tags to be set as the matching
    condition . */
    Tagging tagging ;
    tagging . addTag ( Tag ( " key1 " , " value1 " ));
    tagging . addTag ( Tag ( " key2 " , " value2 " ));

    /* Specifies the lifecycle rule for the object . */
    auto rule1 = LifecycleRule ();
    rule1 . setID ( " rule1 " );
    rule1 . setPrefix ( " test1 / " );
    rule1 . setStatus ( RuleStatus :: Enabled );
    rule1 . setExpiration ( 3 );
    rule1 . setTags ( tagging . Tags ());

    /* Sets the tags as the matching condition of the
    lifecycle rule . */
    LifecycleRuleList list { rule1 };
    request . setLifecycleRules ( list );
    auto outcome = client . SetBucketLifecycle ( request );
```

```

        if (! outcome . isSuccess ()) {
            /* Handles exceptions . */
            std :: cout << " SetBucketL ifecycle fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
            ShutdownSd k ();
            return - 1 ;
        }

        /* Releases resources such as networks . */
        ShutdownSd k ();
        return 0 ;
    }
}

```

View the tags set as the matching condition of a lifecycle rule

Run the following code to view the tags set as the matching condition of a lifecycle rule:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize s the Alibaba Cloud OSS account
    information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize s resources such as networks . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Views the lifecycle rule that includes the tags
    set as the matching condition . */
    auto outcome = client . GetBucketLifecycle ( BucketName );

    if ( outcome . isSuccess ()) {
        std :: cout << " GetBucketLifecycle success ," << std
:: endl ;
        for ( auto const rule : outcome . result ().
LifecycleRules ()) {
            std :: cout << " rule :" << rule . ID () << "," <<
rule . Prefix () << "," << rule . Status () << "," <<
            " hasExpiration :" << rule . hasExpiration () << "," <<
            " hasTransitionList :" << rule . hasTransitionList ()
<< "," << std :: endl ;

            auto taglist = rule . Tags ();
            for ( const auto & tag : taglist )
            {
                std :: cout << " GetBucketLifecycle tag success
, Key :"

```

```

        << tag . Key () << "; Value :" << tag . Value () <<
std :: endl ;
    }
}
else {
    /* Handles exceptions . */
    std :: cout << " GetBucketL ifecycle fail " <<
    ", code :" << outcome . error (). Code () <<
    ", message :" << outcome . error (). Message () <<
    ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
    ShutdownSd k ();
    return - 1 ;
}

/* Releases resources such as networks . */
ShutdownSd k ();
return 0 ;
}

```

5.8 Upload objects

5.8.1 Overview

Objects are the basic unit for data operations in OSS. The OSS C++ SDK provides the following object upload methods:

- **Simple upload:** supports the upload of a file up to 5 GB in size. It includes upload from memory and local file upload.
- **Append upload:** supports the upload of a file up to 5 GB in size.
- **Resumable upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files. It supports concurrent upload and you can define the size of each part.
- **Multipart upload:** supports the upload of files up to 48.8 TB in size. It applies to the upload of large files.



Note:

For more information about the scenarios of each upload method, see the "Upload files" section in OSS Developer Guide.

During the file upload, you can set Object Meta and view upload progress, as shown in [Configure Object Meta](#) and [Use progress bars](#). After you upload the file, you can perform upload callback, as shown in [Upload callback](#).

5.8.2 Simple upload

This topic describes how to upload a file from memory or a local disk to OSS.

Upload from memory

Use the following code to upload a file from memory to OSS:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ();
    * content << " Thank you for using Aliyun Object
Storage Service ! " ;
    PutObjectRequest request ( BucketName , ObjectName ,
content );

    /* Upload the file . */
    auto outcome = client . PutObject ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " PutObject fail " <<
        ", code : " << outcome . error () . Code () <<
        ", message : " << outcome . error () . Message () <<
        ", requestId : " << outcome . error () . RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

Local file upload

Use the following code to upload a local file to OSS:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
```

```

/* Initialize the OSS account information . */
std :: string AccessKeyId = " yourAccess KeyId ";
std :: string AccessKeySecret = " yourAccess KeySecret ";
std :: string Endpoint = " yourEndpoint ";
std :: string BucketName = " yourBucket Name ";
std :: string ObjectName = " yourObject Name ";

/* Initialize network resources . */
InitializeSdk ();

ClientConfiguration conf ;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

/* Upload the file . */
auto outcome = client . PutObject ( BucketName , ObjectName , yourLocalFilename );

if ( ! outcome . isSuccess () ) {
    /* Handle exceptions . */
    std :: cout << " PutObject fail " <<
        ", code : " << outcome . error () . Code () <<
        ", message : " << outcome . error () . Message () <<
        ", requestId : " << outcome . error () . RequestId () << std :: endl ;
    ShutdownSdk ();
    return - 1 ;
}

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

5.8.3 Append upload

This topic describes how to append data from a local file.

Use the following code to append data from a local file:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    auto meta = ObjectMetadata ();
    meta . setContentType ( " text / plain " );
}

```

```

// If the object is appended for the first time
, the append position is 0, and the returned value
is the position for the next append. The position
for the next append is the length of the current
object. */
std::shared_ptr< std::iostream> content1 = std::
make_shared< std::stringstream>();
* content1 << " Thank you for using Aliyun Object
Storage Service !" ;
AppendObjectRequest request ( BucketName , ObjectName ,
content1 , meta );
request . setPosition ( 0L );

/* Start the first append . */
auto result = client . AppendObject ( request );

if (! result . isSuccess ()) {
/* Handle exceptions . */
std::cout << " AppendObject fail " <<
", code :" << result . error (). Code () <<
", message :" << result . error (). Message () <<
", requestId :" << result . error (). RequestId () << std
:: endl ;
ShutdownSdk ();
return - 1 ;
}

std::shared_ptr< std::iostream> content2 = std::
make_shared< std::stringstream>();
* content2 << " Thank you for using Aliyun Object
Storage Service !" ;
auto position = result . result (). Length ();
AppendObjectRequest appendObjectRequest ( BucketName ,
ObjectName , content2 );
appendObjectRequest . setPosition ( position );

/* Start the second append . */
auto outcome = client . AppendObject ( appendObject
Request );

if (! outcome . isSuccess ()) {
/* Handle exceptions . */
std::cout << " AppendObject fail " <<
", code :" << outcome . error (). Code () <<
", message :" << outcome . error (). Message () <<
", requestId :" << outcome . error (). RequestId () << std
:: endl ;
ShutdownSdk ();
return - 1 ;
}

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

If the object already exists, one of the following situations occurs:

- If the object is not appendable, `ObjectNotAppendable` is thrown.

- If the object is appendable, PutObject is used to upload the object, the object overwrites the existing object, and the object type becomes Normal Object. After the HeadObject operation is performed, OSS returns x-oss-object-type.

5.8.4 Resumable upload

You can use resumable upload to split a file into several parts and upload them separately. After you have uploaded all parts, you can combine these parts into a complete object. In this way, you can complete uploading the entire object.

For more information about resumable upload, see the "[Resumable upload](#)" section in OSS Developer Guide. For the complete code, see [GitHub](#).

You can use OSSClient. ResumableUploadObject to implement resumable upload. The following table describes the parameters contained in UploadObjectRequest.

Name	Description	Required	Default value	Configuration method
bucket	The name of the bucket.	Yes	None	Constructor
key	The name of the object.	Yes	None	Constructor
filePath	The path to the local file. The file located in this path is uploaded to OSS.	No	The path to the local file	Constructor
partSize	The size of each part. Valid values: [100 KB , 5 GB].	No	8 MB	setPartSize
threadNum	The number of parts you need to upload simultaneously .	No	3	Constructor or setThreadNum

Name	Description	Required	Default value	Configuration method
checkpointDir	The directory that records the upload results of each local part. This parameter must be configured to enable resumable upload. This file stores information about upload progress. If you fail to upload a part, the part upload continues based on the recorded progress. After the entire local file is uploaded, this file is deleted.	No	The same directory as that of DownloadFile	Constructor or setCheckpointDir

Use the following code for resumable upload:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    std :: string UploadFilePath = " yourUpload filePath ";
    std :: string CheckpointFilePath = " yourCheckpointFilePath ";

    /* Initialize network resources . */
    InitializeSdk ();
```

```

        ClientConfiguration conf ;
        OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

        /* Start resumable upload . */
        UploadObjectRequest request ( BucketName , ObjectName ,
        UploadFilePath , CheckpointFilePath );
        auto outcome = client . ResumableUploadObject ( request
        );

        if ( ! outcome . isSuccess () ) {
            /* Handle exceptions . */
            std :: cout << " ResumableUploadObject fail " <<
            " , code : " << outcome . error () . Code () <<
            " , message : " << outcome . error () . Message () <<
            " , requestId : " << outcome . error () . RequestId () << std
            :: endl ;
            ShutdownSdk ();
            return - 1 ;
        }

        /* Release network resources . */
        ShutdownSdk ();
        return 0 ;
    }
}

```

5.8.5 Multipart upload

This topic describes how to use multipart upload.

To enable multipart upload, perform the following steps:

1. Initialize a multipart upload event.

You can call `InitiateMultipartUpload` to return the globally unique upload ID created in OSS.

2. Start the multipart upload event.

Call `UploadPart` to upload multipart data.



Notice:

- For parts that share an upload ID, their part numbers identify their relative positions in the entire object. If you have uploaded a part and used its part number again to upload another part, the later part overwrites the former part.
- OSS places the MD5 value of part data in ETag and returns the MD5 value to the user.
- The SDK automatically configures Content-MD5. OSS calculates the MD5 value of uploaded data and compares it with the MD5 value calculated by SDK. If the two values vary, the `InvalidDigest` error is returned.

3. Complete the multipart upload event.

After you have uploaded all parts, call `CompleteMultipartUpload` to combine these parts into a complete object.

Use the following code to complete multipart upload:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int64_t getFileSize ( const std :: string & file )
{
    std :: fstream f ( file , std :: ios :: in | std :: ios ::
binary );
    f . seekg ( 0 , f . end );
    int64_t size = f . tellg ();
    f . close ();
    return size ;
}

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );
    InitiateMultipartUploadRequest initUploadRequest (
BucketName , ObjectName );

    /* Initiate the multipart upload event . */
    auto uploadIdResult = client . InitiateMultipartUploa
d ( initUploadRequest );
    auto uploadId = uploadIdResult . result () . UploadId ();
    std :: string fileToUpload = " yourLocalFilename ";
    int64_t partSize = 100 * 1024 ;
    PartList partETagList ;
    auto fileSize = getFileSize ( fileToUpload );
    int partCount = static_cast < int > ( fileSize / partSize
);
    /* Calculate the number of parts . */
    if ( fileSize % partSize != 0 ) {
        partCount ++;
    }

    /* Upload each part sequentially . */
    for ( int i = 1 ; i <= partCount ; i ++ ) {
        auto skipBytes = partSize * ( i - 1 );
        auto size = ( partSize < fileSize - skipBytes ) ?
partSize : ( fileSize - skipBytes );
        std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: fstream > ( fileToUpload , std :: ios :: in
| std :: ios :: binary );
```

```

        content -> seekg ( skipBytes , std :: ios :: beg );

        UploadPart Request uploadPart Request ( BucketName ,
ObjectName , content );
        uploadPart Request . setContent Length ( size );
        uploadPart Request . setUploadI d ( uploadId );
        uploadPart Request . setPartNum ber ( i );
        auto uploadPart Outcome = client . UploadPart (
uploadPart Request );
        if ( uploadPart Outcome . isSuccess () ) {
            Part part ( i , uploadPart Outcome . result (). ETag
());
            partETagLi st . push_back ( part );
        }
        else {
            std :: cout << " uploadPart fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () <<
std :: endl ;
        }

    }

    /* Complete the multipart upload event . */
    CompleteMu ltipartUpl oadRequest request ( BucketName ,
ObjectName );
    request . setUploadI d ( uploadId );
    request . setPartLis t ( partETagLi st );
    auto outcome = client . CompleteMu ltipartUpl oad ( request
);

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " CompleteMu ltipartUpl oad fail " <<
        ", code :" << outcome . error (). Code () <<
        ", message :" << outcome . error (). Message () <<
        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSd k ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSd k ();
    return 0 ;
}

```

**Note:**

Obtain the uploaded parts. ETag values of each part must be provided to call CompleteMultipartUpload.

You can use either of the follows method to obtain the ETag values:

- First, the response of a part upload contains the ETag value of the part. You can save the values for future use.

- You can also call the ListParts API to obtain the ETag value of the uploaded parts. In the example, the second method is used.

List uploaded parts

You can call ListParts to list all parts that are uploaded with the specified upload ID.

- List uploaded parts



Note:

ListParts can only list a maximum of 1,000 parts simultaneously by default. To list more than 1,000 uploaded parts, list them by page.

Use the following code to list all uploaded parts by page:

```
# include < alibabacloud_oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* List uploaded parts . A maximum of 1 , 000 parts can be listed by default . */
    ListPartsRequest listupload request ( BucketName , ObjectName );
    listupload request . setUploadId ( uploadId );
    do {
        auto listupload result = client . ListParts ( listupload request );
        if (! listupload result . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " ListParts fail " <<
                ", code : " << listupload result . error (). Code ()
                <<
                ", message : " << listupload result . error ().
                Message () <<
                ", requestId : " << listupload result . error ().
                RequestId () << std :: endl ;
            break ;
        }
        else {
            for ( const auto & part : listupload result .
                result (). PartList ()) {
                std :: cout << " part "<<
                    ", name : " << part . PartNumber () <<
```

```

        ", size :" << part . Size () <<
        ", etag :" << part . ETag () <<
        ", lastmodify   time :" << part . LastModifi ed ()
<< std :: endl ;
    }
}
listupload request . setPartNum berMarker ( listupload
result . result (). NextPartNu mberMarker ());
} while ( listupload result . result (). IsTruncate d ());

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

- List all uploaded parts by page

Use the following code to specify the maximum number of parts displayed on each page:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* List all uploaded parts by page . */
    /* Set the maximum number of parts that can be
    uploaded on each page . */
    ListPartsRequest listupload request ( BucketName ,
    ObjectName );
    listupload request . setMaxParts ( 50 );
    listupload request . setUploadId ( uploadId );
    do {
        listupload result = client . ListParts ( listupload
request );
        if (! listUploadResult . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " ListParts fail " <<
            ", code :" << listupload result . error (). Code ()
<<
            ", message :" << listupload result . error ().
Message () <<
            ", requestId :" << listupload result . error ().
RequestId () << std :: endl ;
            break ;
        }
    }
    else {

```

```

        for ( const auto & part : listupload result .
result (). PartList () ) {
            std :: cout << " part "<<
            ", name :" << part . PartNumber () <<
            ", size :" << part . Size () <<
            ", etag :" << part . ETag () <<
            ", lastmodify   time :" << part . LastModifi ed ()
<< std :: endl ;
        }
        listupload request . setPartNum berMarker ( listupload
result . result (). NextPartNu mberMarker ());
    } while ( listupload result . result (). IsTruncate d ());

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

List multipart upload events

You can call `ListMultipartUploads` to list all ongoing multipart upload events that have been initiated but not completed or have been cancelled.

- List multipart upload events



Note:

`ListMultipartUploads` can be used to list only a maximum of 1,000 multipart upload events simultaneously by default. If the number of events is greater than 1,000, use the following pagination to list all upload events.

Use the following code to list all multipart upload events:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoi nt ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );

    /* List uploaded events . A maximum of 1 , 000
multipart upload events can be listed . */
    ListMultipartUploads Request listmultiu ploadrequest (
BucketName );
    do {

```

```

        auto listresult = client.ListMultipartUploads (
listmultiuploadrequest);
        if (! listresult.IsSuccess()) {
            /* Handle exceptions. */
            std::cout << "ListMultipartUploads fail " <<
                ", code:" << listresult.error().Code() <<
                ", message:" << listresult.error().Message() <<
                ", requestId:" << listresult.error().RequestId()
<< std::endl;
            break;
        }
        else {
            for (const auto & part : listresult.result
(). MultipartUploadList()) {
                std::cout << " part "<<
                    ", name:" << part.Key <<
                    ", uploadid:" << part.UploadId <<
                    ", initiated time:" << part.Initiated << std
::endl;
            }
            listmultiuploadrequest.SetKeyMarker ( listresult
.result(). NextKeyMarker());
            listmultiuploadrequest.SetUploadIdMarker (
listresult.result(). NextUploadIdMarker());
            while ( listresult.result(). IsTruncated());

            /* Release network resources. */
            ShutdownSdk();
            return 0;
        }
    }
}

```

- List all upload events by page

Use the following code to list all upload events by page:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main (void)
{
    /* Initialize the OSS account information. */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";

    /* Initialize network resources. */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client (Endpoint, AccessKeyId, AccessKeySecret, conf);

    /* List all upload events by page. */
    /* Set the maximum number of upload events that
    can be displayed on each page. */
    ListMultipartUploadsRequest listmultiuploadrequest (
BucketName);
    listmultiuploadrequest.SetMaxUploads (50);
    do {
        listresult = client.ListMultipartUploads (
listmultiuploadrequest);
    }
}

```



```

        if (! listresult . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " ListMulti partUploads fail " <<
                ", code :" << listresult . error (). Code () <<
                ", message :" << listresult . error (). Message () <<
                ", requestId :" << listresult . error (). RequestId ()
            << std :: endl ;
            break ;
        }
        else {
            for ( const auto & part : listresult . result
                (). MultipartUploadList ()) {
                std :: cout << " part "<<
                    ", name :" << part . Key <<
                    ", uploadid :" << part . UploadId <<
                    ", initiated time :" << part . Initiated << std
                :: endl ;
            }
            listmultiuploadrequest . setKeyMarker ( listresult .
                result (). NextKeyMarker ());
            listmultiuploadrequest . setUploadIdMarker (
                listresult . result (). NextUploadIdMarker ());
            while ( listresult . result (). IsTruncated ());

            /* Release network resources . */
            ShutdownSdk ();
            return 0 ;
        }
    }

```

Cancel a multipart upload event

Use the following code to cancel a multipart upload event:

```

#include < alibabacloud_oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccessKeyId ";
    std :: string AccessKeySecret = " yourAccessKeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucketName ";
    std :: string ObjectName = " yourObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    InitiateMultipartUploadRequest initUploadRequest (
        BucketName , ObjectName );

    /* Initiate the multipart upload event . */
    auto uploadIdResult = client . InitiateMultipartUpload
        ( initUploadRequest );
    auto uploadId = uploadIdResult . result (). UploadId ();

    /* Cancel the multipart upload task . */
}

```

```

    AbortMulti partUpload Request abortUploa dRequest (
BucketName , key , uploadId );
    auto abortUploa dIdResult = client . AbortMulti partUpload
( abortUploa dRequest );

    if (! abortUploa dIdResult . isSuccess ()) {
        /* Handle exceptions . */
        std :: cout << " AbortMulti partUpload fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSd k ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSd k ();
    return 0 ;
}

```

**Note:**

After a multipart upload event is cancelled, you cannot use the same upload ID to perform any operations and the parts uploaded with that upload ID are also deleted.

5.8.6 Interrupt an upload task

You can interrupt an upload task at any time when it is being uploaded.

Use the following code to interrupt an upload task:

```

# include < alibabacloud / oss / OssClient . h >
# include < fstream >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    std :: vector < PutObjectOutcomeCallable > Callables ;
    for ( int i = 0 ; i < 4 ; i ++ ) {
        std :: shared_ptr < std :: iostream > content = std ::
make_shared < std :: fstream > ( " yourLocalFilename ", std :: ios
:: in | std :: ios :: binary );
        PutObjectRequest request ( BucketName , ObjectName ,
content );
        auto outcomeCallable = client . PutObjectCallable (
request );
    }
}

```

```

        Callables . emplace_back ( std :: move ( outcomeCallable
));
    }
    std :: this_thread :: sleep_for ( std :: chrono :: milliseconds ( 1000 ));

    /* Interrupt the upload task . */
    client . DisableRequest ();

    for ( size_t i = 0 ; i < Callables . size (); i ++ ) {
        auto outcome = Callables [ i ]. get ();
        if ( outcome . error (). Code () == " ClientError :
100002 " ||
            outcome . error (). Code () == " ClientError : 200042 " ) {
            std :: cout << " disable putobject success " << std
:: endl ;
        }
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

**Note:**

An interrupted upload and a cancelled multipart upload have the following differences:

- When you call the upload API to remotely upload an object to OSS, you can interrupt the upload task. In other words, you stopped the upload task. After you cancel the upload task, OSS no longer processes other multipart upload tasks that use the same upload IP as that of the cancelled task.
- If you interrupt the upload task, you can still call the upload ID to continue uploading the content although the current upload task is interrupted. If you cancel a multipart upload task, the upload ID is unavailable for future uploads.
- Interrupted upload applies to all APIs. However, cancelled multipart upload applies only to multipart upload.

5.8.7 Use a progress bar

Progress bars are used to indicate the progress of an upload or download.

The following code uses PutObject as an example to show you how to view progress information:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

void ProgressCallback ( size_t increment , int64_t
transferred , int64_t total , void * userData )
{

```

```

        std::cout << " ProgressCa llback [" << userData << "] => "
        <<
        increment <<" ," << transfered << "," << total << std::
endl ;
    }

    int main ( void )
    {
        /* Initialize the OSS account informatio n . */
        std::string AccessKeyId = " yourAccess KeyId ";
        std::string AccessKeySecret = " yourAccess KeySecret ";
        std::string Endpoint = " yourEndpoi nt ";
        std::string BucketName = " yourBucket Name ";
        std::string ObjectName = " yourObject Name ";

        /* Initialize network resources . */
        InitializeSdk ();

        ClientConfiguration conf ;
        OssClient client ( Endpoint , AccessKeyId , AccessKeyS
ecret , conf );

        std::shared_ptr < std::iostream > content = std::
make_shared < std::fstream >(" yourLocalF ilename ", std:: ios
:: in | std:: ios:: binary );
        PutObjectRequest request ( BucketName , ObjectName ,
content );

        TransferProgress progressCa llback = { ProgressCa llback
, nullptr };
        request.setTransferProgress ( progressCa llback );

        /* Upload the object . */
        auto outcome = client . PutObject ( request );

        if (! outcome . isSuccess ()) {
            /* Handle exceptions . */
            std::cout << " PutObject fail " <<
            ", code:" << outcome . error (). Code () <<
            ", message:" << outcome . error (). Message () <<
            ", requestId:" << outcome . error (). RequestId () << std
:: endl ;
            ShutdownSdk ();
            return - 1 ;
        }

        /* Release network resources . */
        ShutdownSdk ();
        return 0 ;
    }

```

5.8.8 Upload callback

This topic describes how to use upload callback.

Use the following code for upload callback:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud::OSS ;

int main ( void )

```

```

{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = "yourAccess KeyId ";
    std :: string AccessKeySecret = "yourAccess KeySecret ";
    std :: string Endpoint = "yourEndpoint ";
    std :: string BucketName = "yourBucket Name ";
    std :: string ObjectName = "yourObject Name ";
    /* Specify the IP address of the server you want
    to send the callback request to , such as http ://
    oss - demo . aliyunc s . com : 23450 . */
    std :: string ServerName = "yourServer Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
    std :: shared_ptr < std :: istream > content = std ::
make_shared < std :: stringstream > ();
    * content << " Thank you for using Aliyun Object
Storage Service !" ;

    /* Set the upload callback parameters . */
    std :: string callbackBody = " bucket = ${ bucket } & object =
${ object } & etag = ${ etag } & size = ${ size } & mimeType = ${ mimeType
} & my_var1 = ${ x : var1 }";
    ObjectCallbackBuilder builder ( ServerName , callbackBo
dy , "", ObjectCallbackBuilder :: Type :: URL );
    std :: string value = builder . build ();
    ObjectCallbackBuilder varBuilder ;
    varBuilder . addCallbackVariable ( " x : var1 ", " value1 " );
    std :: string varValue = varBuilder . build ();
    PutObjectRequest request ( BucketName , ObjectName ,
content );
    request . Metadata (). addHeader ( " x - oss - callback ", value
);
    request . Metadata (). addHeader ( " x - oss - callback - var ",
varValue );
    auto outcome = client . PutObject ( request );

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

For more information about upload callback, see the ["Upload callback"](#) section in OSS Developer Guide. For more information about debugging methods and troubleshooting, see [Upload callback errors and troubleshooting](#). For more information about the implementation of upload callback, see the ["Callback"](#) section in API reference.

5.9 Download objects

5.9.1 Overview

The OSS C++ SDK provides the following object download methods:

- [Download to a local file](#)
- [Download to local memory](#)
- [Range download](#)
- [Resumable download](#)

You can view the download progress, as show in [Use a progress bar](#).

5.9.2 Download to a local file

This topic describes how to download an object from OSS to a local file.

Use the following code to download an object to a local file:

```
# include < alibabacloud / oss / OssClient . h >
# include < memory >
# include < fstream >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    /* Set the name of the local file you want to
download to . */
    std :: string FileNameToSave = " yourFileName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Download the object to the local file . */
    GetObjectRequest request ( BucketName , ObjectName );
    request . setResponseStreamFactory ([=]() { return std ::
make_shared < std :: fstream > ( FileNameToSave , std :: ios_base
:: out | std :: ios_base :: in | std :: ios_base :: trunc | std
:: ios_base :: binary ); });

    auto outcome = client . GetObject ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " GetObjectToFile success " << outcome .
result (). Metadata (). ContentLength () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GetObjectToFile fail " <<
        ", code : " << outcome . error (). Code () <<
```

```

        ", message :" << outcome . error (). Message () <<
        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

5.9.3 Download to local memory

This topic describes how to download an object to local memory.

Run the following code to download an object to local memory:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Download the object to local memory . */
    GetObjectRequest request ( BucketName , ObjectName );
    auto outcome = client . GetObject ( request );
    if ( outcome . isSuccess () ) {
        std :: cout << " getObjectToBuffer " << " success ,
Content - Length :" << outcome . result (). Metadata (). ContentLength () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " getObjectToBuffer fail " <<
        ", code :" << outcome . error (). Code () <<
        ", message :" << outcome . error (). Message () <<
        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

```
}
```

5.9.4 Range download

If you only need part of the data in an object, you can use range download to download the specified content.

Use the following code for range download:

```
# include < alibabacloud_oss / OssClient . h >
using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std::string AccessKeyId = " yourAccess KeyId ";
    std::string AccessKeySecret = " yourAccess KeySecret ";
    std::string Endpoint = " yourEndpoint ";
    std::string BucketName = " yourBucket Name ";
    std::string ObjectName = " yourObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf;
    OssClient client ( Endpoint, AccessKeyId, AccessKeySecret, conf );

    /* Obtain the object . */
    GetObjectRequest request ( BucketName, ObjectName );
    /* Set the download range . */
    request . setRange ( 0, 1 );
    auto outcome = client . GetObject ( request );

    if (! outcome . isSuccess ()) {
        /* Handle exceptions . */
        std::cout << " getObject fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std
:: endl;
        ShutdownSdk ();
        return - 1;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0;
}
```



```
}
```

5.9.5 Resumable download

Large-sized objects may fail to be downloaded because the network is unstable or the program exits abnormally. Objects may still fail to be downloaded after multiple attempts. To address this problem, OSS provides resumable download.

To enable resumable download, you must split the object you want to download into multiple parts and download them separately. After you download all parts, these parts are combined into a complete object.

You can use `OSSClient.ResumableDownloadObject` to achieve resumable download. The following table describes the parameters contained in `DownloadObjectRequest`.

Name	Description	Required	Default value	Configuration method
bucket	The name of the bucket.	Yes	None	Constructor
key	The name of the object.	Yes	None	Constructor
filePath	The path to the local file. The file located in this path is downloaded to OSS.	No	Name of the object	Constructor
partSize	The size of each part. Valid values: [100 KB, 5 GB].	No	8 MB	setPartSize
threadNum	The number of parts you need to download simultaneously.	No	3	Constructor or setThreadNum

Name	Description	Required	Default value	Configuration method
checkpointDir	The file to record the results of multipart download. This parameter must be configured to enable resumable upload. This file stores information about download progress. If you fail to download a part, the part download continues based on the recorded progress. After the object is downloaded to your local device, the checkpoint file is deleted.	No	The same directory as that of DownloadFile	Constructor or setCheckpointDir

Use the following code for resumable download:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    std :: string DownloadFilePath = " yourDownloadFilePath ";
```

```

        std::string Checkpoint FilePath = "yourCheckpointFilePath";

        /* Initialize network resources. */
        InitializeSdk();

        ClientConfiguration conf;
        OssClient client(Endpoint, AccessKeyId, AccessKeySecret, conf);

        /* Start resumable download. */
        DownloadObjectRequest request(BucketName, ObjectName,
        DownloadFilePath, Checkpoint FilePath);
        auto outcome = client.ResumableDownloadObject(request);

        if (!outcome.IsSuccess()) {
            /* Handle exceptions. */
            std::cout << "ResumableDownloadObject fail " <<
            ", code:" << outcome.error().Code() <<
            ", message:" << outcome.error().Message() <<
            ", requestId:" << outcome.error().RequestId() << std
            ::endl;
            ShutdownSdk();
            return -1;
        }

        /* Release network resources. */
        ShutdownSdk();
        return 0;
    }
}

```

5.9.6 Use a progress bar

Progress bars are used to indicate the progress of an upload or download.

For the complete code of download progress bars, see [GitHub](#).

Use the following code to view the progress information:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

void ProgressCallback(size_t increment, int64_t
transferred, int64_t total, void * userData)
{
    std::cout << "ProgressCallback [" << userData << "] => "
    <<
        increment << ", " << transferred << ", " << total
    << std::endl;
}

int main(void)
{
    /* Initialize the OSS account information. */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";
    std::string ObjectName = "yourObjectName";

    /* Initialize network resources. */
}

```

```

Initialize Sdk ();

ClientConf figuration conf ;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

/* Obtain the object and its download progress information . */
GetObjectRequest request ( BucketName , ObjectName );
TransferProgress progressCallback = { ProgressCallback , nullptr };
request . setTransferProgress ( progressCallback );

auto outcome = client . GetObject ( request );

if ( ! outcome . isSuccess () ) {
    /* Handle exceptions . */
    std :: cout << " getObject fail " <<
        ", code :" << outcome . error () . Code () <<
        ", message :" << outcome . error () . Message () <<
        ", requestId :" << outcome . error () . RequestId () << std
        :: endl ;
    ShutdownSdk ();
    return - 1 ;
}

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

5.10 Manage objects

5.10.1 Overview

You can use APIs to perform the following operations on objects in a bucket:

- [Manage the ACL of an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Copy an object](#)
- [Delete objects](#)
- [Determine whether the specified object exists](#)
- [Restore an archive object](#)
- [Manage a symbolic link](#)

5.10.2 Manage the ACL of an object

This topic describes how to manage the access control list (ACL) of an object.

The following table describes the permissions included in the ACL of an object.

Permission	Description	Value
Inherit Bucket	The ACL of an object is the same as that of its bucket.	CannedAccessControlList::Default
Private	Only the owner or authorized users have read and write permissions on the object.	CannedAccessControlList::Private
Public Read	Only the owner or authorized users of an object have read and write permissions on the object . Other users (including anonymous users) have only read permissions on the object. Use caution when using this operation.	CannedAccessControlList::PublicRead
Public	All users of an object have read and write permissions on the object . Use caution when using this operation.	CannedAccessControlList::PublicReadWrite

The ACL privileges of objects take precedence over those of buckets. For example, the ACL of a bucket is Private, while the object ACL is Public. (All users of an object have read and write permissions on the object.) If an object does not have an ACL, the ACL of its bucket is used by default.

Configure an ACL of an object

Use the following code to configure an ACL for a specified object:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
```

```

    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Set the ACL of the object . */
    SetObjectAclRequest request ( BucketName , ObjectName );
    request . setAcl ( CannedAccessControlList :: Private );
    auto outcome = client . SetObjectAcl ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " SetObjectAcl fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Obtain the ACL of an object

Use the following code to obtain an ACL to access a specified object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccessKeyId ";
    std :: string AccessKeySecret = " yourAccessKeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucketName ";
    std :: string ObjectName = " yourObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Obtain the ACL of the object . */
    GetObjectAclRequest request ( BucketName , ObjectName );
    auto outcome = client . GetObjectAcl ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " GetObjectAcl fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
    else {

```

```

        std::cout << "GetObjectAcl success, Acl:" <<
outcome.result().Acl() << std::endl;
    }

    /* Release network resources. */
    ShutdownSdk();
    return 0;
}

```

5.10.3 Manage Object Meta

Object Meta contains HTTP headers and User Meta. For more information, see [Object Meta](#) in OSS Developer Guide.

Configure Object Meta

Use the following code to configure Object Meta:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main(void)
{
    /* Initialize the OSS account information. */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";
    std::string ObjectName = "yourObjectName";

    /* Initialize network resources. */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client(Endpoint, AccessKeyId, AccessKeySecret, conf);

    /* Set the HTTP header. */
    auto meta = ObjectMetaData();
    meta.setContentType("text/plain");
    meta.setCacheControl("max-age=3");
    /* Set User Meta. */
    meta.setUserMetadata()["meta"] = "meta-value";

    std::shared_ptr<std::iostream> content = std::make_shared<std::stringstream>();
    *content << "Thank you for using Aliyun Object Storage Service!";
    client.PutObject(BucketName, ObjectName, content, meta);

    if (!outcome.isSuccess()) {
        /* Handle exceptions. */
        std::cout << "PutObject fail" <<
            ", code:" << outcome.error().Code() <<
            ", message:" << outcome.error().Message() <<
            ", requestId:" << outcome.error().RequestId() << std::endl;
        ShutdownSdk();
    }
}

```

```

        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Obtain Object Meta

The following table lists the methods you can use to obtain Object Meta.

Method	Description	Benefit
GetObjectMeta	Obtains some Object Meta such as the ETag, Size, and LastModified (the latest time an object was modified) values of the object.	More lightweight and faster
HeadObject	Obtains all Object Meta.	None

Use the following code to obtain Object Meta:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Obtain some Object Meta . */
    auto outcome = client . GetObjectMeta ( BucketName ,
    ObjectName );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " GetObjectMeta fail " <<
        ", code : " << outcome . error () . Code () <<
        ", message : " << outcome . error () . Message () <<
        ", requestId : " << outcome . error () . RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
    else {

```



```

        auto metadata = outcome.result();
        std::cout << " get metadata success , ETag : " <<
        metadata.ETag() << "; LastModified : "
        << metadata.LastModified() << "; Size : " << metadata.
        ContentLength() << std::endl;
    }

    /* Obtain all Object Meta . */
    outcome = client.HeadObject( BucketName , ObjectName );

    if (! outcome.isSuccess()) {
        /* Handle exceptions . */
        std::cout << " HeadObject fail " <<
        ", code : " << outcome.error().Code() <<
        ", message : " << outcome.error().Message() <<
        ", requestId : " << outcome.error().RequestId() << std
        ::endl;
        ShutdownSdk();
        return -1;
    }
    else {
        auto headMeta = outcome.result();
        std::cout << " headMeta success , ContentType : "
        << headMeta.ContentType() << "; ContentLength : " <<
        headMeta.ContentLength()
        << "; CacheControl : " << headMeta.CacheControl() <<
        std::endl;
    }

    /* Release network resources . */
    ShutdownSdk();
    return 0;
}

```

**Note:**

For more information about HTTP headers, see [RFC2616](#).

5.10.4 List objects

This topic describes how to list the objects stored in a bucket.

Objects are listed alphabetically. You can use `OssClient.ListObjects` to list the objects in a bucket. `ListObjects` supports three types of parameter formats:

- `ListObjectOutcome ListObjects(const std::string& bucket) const`: lists objects in a bucket. A maximum of 100 objects can be listed.
- `ListObjectOutcome ListObjects(const std::string& bucket, const std::string& prefix) const`: lists objects with the specified prefix in a bucket. A maximum of 100 objects can be listed.
- `ListObjectOutcome ListObjects(const ListObjectsRequest& request) const`: provides multiple filtering functions to flexibly query objects.

The following table describes parameters for object listing.

Name	Description
delimiter	The delimiter used to group objects . CommonPrefixes specify a set of substrings that start with a specified prefix and end with the first specified delimiter.
prefix	The prefix that must be included in the returned objects.
maxKeys	The maximum number of objects that can be listed. The default value is 100. The maximum value is 1,000.
marker	The initial object in the list.

Simple list

Use the following code to list objects in the specified bucket. A maximum of 100 objects can be listed by default.

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* List the objects . */
    ListObjectsRequest request ( BucketName );
    auto outcome = client . ListObjects ( request );

    if (! outcome . isSuccess ()) {
        /* Handle exceptions . */
        std :: cout << " ListObjects fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
    else {
        for ( const auto & object : outcome . result ().
            ObjectSummaries () ) {
```

```

        std::cout << " object "<<
        ", name:" << object.Key() <<
        ", size:" << object.Size() <<
        ", lastmodify time:" << object.LastModified() <<
std::endl;
    }

    /* Release network resources. */
    ShutdownSdk();
    return 0;
}

```

List a specified number of objects

Use the following code to list a specified number of objects:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main(void)
{
    /* Initialize the OSS account information. */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";

    /* Initialize network resources. */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client(Endpoint, AccessKeyId, AccessKeySecret, conf);

    /* List the objects. */
    ListObjectsRequest request(BucketName);
    /* Configure the maximum number of objects you want to list. */
    request.setMaxKeys(200);
    auto outcome = client.ListObjects(request);

    if (!outcome.IsSuccess()) {
        /* Handle exceptions. */
        std::cout << "ListObjects fail " <<
        ", code:" << outcome.error().Code() <<
        ", message:" << outcome.error().Message() <<
        ", requestId:" << outcome.error().RequestId() << std::endl;
        ShutdownSdk();
        return -1;
    }
    else {
        for (const auto & object : outcome.result().ObjectSummaries()) {
            std::cout << " object "<<
            ", name:" << object.Key() <<
            ", size:" << object.Size() <<
            ", lastmodify time:" << object.LastModified() <<
            std::endl;
        }
    }
}

```

```

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

List the objects and subfolders of a folder

Use the following code to list the objects and subfolders of a folder:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string keyPrefix = " yourkeyPrefix ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    std :: string nextMarker = "";
    do {
        /* List the objects . */
        ListObjectsRequest request ( BucketName );
        /* Set forward slashes (/) as the delimiter for folders . */
        request . setDelimiter ("/");
        request . setPrefix ( keyPrefix );
        request . setMarker ( nextMarker );
        auto outcome = client . ListObjects ( request );

        if (! outcome . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " ListObjects fail " <<
                ", code : " << outcome . error (). Code () <<
                ", message : " << outcome . error (). Message () <<
                ", requestId : " << outcome . error (). RequestId () <<
            std :: endl ;
            break ;
        }
        for ( const auto & object : outcome . result ().
            ObjectSummaries ()) {
            std :: cout << " object "<<
                ", name : " << object . Key () <<
                ", size : " << object . Size () <<
                ", lastmodify time : " << object . LastModified () <<
            std :: endl ;
        }
        for ( const auto & commonPrefix : outcome . result ().
            CommonPrefixes ()) {
            std :: cout << " commonPrefix " << ", name : " <<
            commonPrefix << std :: endl ;
        }
    } while ( true );
}

```

```

        nextMarker = outcome . result (). NextMarker ();
    } while ( outcome . result (). IsTruncate d ());

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

List objects by prefix

Use the following code to list objects with the specified prefix:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string keyPrefix = " yourkeyPre fix ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    std :: string nextMarker = "";
    do {
        /* List the objects . */
        ListObjectRequest request ( BucketName );
        /* Specify the prefix . */
        request . setPrefix ( keyPrefix );
        request . setMarker ( nextMarker );
        auto outcome = client . ListObjects ( request );

        if (! outcome . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " ListObjects fail " <<
                ", code : " << outcome . error (). Code () <<
                ", message : " << outcome . error (). Message () <<
                ", requestId : " << outcome . error (). RequestId () <<
            std :: endl ;
            break ;
        }
        for ( const auto & object : outcome . result ().
            ObjectSummaries ()) {
            std :: cout << " object "<<
                ", name : " << object . Key () <<
                ", size : " << object . Size () <<
                ", lastmodify time : " << object . LastModified () <<
            std :: endl ;
        }
        nextMarker = outcome . result (). NextMarker ();
    } while ( outcome . result (). IsTruncated ());

    /* Release network resources . */
    ShutdownSdk ();
}

```

```

        return 0 ;
    }

```

List objects by marker

The marker parameter indicates the name of an object from which to start listing objects. Use the following code to specify an object from which to start listing objects:

```

#include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string YourMarker = " yourMarker ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    do {
        /* List the objects . */
        ListObjectsRequest request ( BucketName );
        /* List objects after the object specified by
the marker parameter . */
        request . setMarker ( YourMarker );
        auto outcome = client . ListObjects ( request );

        if ( ! outcome . isSuccess () ) {
            /* Handle exceptions . */
            std :: cout << " ListObjects fail " <<
                ", code : " << outcome . error () . Code () <<
                ", message : " << outcome . error () . Message () <<
                ", requestId : " << outcome . error () . RequestId () <<
            std :: endl ;
            break ;
        }
        YourMarker = outcome . result () . NextMarker ();
        for ( const auto & object : outcome . result () .
ObjectSummaries () ) {
            std :: cout << " object " <<
                ", name : " << object . Key () <<
                ", size : " << object . Size () <<
                ", lastmodify time : " << object . LastModified () <<
            std :: endl ;
        }
    } while ( outcome . result () . IsTruncated () );

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

```
}
```

Encode the name of an object

You must encode the name of an object if the name contains any of the following special characters. Only URL encoding is supported in OSS.

- Single quotation marks (')
- Double quotations marks (")
- Ampersands (&)
- Angle brackets (<>)
- Chinese characters

Use the following code to encode the names of specified objects:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    std :: string nextMarker = "";
    do {
        /* List the objects that need to be encoded . */
        ListObjectsRequest request ( BucketName );
        /* Encode the names of specified objects . */
        request . setEncodingType ( " url " );
        request . setMarker ( nextMarker );
        auto outcome = client . ListObjects ( request );

        if ( ! outcome . isSuccess () )
            /* Handle exceptions . */
            std :: cout << " ListObjects fail " <<
                ", code : " << outcome . error () . Code () <<
                ", message : " << outcome . error () . Message () <<
                ", requestId : " << outcome . error () . RequestId () <<
            std :: endl ;
        break ;
    }
    for ( const auto & object : outcome . result () .
ObjectSummaries () ) {
        std :: cout << " object " <<
            ", name : " << object . Key () <<
            ", size : " << object . Size () <<
```

```

        ", lastmodify   time : " << object . LastModifi   ed () <<
std :: endl ;
    }
    nextMarker = outcome . result (). NextMarker ();
} while ( outcome . result (). IsTruncate   d ());

/* Release   network   resources . */
ShutdownSdk ();
return   0 ;
}

```

5.10.5 Delete objects

This topic describes how to delete objects.



Warning:

Use caution when deleting objects because deleted objects cannot be recovered.

Delete an object

Use the following code to delete an object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize   the   OSS   account   informatio   n . */
    std :: string   AccessKeyId = " yourAccess   KeyId ";
    std :: string   AccessKeySecret = " yourAccess   KeySecret ";
    std :: string   Endpoint = " yourEndpoi   nt ";
    std :: string   BucketName = " yourBucket   Name ";
    std :: string   ObjectName = " yourObject   Name ";

    /* Initialize   network   resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    DeleteObjectRequest request ( BucketName , ObjectName );

    /* Delete   the   object . */
    auto outcome = client . DeleteObject ( request );

    if (! outcome . isSuccess ()) {
        /* Handle   exceptions . */
        std :: cout << " DeleteObject   fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release   network   resources . */
    ShutdownSdk ();
}

```



```
    return 0 ;
}
```

Delete multiple objects

Use the following code to delete multiple objects:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information .*/
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources .*/
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    DeleteObjectsRequest request ( BucketName );
    /* Add the names of the objects you want to delete .*/
    request . addKey ( ObjectName );

    /* Delete the objects .*/
    auto outcome = client . DeleteObjects ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions .*/
        std :: cout << " DeleteObjects fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources .*/
    ShutdownSdk ();
    return 0 ;
}
```

5.10.6 Copy an object

OSS allows you to copy objects of any sizes.

Copy a small object

You can use `OssClient::CopyObject` to copy an object smaller than 1 GB from a bucket (source bucket) to another bucket (destination bucket) within the same region. The configuration method is as follows:

Configuration method	Description
<code>CopyObjectOutcome OssClient::CopyObject(const CopyObjectRequest & request)</code>	Allows you to specify destination Object Meta and copy conditions. If the URLs of the source object and destination object for the COPY operation are the same, the source Object Meta is directly replaced.

**Note:**

Object copy rules are as follows:

- You must have read and write permissions on the source object.
- You cannot copy the object from a region to another region. For example, you cannot copy an object in a bucket from China (Hangzhou) to China (Qingdao).
- The object size is limited to a maximum of 1 GB.

Use the following code to copy a small object:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string SourceBucketName = " yourSource BucketName ";
    std :: string CopyBucketName = " yourCopyBucketName ";
    std :: string SourceObjectName = " yourSource ObjectName ";
    std :: string CopyObjectName = " yourCopyObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    CopyObjectRequest request ( CopyBucketName , CopyObjectName );
    request . setCopySource ( SourceBucketName , SourceObjectName );

    /* Copy the object . */
    auto outcome = client . CopyObject ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " CopyObject fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
            :: endl ;
        ShutdownSdk ();
    }
}
```

```

        return -1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Copy a large object

To copy an object larger than 1 GB, you must use `UploadPartCopy`. In other words, you must split the object into multiple parts and copy each of them sequentially. To enable `UploadPartCopy`, perform the following steps: Use `ossClient.initiateMultipartUpload` to initiate an `UploadPartCopy` task. Use `ossClient.uploadPartCopy` to perform the multipart copy task. Aside from the last part, all parts must be larger than 100 KB. Use `$ossClient->completeMultipartUpload` to submit the multipart copy task.



Note:

Large objects cannot be copied from one region to another region. For example, you cannot copy an object in a bucket from China (Hangzhou) to China (Qingdao).

Use the following code to copy a large object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string SourceBucketName = " yourSource BucketName ";
    std :: string CopyBucketName = " yourCopyBucketName ";
    std :: string SourceObjectName = " yourSource ObjectName ";
    std :: string CopyObjectName = " yourCopyObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    auto getObjectMetaReq = GetObjectMetaRequest (
        SourceBucketName , SourceObjectName );
    auto getObjectMetaResult = GetObjectMeta ( getObjectMetaReq );
    if ( ! getObjectMetaResult . isSuccess () ) {
        std :: cout << " GetObjectMeta fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<

```

```

        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
    /* Obtain the size of the object you want to
copy . */
    auto objectSize = getObjectMetadataResult . result ().
ContentLength ();

    /* Copy the large object . */
    InitiateMultipartUploadRequest initUploadRequest (
CopyBucketName , CopyObjectName );

    /* Initiate an UploadPart Copy task . */
    auto uploadIdResult = client . InitiateMultipartUpload
( initUploadRequest );
    auto uploadId = uploadIdResult . result (). UploadId ();
    int64_t partSize = 100 * 1024 ;
    PartList partETagList ;
    int partCount = static_cast < int >( objectSize /
partSize );
    /* Calculate the number of parts . */
    if ( objectSize % partSize != 0 ) {
        partCount ++;
    }

    /* Copy each part sequentially . */
    for ( int i = 1 ; i <= partCount ; i ++ ) {
        auto skipBytes = partSize * ( i - 1 );
        auto size = ( partSize < objectSize - skipBytes ) ?
partSize : ( objectSize - skipBytes );
        auto uploadPartCopyReq = UploadPartCopyRequest
( CopyBucketName , CopyObjectName , SourceBucketName ,
SourceObjectName , uploadId , i + 1 );
        uploadPartCopyReq . setCopySourceRange ( skipBytes ,
skipBytes + size - 1 );
        auto uploadPartOutcome = client . UploadPartCopy (
uploadPartCopyReq );
        if ( uploadPartOutcome . isSuccess () ) {
            Part part ( i , uploadPartOutcome . result (). ETag
());
            partETagList . push_back ( part );
        }
        else {
            std :: cout << " UploadPart Copy fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () <<
std :: endl ;
        }
    }

    /* Complete the multipart copy task . */
    CompleteMultipartUploadRequest request ( CopyBucketName
, CopyObjectName , partETagList , uploadId );
    auto outcome = client . CompleteMultipartUpload ( request
);

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " CompleteMultipartUpload fail " <<
        ", code :" << outcome . error (). Code () <<

```

```

        ", message :" << outcome . error () . Message () <<
        ", requestId :" << outcome . error () . RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

5.10.7 Determine whether the specified object exists

This topic describes how to determine whether an object exists.

Use the following code to determine whether an object exists:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Determine whether the object exists . */
    auto outcome = client . DoesObjectExist ( BucketName , ObjectName );

    if ( outcome ) {
        std :: cout << " The Object exists !" << std :: endl ;
    }
    else {
        std :: cout << " The Object does not exist !" << std
:: endl ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

```
}
```

5.10.8 Restore an archive object

This topic describes how to restore an archive object.

You must restore an archive object before you can read it. Do not call `RestoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

- The archive object is initially in the frozen state.
- After you submit it for restoration, the server restores the object. The object is in the restoring state.



Note:

Typically, it takes about one minute to restore an archive object.

- You can read the object after it is restored. The restored state of the object lasts one day by default. You can prolong this period to a maximum of seven days. After this period expires, the object returns to the frozen state.

Use the following code to restore an archive object:

```
# include < alibabacloud / oss / OssClient . h >
# include < thread >
# include < chrono >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Restore the archive object . */
    auto outcome = client . RestoreObject ( BucketName ,
    ObjectName );

    /* Non - archive objects cannot be restored . */
    if (! outcome . isSuccess ()) {
        /* Handle exceptions . */
        std :: cout << " RestoreObject fail " <<
        ", code : " << outcome . error (). Code () <<
```

```

        ", message :" << outcome . error (). Message () <<
        ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSd k ();
        return - 1 ;
    }

    std :: string  onGoingRes  tore (" ongoing - request =" false
\\");

    int  maxWaitTim  eInSeconds = 600 ;
    while ( maxWaitTim  eInSeconds > 0 )
    {
        auto  meta  = client . HeadObject ( BucketName ,
ObjectNme );

        std :: string  restoreSta  tus = meta . result ().
HttpMetaDa  ta ()[" x - oss - restore "];
        std :: transform ( restoreSta  tus . begin (), restoreSta
tus . end (), restoreSta  tus . begin (), :: tolower );
        if (! restoreSta  tus . empty () &&
restoreSta  tus . compare ( 0 , onGoingRes  tore . size (),
onGoingRes  tore )== 0 ) {
            std :: cout << " success , restore  status :" <<
restoreSta  tus << std :: endl ;
            /* The  archive  object  is  restored .*/
            break ;
        }

        std :: cout << " info , WaitTime :" << maxWaitTim
eInSeconds
        << "; restore  status :" << restoreSta  tus << std ::
endl ;

        std :: this_threa  d :: sleep_for ( std :: chrono :: seconds
( 10 ));
        maxWaitTim  eInSeconds --;
    }

    if ( maxWaitTim  eInSeconds == 0 )
    {
        std :: cout << " RestoreObj  ect  fail , TimeoutExc
eption " << std :: endl ;
    }

    /* Release  network  resources . */
    ShutdownSd k ();
    return 0 ;
}

```

For more information about archive storage classes, see [Introduction to storage classes](#). For more information about related status codes, see [RestoreObject](#) in API Reference.

5.10.9 Manage a symbolic link

This document describes how to create a symbolic link and obtain the object content to which the symbolic link maps.

Create a symbolic link

A symbolic link is a special object that maps to an object. It is similar to a shortcut used in Windows. You can use a symbolic link to customize Object Meta.

Use the following code to create a symbolic link:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";
    std :: string LinkName = " yourLinkName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Set the HTTP header . */
    auto meta = ObjectMetaData ();
    meta . setContent Type ( " text / plain " );

    /* Set User Meta . */
    meta . UserMetadata () [ " meta " ] = " meta - value ";

    /* Create the symbolic link . */
    CreateSymlinkRequest request ( BucketName , ObjectName , meta );
    request . SetSymlink Target ( LinkObjectName );
    auto outcome = client . CreateSymlink ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " CreateSymlink fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
}
```



```

        return 0 ;
    }

```

Obtain object content of a symbolic link

Use the following code to obtain the object content of a symbolic link:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string LinkName = " yourLinkName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Obtain the object content of a symbolic link . */
    GetSymlinkRequest request ( BucketName , LinkName );
    auto outcome = client . GetSymlink ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " GetSymlink fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
    else {
        std :: cout << " GetSymlink success Symlink name : "
<< outcome . result () . SymlinkTarget () << std :: endl ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```



Note:

To obtain a symbolic link, you must have the read permission on it. For more information about symbolic links, see [GetSymlink](#).

5.11 Authorized access

This topic describes how to authorize access to OSS.

Use STS for temporary access authorization

You can use Alibaba Cloud STS to authorize temporary access to OSS. STS is a Web service that provides a temporary access token to a cloud computing user. You can use STS to grant a third-party application or a RAM user (whose user ID is managed by yourself) an access credential with a customized validity period and permissions. For more information about STS, see [STS Introduction](#).

STS benefits:

- You do not need to expose your long-term key (AccessKey) to a third-party application. You need only to generate an access token and send the access token to the third-party application. You can customize the access permissions and validity period of this token.
- The access token automatically becomes invalid after expiration.

For more information about the process of access to OSS with STS, see the "[Identity authentication](#)" section in OSS Developer Guide.

Use the following code to create a signature request with STS:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string SecurityToken = " securityToken ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , SecurityToken , conf )

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

Use a signed URL to authorize temporary access

You can provide a signed URL to a visitor for temporary access. When you generate a signature for a URL, you can specify the valid period for the URL. After the valid period expires, access from visitors is restricted.

- Use a signed URL to upload an object

Use the following code to generate a signed URL to upload an object:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string PutobjectUrlName = " yourPutobjectUrlName " ;

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Specify the valid period for the URL . */
    std :: time_t t = std :: time ( nullptr ) + 1200 ;
    /* Generate the signed URL for uploading an object . */
    auto genOutcome = client . GeneratePresignedUrl (
        BucketName , PutobjectUrlName , t , Http :: Put );
    if ( genOutcome . isSuccess () ) {
        std :: cout << " GeneratePresignedUrl success , GeneratedUrl : " << genOutcome . result () . c_str () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GeneratePresignedUrl fail " <<
            " , code : " << genOutcome . error () . Code () <<
            " , message : " << genOutcome . error () . Message () <<
            " , requestId : " << genOutcome . error () . RequestId () <<
            std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    std :: shared_ptr < std :: istream > content = std ::
        make_shared < std :: stringstream > ();
    * content << " test cpp sdk ";

    /* Use the URL to upload the object . */
    auto outcome = client . PutObjectByUrl ( genOutcome .
        result () , content );
```

```

        if (! outcome . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " PutObjectB yUrl fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () <<
            std :: endl ;
            ShutdownSd k ();
            return - 1 ;
        }

        /* Release network resources . */
        ShutdownSd k ();
        return 0 ;
    }
}

```

- Use a signed URL to download an object

Use the following code to generate a signed URL to download an object:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string GetobjectUrlName = " yourGetObjectUrlName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Specify the valid period for the URL . */
    std :: time_t t = std :: time ( nullptr ) + 1200 ;
    /* Generate the URL to download the object . */
    auto genOutcome = client . GeneratePresignedUrl (
    BucketName , GetobjectUrlName , t , Http :: Get );
    if ( genOutcome . isSuccess ()) {
        std :: cout << " GeneratePresignedUrl success , Gen
        url :" << genOutcome . result (). c_str () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GeneratePresignedUrl fail " <<
        ", code :" << genOutcome . error (). Code () <<
        ", message :" << genOutcome . error (). Message () <<
        ", requestId :" << genOutcome . error (). RequestId () <<
        std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }
}

/* Use the URL to download the object . */

```

```

    auto outcome = client . GetObjectB yUrl ( genOutcome .
result ());

    if (! outcome . isSuccess ()) {
        /* Handle exceptions . */
        std :: cout << " GetObjectB yUrl fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () <<
std :: endl ;
        ShutdownSd k ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSd k ();
    return 0 ;
}

```

5.12 Lifecycle rules

This topic describes how to manage lifecycle rules.

You can configure lifecycle rules for OSS buckets or objects to automatically delete expired objects and parts or change the storage class of expired objects to IA or Archive, saving storage costs. A lifecycle rule includes the following elements:

- **Rule ID:** Identifies a lifecycle rule. A rule ID is unique in a bucket.
- **Strategy:** You can set one of the following two application strategies for a rule. The strategy for all rules set for a bucket must be the same.
 - The rule applies to objects with specified prefixes. You can create multiple rules with this strategy. Prefixes specified in different rules must be different.
 - The rule applies to the entire bucket. You can create only one rule with this strategy.
- **Expiration period:**
 - Specifies the expiration period (N days) of a lifecycle rule. An object expires N days after it is modified for the last time.
 - Specifies an expiration time. Objects modified before the time expire.
- **Status:** Indicates whether the rule is enabled or disabled.

Lifecycle rules apply to resources uploaded by uploadPart. The last modification time of an object is the time the multipart upload event was initiated.

For more information about lifecycle management, see [Manage object lifecycle](#).

Configure lifecycle rules

Use the following code to configure lifecycle rules:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    SetBucketLifecycleRequest request ( BucketName );
    std :: string date ( " 2022 - 10 - 12T00 : 00 : 00 . 000Z " ) ;

    /* Specify the period to retain objects after their
    most recent modification time . After the period
    expires , the objects are deleted . */
    auto rule1 = LifecycleRule ();
    rule1 . setID ( " rule1 " );
    rule1 . setPrefix ( " test1 /" );
    rule1 . setStatus ( RuleStatus :: Enabled );
    rule1 . setExpiration ( 3 );

    /* Specify the date before which objects are
    created . After the date , the objects expire . */
    auto rule2 = LifecycleRule ();
    rule2 . setID ( " rule2 " );
    rule2 . setPrefix ( " test2 /" );
    rule2 . setStatus ( RuleStatus :: Disabled );
    rule2 . setExpiration ( date );

    /* Configure lifecycle rules . */
    LifecycleRuleList list { rule1 , rule2 };
    request . setLifecycleRules ( list );
    auto outcome = client . SetBucketLifecycle ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " SetBucketLifecycle fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

View lifecycle rules

Use the following code to view lifecycle rules:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* View lifecycle rules . */
    auto outcome = client . GetBucketLifecycle ( BucketName );

    if ( outcome . isSuccess () ) {
        std :: cout << " GetBucketLifecycle success ," << std :: endl ;
        for ( auto const rule : outcome . result (). LifecycleRules () ) {
            std :: cout << " rule : " << rule . ID () << " , " << rule . Prefix () << " , " << rule . Status () << " , " << " hasExpiration : " << rule . hasExpiration () << " , " << " hasTransitionList : " << rule . hasTransitionList () << " , " << std :: endl ;
        }
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GetBucketLifecycle fail " << " , code : " << outcome . error (). Code () << " , message : " << outcome . error (). Message () << " , requestId : " << outcome . error (). RequestId () << std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

Clear lifecycle rules

Use the following code to clear lifecycle rules:

```
# include < alibabacloud / oss / OssClient . h >
```

```

using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std::string AccessKeyId = "yourAccess KeyId ";
    std::string AccessKeySecret = "yourAccess KeySecret ";
    std::string Endpoint = "yourEndpoint ";
    std::string BucketName = "yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Clear lifecycle rules . */
    DeleteBucketLifecycleRequest request ( BucketName );
    auto outcome = client.DeleteBucketLifecycle ( request );

    if ( ! outcome.IsSuccess () ) {
        /* Handle exceptions . */
        std::cout << "DeleteBucketLifecycle fail " <<
            ", code:" << outcome.error().Code() <<
            ", message:" << outcome.error().Message() <<
            ", requestId:" << outcome.error().RequestId() << std
        :: endl;
        ShutdownSdk();
        return -1;
    }

    /* Release network resources . */
    ShutdownSdk();
    return 0;
}

```

5.13 CORS

This topic describes how to perform cross-origin resource sharing (CORS).

CORS allows Web applications to access resources that belong to another region. OSS provides CORS APIs to facilitate cross-origin access control.

For more information, see the "[CORS](#)" and "[PutBucketcors](#)" sections in OSS Developer Guide.

Configure CORS rules

Use the following code to configure CORS rules for a specified bucket:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */

```



```

std::string AccessKeyId = "yourAccessKeyId";
std::string AccessKeySecret = "yourAccessKeySecret";
std::string Endpoint = "yourEndpoint";
std::string BucketName = "yourBucketName";

/* Initialize network resources. */
InitializeSdk();

ClientConfiguration conf;
OssClient client(Endpoint, AccessKeyId, AccessKeySecret, conf);

SetBucketCORSRequest request(BucketName);

/* Configure CORS rules. */
auto rule1 = CORSRule();
/* Specify the cross-origin source to share resources. */
rule1.addAllowedOrigin("http://www.a.com");
/* Specify the allowed CORS request method. (You can select GET, PUT, POST, DELETE or HEAD.)* */
rule1.addAllowedMethod("POST");
/* Determine whether to allow the header specified in Access-Control-Request-Headers of the pre-flight (OPTIONS) request. */
rule1.addAllowedHeader("*");
/* Specify the response header that allows user access from applications. */
rule1.addExposeHeader("x-oss-test");
/* You can set a maximum of 10 rules. */
request.addCORSRule(rule1);

auto rule2 = CORSRule();
rule2.addAllowedOrigin("http://www.b.com");
rule2.addAllowedMethod("GET");
rule2.addExposeHeader("x-oss-test2");
rule2.setMaxAgeSeconds(100);
request.addCORSRule(rule2);

auto outcome = client.SetBucketCORS(request);

if (!outcome.IsSuccess()) {
    /* Handle exceptions. */
    std::cout << "SetBucketCORS fail " <<
        ", code:" << outcome.error().Code() <<
        ", message:" << outcome.error().Message() <<
        ", requestId:" << outcome.error().RequestId() << std::endl;
    ShutdownSdk();
    return -1;
}

/* Release network resources. */
ShutdownSdk();
return 0;
}

```

View CORS rules

Use the following code to view CORS rules set for a bucket:

```
#include <alibabacloud/oss/OssClient.h>
```

```

using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";

    /* Initialize network resources . */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client ( Endpoint, AccessKeyId, AccessKeySecret, conf );

    /* View CORS rules . */
    auto outcome = client.GetBucketCORS ( BucketName );

    if ( outcome.IsSuccess () ) {
        std::cout << "GetBucketCORS success " << std::endl;
        for ( auto const rule : outcome.result().CORSRules () ) {
            std::cout << "GetBucketCORS List:" << std::endl;
            for ( auto const origin : rule.AllowedOrigins () ) {
                std::cout << "Allowed origin:" << origin << std::endl;
            }
        }
    }
    else {
        /* Handle exceptions . */
        std::cout << "GetBucketCORS fail " <<
            ", code:" << outcome.error().Code () <<
            ", message:" << outcome.error().Message () <<
            ", requestId:" << outcome.error().RequestId () << std::endl;
        ShutdownSdk();
        return -1;
    }

    /* Release network resources . */
    ShutdownSdk();
    return 0;
}

```

Delete CORS rules

Use the following code to delete the CORS rules set for a specified bucket:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std::string AccessKeyId = "yourAccessKeyId";

```

```

std :: string AccessKeySecret = "yourAccessKeySecret ";
std :: string Endpoint = "yourEndpoint ";
std :: string BucketName = "yourBucketName ";

/* Initialize network resources . */
InitializeSdk ();

ClientConfiguration conf ;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

/* Delete the CORS rules . */
auto outcome = client . DeleteBucketCors ( BucketName );

if (! outcome . isSuccess ()) {
    /* Handle exceptions . */
    std :: cout << "DeleteBucketCors fail " <<
        ", code : " << outcome . error (). Code () <<
        ", message : " << outcome . error (). Message () <<
        ", requestId : " << outcome . error (). RequestId () << std
:: endl ;
    ShutdownSdk ();
    return - 1 ;
}

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

5.14 Access logging

You can enable access logging to record access to a bucket in a log. The log is stored in a specified bucket.

The log format is as follows:

```

< TargetPrefix >< SourceBucket >- YYYY - mm - DD - HH - MM - SS -
UniqueString

```

For more information about access logging, see the "[Configure access logging](#)" section in OSS Developer Guide.

Enable access logging

Use the following code to enable access logging for a bucket:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = "yourAccessKeyId ";
    std :: string AccessKeySecret = "yourAccessKeySecret ";
    std :: string Endpoint = "yourEndpoint ";

```

```

std :: string BucketName = " yourBucket Name ";
std :: string TargetBucketName = " yourTarget BucketName ";
std :: string TargetPrefix = " yourTarget Prefix ";

/* Initialize network resources . */
InitializeSdk ();

ClientConfiguration conf ;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

/* Enable access logging . */
SetBucketLoggingRequest request ( BucketName , TargetBucketName , TargetPrefix );
auto outcome = client . SetBucketLogging ( request );

if ( ! outcome . isSuccess () ) {
    /* Handle exceptions . */
    std :: cout << " SetBucketLogging fail " <<
        ", code : " << outcome . error () . Code () <<
        ", message : " << outcome . error () . Message () <<
        ", requestId : " << outcome . error () . RequestId () << std
:: endl ;
    ShutdownSdk ();
    return - 1 ;
}

/* Release network resources . */
ShutdownSdk ();
return 0 ;
}

```

View access logging configurations

Use the following code to view the access logging configurations of a bucket:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* View the access logging configurations . */
    GetBucketLoggingRequest request ( BucketName );
    auto outcome = client . GetBucketLogging ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " GetBucketLogging success , TargetBucket : " << outcome . result () . TargetBucket <<

```

```

        ", TargetPref ix : " << outcome . result (). TargetPref ix
    () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GetBucketL ogging fail " <<
        ", code : " << outcome . error (). Code () <<
        ", message : " << outcome . error (). Message () <<
        ", requestId : " << outcome . error (). RequestId () << std
        :: endl ;
        ShutdownSd k ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSd k ();
    return 0 ;
}

```

Disable access logging

Use the following code to disable access logging for a bucket:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account informatio n . */
    std :: string AccessKeyI d = " yourAccess KeyId ";
    std :: string AccessKeyS ecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoi nt ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    Initialize Sdk ();

    ClientConf igureation conf ;
    OssClient client ( Endpoint , AccessKeyI d , AccessKeyS
ecret , conf );

    /* Disable access logging . */
    DeleteBuck etLoggingR equest request ( BucketName );
    auto outcome = client . DeleteBuck etLogging ( request );

    if (! outcome . isSuccess ()) {
        /* Handle exceptions . */
        std :: cout << " DeleteBuck etLogging fail " <<
        ", code : " << outcome . error (). Code () <<
        ", message : " << outcome . error (). Message () <<
        ", requestId : " << outcome . error (). RequestId () << std
        :: endl ;
        ShutdownSd k ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSd k ();
    return 0 ;
}

```

```
}
```

5.15 Static website hosting

You can set the static website hosting mode for your bucket. After the configuration takes effect, you can use the bucket domain to access this static website and be redirected to a specified index page or error page.

For more information about static website hosting, see the "[Configure static website hosting](#)" section in OSS Developer Guide.

Configure static website hosting

Use the following code to configure static website hosting:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Configure static website hosting . */
    SetBucketWebsiteRequest request ( BucketName );
    request . setIndexDocument ( " index . html " );
    request . setErrorDocument ( " error . html " );

    auto outcome = client . SetBucketWebsite ( request );

    if (! outcome . isSuccess ()) {
        /* Handle exceptions . */
        std :: cout << " SetBucketWebsite fail " <<
            ", code : " << outcome . error (). Code () <<
            ", message : " << outcome . error (). Message () <<
            ", requestId : " << outcome . error (). RequestId () << std
        :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

```
}
```

View static website hosting configurations

Use the following code to view static website hosting configurations:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* View static website hosting configurations . */
    GetBucketWebsiteRequest request ( BucketName );
    auto outcome = client . GetBucketWebsite ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " GetBucketWebsite success , IndexDocument : " << outcome . result (). IndexDocument () << " , ErrorDocument : " << outcome . result (). ErrorDocument () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GetBucketWebsite fail " << " , code : " << outcome . error (). Code () << " , message : " << outcome . error (). Message () << " , requestId : " << outcome . error (). RequestId () << std :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

Delete static website hosting configurations

Use the following code to delete static website hosting configurations:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
```

```

std :: string AccessKeySecret = "yourAccessKeySecret";
std :: string Endpoint = "yourEndpoint";
std :: string BucketName = "yourBucketName";

/* Initialize network resources . */
InitializeSdk();

ClientConfiguration conf;
OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

/* Delete static website hosting configurations . */
DeleteBucketWebsiteRequest request ( BucketName );

auto outcome = client . DeleteBucketWebsite ( request );

if ( ! outcome . isSuccess () ) {
    /* Handle exceptions . */
    std :: cout << "DeleteBucketWebsite fail " <<
        ", code :" << outcome . error () . Code () <<
        ", message :" << outcome . error () . Message () <<
        ", requestId :" << outcome . error () . RequestId () << std
:: endl ;
    ShutdownSdk();
    return - 1 ;
}

/* Release network resources . */
ShutdownSdk();
return 0 ;
}

```

5.16 Anti-leech

This topic describes how to use anti-leech.

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leech, see the "[Anti-leeching settings](#)" section in the OSS Developer Guide.

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```

#include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

```



```

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf ) ;

    /* Configure the referer whitelist . */
    SetBucketRefererRequest request ( BucketName );
    request . addReferer ( " http :// www . referersample . com " );
    request . addReferer ( " https :// www . referersample . com " );
    request . addReferer ( " https :// www . ?. referersample . com " );
    request . addReferer ( " https :// www . *. cn " );
    request . setAllowEmptyReferer ( false );

    auto outcome = client . SetBucketReferer ( request );

    if ( ! outcome . isSuccess () ) {
        /* Handle exceptions . */
        std :: cout << " SetBucketReferer fail " <<
            ", code : " << outcome . error () . Code () <<
            ", message : " << outcome . error () . Message () <<
            ", requestId : " << outcome . error () . RequestId () << std
            :: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Obtain a referer whitelist

Run the following code to obtain a referer whitelist:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;

```

```

    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Obtain the referer list for a bucket . */
    GetBucketRefererRequest request ( BucketName );
    auto outcome = client . GetBucketReferer ( request );

    if ( outcome . isSuccess () ) {
        std :: cout << " GetBucketReferer success ,
AllowEmpty Referer : " << outcome . result (). AllowEmpty Referer
() <<
        " , Referer size : " << outcome . result (). RefererList
(). size () << std :: endl ;
    }
    else {
        /* Handle exceptions . */
        std :: cout << " GetBucketReferer fail " <<
        " , code : " << outcome . error (). Code () <<
        " , message : " << outcome . error (). Message () <<
        " , requestId : " << outcome . error (). RequestId () << std
:: endl ;
        ShutdownSdk ();
        return - 1 ;
    }

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```

# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret ,
conf );

    /* You cannot clear a referer whitelist directly .
To clear a referer whitelist , you must create a
rule that allows an empty referer field and replace
the original rule with the new rule . */
    SetBucketRefererRequest request ( BucketName );
    request . setAllowEmptyReferer ( true );

    auto outcome = client . SetBucketReferer ( request );
}

```

```
        if (! outcome . isSuccess ()) {
            /* Handle exceptions . */
            std :: cout << " CleanBucke tReferer fail " <<
            ", code :" << outcome . error (). Code () <<
            ", message :" << outcome . error (). Message () <<
            ", requestId :" << outcome . error (). RequestId () << std
:: endl ;
            ShutdownSd k ();
            return - 1 ;
        }

        /* Release network resources . */
        ShutdownSd k ();
        return 0 ;
    }
}
```

5.17 IMG

Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service provided by OSS. IMG is capable of processing large amounts of images. After source images are uploaded to OSS, you can use Representational State Transfer (RESTful) APIs to process images anytime, anywhere, and on any Internet device.

For more information about IMG, see [OSS Image Processing Guide](#).

Basic features

IMG has the following features:

- [Obtain image information](#)
- [Convert formats](#)
- [Resize images](#)
- [Crop images](#)
- [Rotate images](#)
- [Apply effects](#)
- [Add watermarks](#): allows you to add image, text, or image-text watermarks to another image.
- [Customize image styles](#)
- [Cascade operations](#): calls multiple IMG operations.

How to use IMG

IMG uses the standard HTTP GET request. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is Private, only authorized users are allowed access.

- Anonymous access

You can use level-3 domains to access processed images. The access format is as follows:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >?  
? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Name	Description
bucket	The name of the bucket.
endpoint	The endpoint used to access the region to which the bucket belongs.
object	The name of the image object.
image	The reserved identifier for IMG.
action	Operations on the image, such as resizing, cropping, and rotating.

Name	Description
param	The parameter corresponding to the operations performed on the image.

- Basic operations

For example, scale an image to a width of 100 px. Adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customize an image style

You can use level-3 domains to access processed images. The access format is as follows:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

■ **style**: specifies the reserved identifier of a custom image style.

■ **yourStyleName**: specifies the name of a custom image style. It is the name of the rule that is specified in the OSS console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image in sequence. The format is as follows:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Access through HTTPS

IMG supports access through HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

Authorized access allows you to access an image through HTTPS, customize the image style, and perform cascade operations.

Use the following code to generate a signed URL for IMG:

```
# include <alibabacloud/oss/OssClient.h>
using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information. */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";
    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";
    std::string ObjectName = "yourObjectName";

    /* Initialize network resources. */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client ( Endpoint, AccessKeyId, AccessKeySecret, conf );

    /* Generate the signed URL. */
    std::string Process = "image/resize,m_fixed,w_100,h_100";
    GeneratePresignedUrlRequest request ( BucketName, ObjectName, Http::Get );
    request.setProcess ( Process );
    auto outcome = client.GeneratePresignedUrl ( request );

    /* Release network resources. */
    ShutdownSdk();
    return 0;
}
```

- **Access with an SDK**

You can use an SDK to access and process an image object.

The SDK allows you to access an image through HTTPS, customize the image style, and perform cascade operations.

- **Basic operations**

Use the following code to perform basic operations on an image:

```
# include <alibabacloud/oss/OssClient.h>
```

```

using namespace AlibabaCloud::OSS;

int main ( void )
{
    /* Initialize the OSS account information . */
    std::string AccessKeyId = "yourAccessKeyId";
    std::string AccessKeySecret = "yourAccessKeySecret";

    std::string Endpoint = "yourEndpoint";
    std::string BucketName = "yourBucketName";
    std::string ObjectName = "yourObjectName";

    /* Initialize network resources . */
    InitializeSdk();

    ClientConfiguration conf;
    OssClient client ( Endpoint, AccessKeyId, AccessKeySecret, conf );

    /* Scale the image . */
    std::string Process = "image/resize,m_fixed,w_100,h_100";
    GetObjectRequest request ( BucketName, ObjectName );
    request.setProcess ( Process );
    auto outcome = client.GetObject ( request );

    /* Crop the image . */
    Process = "image/crop,w_100,h_100,x_100,y_100,return_1";
    request.setProcess ( Process );
    outcome = client.GetObject ( request );

    /* Rotate the image . */
    Process = "image/rotate,90";
    request.setProcess ( Process );
    outcome = client.GetObject ( request );

    /* Sharpen the image . */
    Process = "image/sharpen,100";
    request.setProcess ( Process );
    outcome = client.GetObject ( request );

    /* Add the watermark . */
    Process = "image/watermark,text_SGVsbG8g5Zu-54mH5pyN5YqhIQ";
    request.setProcess ( Process );
    outcome = client.GetObject ( request );

    /* Convert the image format . */
    Process = "image/format,png";
    request.setProcess ( Process );
    outcome = client.GetObject ( request );

    /* Obtain image information . */
    request = request ( BucketName, ObjectName, "image/info" );
    outcome = client.GetObject ( request );

    /* Release network resources . */
    ShutdownSdk();
    return 0;
}

```

```
}

```

- Customize an image style

Use the following code to customize an image style:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf ) ;

    /* Customize the image style . */
    std :: string Process = " style /< yourCustom StyleName ";
    ";
    GetObjectRequest request ( BucketName , ObjectName );
    request . setProcess ( Process );
    auto outcome = client . GetObject ( request );

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}
```

- Cascade operations

Use the following code to perform cascade operations on an image:

```
# include < alibabacloud / oss / OssClient . h >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string ObjectName = " yourObject Name ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );
```



```

    /* Perform cascade operations . */
    std :: string Process = " image / resize , m_fixed , w_100
, h_100 / rotate , 90 ";
    GetObjectRequest request ( BucketName , ObjectName );
    request . setProcess ( Process );
    auto outcome = client . GetObject ( request );

    /* Release network resources . */
    ShutdownSdk ();
    return 0 ;
}

```

Implement permanent storage for a processed image

Use the following code to store a processed image in a bucket for permanent storage:

```

# include < alibabacloud / oss / OssClient . h >
# include < sstream >
using namespace AlibabaCloud :: OSS ;

int main ( void )
{
    /* Initialize the OSS account information . */
    std :: string AccessKeyId = " yourAccess KeyId ";
    std :: string AccessKeySecret = " yourAccess KeySecret ";
    std :: string Endpoint = " yourEndpoint ";
    std :: string BucketName = " yourBucket Name ";
    std :: string SourceObjectName = " yourSource ObjectName ";
    std :: string TargetObjectName = " yourTarget ObjectName ";

    /* Initialize network resources . */
    InitializeSdk ();

    ClientConfiguration conf ;
    OssClient client ( Endpoint , AccessKeyId , AccessKeySecret , conf );

    /* Scale the image . */
    std :: string Process = " image / resize , m_fixed , w_100 ,
h_100 ";
    std :: stringstream ss ;
    ss << Process
<< "| sys / saveas "
<< ", o_ " << Base64EncodeUrlSafe ( TargetObjectName )
<< ", b_ " << Base64EncodeUrlSafe ( BucketName );
    ProcessObjectRequest request ( BucketName , SourceObjectName , ss . str ());
    auto outcome = client . ProcessObject ( request );

    /* Crop the image . */
    Process = " image / crop , w_100 , h_100 , x_100 , y_100 , r_1
";
    ss . str ("" );
    ss << Process
<< "| sys / saveas "
<< ", o_ " << Base64EncodeUrlSafe ( TargetObjectName )
<< ", b_ " << Base64EncodeUrlSafe ( BucketName );
    request = request ( BucketName , SourceObjectName , ss . str ());
    outcome = client . ProcessObject ( request );

    /* Rotate the image . */
}

```

```

        Process = " image / rotate , 90 ";
        ss . str ("");
        ss << Process
<<"| sys / saveas "
<< " , o_ " << Base64Enco deUrlSafe ( TargetObje ctName )
<< " , b_ " << Base64Enco deUrlSafe ( BucketName );
        request = request ( BucketName , SourceObje ctName , ss .
str ());
        outcome = client . ProcessObj ect ( request );

/* Sharpen the image . */
        Process = " image / sharpen , 100 ";
        ss . str ("");
        ss << Process
<<"| sys / saveas "
<< " , o_ " << Base64Enco deUrlSafe ( TargetObje ctName )
<< " , b_ " << Base64Enco deUrlSafe ( BucketName );
        request = request ( BucketName , SourceObje ctName , ss .
str ());
        outcome = client . ProcessObj ect ( request );

/* Add the watermark . */
        Process = " image / watermark , text_SGVsb G8g5Zu - 54mH5pyN5Y
qhIQ ";
        ss . str ("");
        ss << Process
<<"| sys / saveas "
<< " , o_ " << Base64Enco deUrlSafe ( TargetObje ctName )
<< " , b_ " << Base64Enco deUrlSafe ( BucketName );
        request = request ( BucketName , SourceObje ctName , ss .
str ());
        outcome = client . ProcessObj ect ( request );

/* Convert the image format . */
        Process = " image / format , png ";
        ss . str ("");
        ss << Process
<<"| sys / saveas "
<< " , o_ " << Base64Enco deUrlSafe ( TargetObje ctName )
<< " , b_ " << Base64Enco deUrlSafe ( BucketName );
        request = request ( BucketName , SourceObje ctName , ss .
str ());
        outcome = client . ProcessObj ect ( request );

/* Release network resources . */
        ShutdownSd k ();
        return 0 ;
}

```

6 PHP

6.1 Preface

Source code

For the source code, see [GitHub](#).

Code samples

OSS SDK for PHP provides a variety of code samples for your reference or use. The following table describes the content of code samples:

Sample file	Content
Object.php	Object operations, including upload objects , download objects , and manage objects
MultipartUpload.php	Multipart upload
Signature.php	Authorized access with the signed URL
Callback.php	Upload callback
Image.php	Image processing
Bucket.php	Bucket operations, including create a bucket , delete a bucket , list buckets , and ACL .
BucketLifecycle.php	Configure, view, and delete the lifecycle rules of a bucket
BucketLogging.php	Configure, view, and delete the access log files of a bucket
BucketReferer.php	Configure, view, and delete the anti-leech settings of a bucket
BucketWebsite.php	Configure, view, and delete the static website hosting settings of a bucket
BucketCors.php	Configure, view, and delete the CORS settings of a bucket

6.2 Installation

Preparation

OSS PHP SDK is applicable to PHP 5.3 and later versions. PHP 5.6.22 is used in this topic as an example.

- Installation environment

You must install PHP cURL extensions:

- In Windows, see [Compile and use Alibaba Cloud OSS PHP SDK in Windows](#) to install PHP cURL extension. In Windows, if you cannot find the required module, specify the extension_dir to `C :/ Windows / System32 / in php.ini`.
 - In Ubuntu, you can use the apt-get package manager to install the PHP cURL extension by running `sudo apt - get install php - curl`.
 - In CentOS, you can use the yum package manager to install the PHP cURL extension by running `sudo yum install php - curl`.
- Check PHP version
 - Run `php - v` to check the current PHP version.
 - Run `php - m` to check whether cURL extension is installed.

Download SDK

- [Download from GitHub](#)
- [Download previous versions](#)

For more information, see [OSS API documents](#).



Note:

We recommend that you use the latest version of the SDK. You can download documents of OSS PHP SDK 2.0.0 and earlier versions from [here](#).

Install SDK

You can install the SDK in three methods:

• Composer

You can install the SDK through composer as follows:

1. Run `composer require aliyuncs / oss - sdk - php` in the root directory of your project, or declare the dependency on Alibaba Cloud OSS SDK for PHP in `composer . json` as follows:

```
"require": {
    "aliyuncs / oss - sdk - php": "~ 2 . x . x "
}
```

2. Run `composer install` to install the dependency. After the dependency is installed, check whether your directory structure complies with the following structure:

```
├── app . php
├── composer . json
├── composer . lock
└── vendor
```

The `app . php` is a user application. The `vendor /` directory contains the dependent library. You must introduce the dependency in `app . php` :

```
require_once __DIR__ . '/ vendor / autoload . php ';
```



Note:

- If your project references `autoload . php` , you do not need to introduce the file again after adding the SDK dependency.
- If a network error occurs when you use the Composer Dependency Manager, you can use the [Image Source](#) in the Composer China region by running the following command: `composer config - g repositories . packagist composer http :// packagist . phpcomposer . com .`

- Phar

You can install the SDK through phar as follows:

1. Select the required version and download the Phar package on [GitHub](#).
2. Introduce the file in your code:

```
require_once '/path/to/oss-sdk-php.phar';
```

Source code

You can install the SDK through source code as follows:

1. Select the required version and download the zip package on [GitHub](#).
2. Find the `autoload.php` file in the root directory of the unzipped package, and introduce the file in your code:

```
require_once '/path/to/oss-sdk/autoload.php';
```

6.3 Quick start

This topic describes how to use OSS PHP SDK to perform routine operations such as bucket creation, object uploads, and object downloads.

Create a bucket

Run the following code to create a bucket:

```
<? php
if ( is_file ( __DIR__ . '/../autoload.php' )) {
    require_once __DIR__ . '/../autoload.php';
}
if ( is_file ( __DIR__ . '/../vendor/autoload.php' )) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";
```

```
try {
    $ ossClient = new    OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );
    $ ossClient->createBucket ($ bucket );
} catch ( OssException $ e ) {
    print $ e->getMessage ();
}
```

For more information about bucket naming rules, see naming conventions in [Basic concepts](#). For more information about how to create a bucket, see [Manage a bucket](#).

For more information about endpoints, see [Regions and endpoints](#).

Upload objects

Run the following code to upload a file to OSS:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use    OSS \ OssClient ;
use    OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccessKeyId >";
$ accessKeySecret = "< yourAccessKeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = " < yourBucket Name >";
// Object name
$ object = " < yourObjectName >";
$ content = " Hi , OSS .";

try {
    $ ossClient = new    OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );
    $ ossClient->putObject ($ bucket , $ object , $ content );
} catch ( OssException $ e ) {
    print $ e->getMessage ();
}
```

For more information, see [Upload objects](#).

Download objects

Run the following code to download an object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";
// Object name
$ object = "< yourObject Name >";

try {
    $ ossClient = new OsscClient ($ accessKeyId , $ accessKeyS
    ecret , $ endpoint );
    $ content = $ ossClient -> getObject ($ bucket , $ object );
    print (" object content : " . $ content );
} catch ( OsscExcepti on $ e ) {
    print $ e -> getMessage ();
}
```

For more information, see [Download objects](#).

List objects

Run the following code to list objects in a specified bucket. You can list a maximum of 100 objects by default.

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
```



```
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ listObjectInfo = $ ossClient -> listObject s ( $ bucket );
    $ objectList = $ listObjectInfo -> getObjectList ();
    if ( ! empty ( $ objectList ) ) {
        foreach ( $ objectList as $ objectInfo ) {
            print ( $ objectInfo -> getKey () . "\ t " . $ objectInfo -> getSize
            () . "\ t " . $ objectInfo -> getLastMod ified () . "\ n " );
        }
    }
} catch ( OssiException $ e ) {
    print $ e -> getMessage ();
}
```

For more information, see [List objects](#) in [Manage objects](#).

Delete objects

Run the following code to delete a specified object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssiException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";
```

```
// Object name
$ object = "< yourObject Name >";

try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );
    $ ossClient -> deleteObject ($ bucket , $ object );
} catch ( OssException $ e ) {
    print $ e -> getMessage ();
}
```

For more information, see [Delete objects](#) in [Manage objects](#).

6.4 Initialization

OSSClient serves as the OSS Java client to manage OSS resources such as buckets and objects.

Create an OSSClient instance

To create an OSSClient instance, you need to specify an endpoint. For more information about endpoints, see [Regions and endpoints](#) and [Bind a custom domain name](#).

- Use an OSS domain to create an OSSClient instance

Run the following code to create an OSSClient instance with a domain assigned by OSS:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";

try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );
```

```

} catch ( OssExcepti on $ e ) {
    print $ e -> getMessage ();
}

```

- Use a custom domain (CNAME) to create an OSSClient instance

Run the following code to create an OSSClient instance with CNAME:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
$ endpoint = "< yourEndpoi nt >";

try {
    // The true parameter indicates that the CNAME is
    // enabled . CNAME indicates a custom domain bound to a
    // bucket .
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint , true );
} catch ( OssExcepti on $ e ) {
    print $ e -> getMessage ();
}

```



Note:

The `ossClient.listBuckets` method is not available when CNAME is used.

- Use STS to create an OSSClient instance

Run the following code to create an OSSClient instance with Security Token Service (STS):

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

```

```
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ securityTo ken = "< yourSecuri tyToken >";

try {
    $ ossClient = new OssClient (
        $ accessKeyI d , $ accessKeyS ecret , $ endpoint , false
    , $ securityTo ken );
} catch ( OssExcepti on $ e ) {
    print $ e -> getMessage ();
}
```

For more information, see [What is RAM and STS](#) and [Authorized access](#).

- Use a proxy server to create an OSSClient instance

Run the following code to create an OSSClient instance with a proxy server, which is supported by PHP 5.3 and later.

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram .
// console . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// The address of the proxy server , for example ,
// http ://< username >:< password >@< proxy ip >:< proxy port >
$ requestPro xy = "< yourReques tProxy >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com >";

try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint , false , $ requestPro xy );
} catch ( OssExcepti on $ e ) {
    print $ e -> getMessage ();
}
```

```
}
```

Configure network parameters

Run the following code to configure the network parameters of an OSSClient instance:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";

try {
    $ ossClient = new OsscClient (
        $ accessKeyId , $ accessKeyS ecret , $ endpoint );
    // Set the timeout time in seconds for data
    // transmissi on at the sockets layer . The default value
    // is 5184000 .
    $ ossClient -> setTimeout ( 3600 );
    // Set the timeout time in seconds when connection s
    // are establishe d . The default value is 10 .
    $ ossClient -> setConnect Timeout ( 10 );
} catch ( OsscExcepti on $ e ) {
    print $ e -> getMessage ();
}
```

6.5 Bucket

A bucket serves as a container that stores objects. Objects belong to a bucket.

For the complete code of the following scenarios, see [GitHub](#).

Create a bucket

Run the following code to create a bucket:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
```

```

        require_once __DIR__ . '/../vendor/autoload.php';
    }

    use OSS \ OssClient ;
    use OSS \ Core \ OssException ;

    // It is highly risky to log on with AccessKey of
    // an Alibaba Cloud account because the account has
    // permission s on all the APIs in OSS . We recommend
    // that you log on as a RAM user to access APIs
    // or perform routine operations and maintenance . To
    // create a RAM account , log on to https :// ram . console
    // . aliyun . com .
    $ accessKeyId = "< yourAccess KeyId >";
    $ accessKeySecret = "< yourAccess KeySecret >";
    // This example uses endpoint China ( Hangzhou ). Specify
    // the actual endpoint based on your requirements .
    $ endpoint = "http :// oss - cn - hangzhou . aliyuncs . com ";
    // Bucket name
    $ bucket = "< yourBucket Name >";
    try {
        $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );
        // Set the storage class to Infrequent Access for
        // a bucket . The default value is Standard .
        $ options = array (
            OssClient :: OSS_STORAGE_CLASS => OssClient :: OSS_STORAGE_CLASS_IA
        );
        // Set the ACL to public read for a bucket . The
        // default value is Private .
        $ ossClient -> createBucket ($ bucket , OssClient ::
            OSS_ACL_TYPE_PUBLIC_READ , $ options );
    } catch ( OssException $ e ) {
        printf ( __FUNCTION__ . " : FAILED \n ");
        printf ($ e -> getMessage () . "\n ");
        return ;
    }
    print ( __FUNCTION__ . " : OK " . "\n ");

```

For more information about bucket naming rules, see naming conventions in [Basic concepts](#).

Determine whether a bucket exists

Run the following code to determine whether a specified bucket exists:

```

<? php
if ( is_file ( __DIR__ . '/../autoload.php' )) {
    require_once __DIR__ . '/../autoload.php';
}
if ( is_file ( __DIR__ . '/../vendor/autoload.php' )) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend

```

```

        that you log on as a RAM user to access APIs
        or perform routine operations and maintenance. To
        create a RAM account, log on to https://ram.console
        .aliyun.com.
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China (Hangzhou). Specify
the actual endpoint based on your requirements.
$ endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
// Bucket name
$ bucket = "< yourBucket Name >";
try {
    $ ossClient = new OssClient($ accessKeyId, $ accessKeySecret, $ endpoint);

    $ res = $ ossClient->doesBucketExist($ bucket);
} catch (OssException $ e) {
    printf(__FUNCTION__ . ": FAILED \n");
    printf($ e->getMessage() . "\n");
    return;
}
if ($ res === true) {
    print(__FUNCTION__ . ": OK " . "\n");
} else {
    print(__FUNCTION__ . ": FAILED " . "\n");
}

```

List buckets

Run the following code to list buckets:

```

<? php
if (is_file(__DIR__ . '/../autoload.php')) {
    require_once __DIR__ . '/../autoload.php';
}
if (is_file(__DIR__ . '/../vendor/autoload.php')) {
    require_once __DIR__ . '/../vendor/autoload.php';
}

use OSS\OssClient;
use OSS\Core\OssException;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS. We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance. To
create a RAM account, log on to https://ram.console
.aliyun.com.
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China (Hangzhou). Specify
the actual endpoint based on your requirements.
$ endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
try {
    $ ossClient = new OssClient($ accessKeyId, $ accessKeySecret, $ endpoint);

    $ bucketListInfo = $ ossClient->listBuckets();
} catch (OssException $ e) {
    printf(__FUNCTION__ . ": FAILED \n");
    printf($ e->getMessage() . "\n");
}

```

```

        return ;
    }
    $ bucketList = $ bucketList Info -> getBucketList ();
    foreach ($ bucketList as $ bucket ) {
        print ($ bucket -> getLocation () . "\ t " . $ bucket ->
            getName () . "\ t " . $ bucket -> getCreatedate () . "\ n ");
    }
}

```

Configure an ACL for a bucket

The ACL of a bucket includes the following permissions.

Permission	Description	Value
Private	The bucket owner and the authorized users can read and write objects in the bucket. Other users cannot perform any operation on the objects.	OssClient::OSS_ACL_TYPE_PRIVATE
Public read	The bucket owner and the authorized users can read and write objects in the bucket. Other users can only read the objects in the bucket. Authorize this permission with caution.	OssClient::OSS_ACL_TYPE_PUBLIC_READ
Public read-write	All users can read and write objects in the bucket . Authorize this permission with caution.	OssClient::OSS_ACL_TYPE_PUBLIC_READ_WRITE

You can run the following code to configure an ACL for a bucket:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccessKeyId >";

```



```

$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";
// Set the ACL of the bucket to private .
$ acl = OssClient :: OSS_ACL_TYPE_PRIVATE ;
try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ ossClient -> putBucketAcl ( $ bucket , $ acl );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n " );
    printf ( $ e -> getMessage () . "\n " );
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n " );

```

Obtain the ACL for a bucket

Run the following code to obtain the ACL for a bucket:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
.aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ res = $ ossClient -> getBucketAcl ( $ bucket );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n " );
    printf ( $ e -> getMessage () . "\n " );
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n " );
print ( ' acl : ' . $ res );

```

Obtain the region of a bucket

Run the following code to obtain the region (or location) for a bucket:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ Regions = $ ossClient -> getBucketL ocation ($ bucket );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");
var_dump ($ Regions );
```

For more information about region, see region in the [Basic concepts](#).

Obtain bucket information

Run the following code to obtain bucket information:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
```

```
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ Metas = $ ossClient -> getBucketM eta ($ bucket );
} catch ( OssiException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
var_dump ($ Metas );
```

Delete a bucket

Before a bucket is deleted, ensure that all objects in the bucket and fragments that are generated from multipart upload are deleted.



Note:

To delete the fragments that are generated from multipart upload, use [Bucket.ListMultipartUploads](#) to list all fragments, and then use [Bucket.AbortMultipartUpload](#) to delete the fragments.

Run the following code to delete a bucket:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssiException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
```

```

create a RAM account , log on to https :// ram . console
. aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    $ ossClient -> deleteBuck et ( $ bucket );
} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n " );
    printf ( $ e -> getMessage () . " \ n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . " \ n " );

```

6.6 Upload objects

6.6.1 Overview

In OSS, an object is the basic unit for data operations. OSS PHP SDK provides the following file upload methods:

- **Simple upload**: supports the upload of a file up to 5 GB in size.
- **Append upload**: supports the upload of a file up to 5 GB in size.
- **Multipart upload**: supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files



Note:

For more information about the usage scenarios of each upload method, see “Upload files” in OSS Developer Guide. After you complete the file upload, you can perform [upload callback](#).

6.6.2 Simple upload

Simple upload applies to two scenarios: upload a string and upload a local file.

Upload a string

You can run the following code to upload a string to OSS:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';

```

```

}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";
// Object name
$ object = "< yourObject Name >";
// Object content
$ content = " Hello OSS ";
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> putObject ($ bucket , $ object , $ content );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . " : OK " . "\ n ");

```

Upload a local file

You can run the following code to upload a local file to OSS:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";

```

```
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Bucket name
$ bucket = "< yourBucket Name >";
// Object name
$ object = "< yourObject Name >";
// < yourLocalFile > consists of the local file path
and the file name ( including the extension ), such as
/ users / local / myfile . txt
$ filePath = "< yourLocalFile >";

try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ ossClient -> uploadFile ( $ bucket , $ object , $ filePath );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . " : FAILED \n " );
    printf ( $ e -> getMessage () . "\n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . "\n " );
```

6.6.3 Append upload

You can append a string or a file to an OSS object.

You cannot perform copyObject for appended objects.

Append a string

Run the following code to append a string to an OSS object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Specify the bucket name .
$ bucket = "< yourBucket Name >";
// Specify the object name .
```

```

$ object = "< yourObject Name >";
// Specify the content of the string .
$content_array = array ( ' Hello OSS ', ' Hi OSS ', ' OSS OK
');
try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ position = $ ossClient -> appendObject ($ bucket , $ object ,
$content_array [ 0 ], 0 );
    $ position = $ ossClient -> appendObject ($ bucket , $ object ,
$content_array [ 1 ], $ position );
    $ position = $ ossClient -> appendObject ($ bucket , $ object ,
$content_array [ 2 ], $ position );
} catch (OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");

```

Append a file

Run the following code to append a local file to an OSS object:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
.aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
// Specify the bucket name .
$ bucket = "< yourBucket Name >";
// Specify the object name .
String objectName = "< yourObject Name >";
// Specify the first local file .
$ filePath = "< yourLocalFile1 >";
// Specify the second local file
$ filePath1 = "< yourLocalFile2 >";
try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

```

```

    $ position = $ ossClient -> appendFile ($ bucket , $ object , $
filePath , 0 );
    $ position = $ ossClient -> appendFile ($ bucket , $ object , $
filePath1 , $ position );
} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . ":  FAILED \ n " );
    printf ($ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . ":  OK " . "\ n " );

```

6.6.4 Multipart upload

To enable multipart upload, perform the following steps:

1. Initiate a multipart upload event.

You can call `$ossClient->initiateMultipartUpload` to return the globally unique `uploadId` created in OSS.

2. Upload parts.

You can call `$ossClient->uploadPart` to upload part data.



Notice:

- For parts with a same `uploadId`, parts are sequenced by their part numbers. If you have uploaded a part and use the same part number to upload another part, the later part will replace the former part.
- Except for the last part, the minimum size of other size is 100 KB. The size of the last part is not limited.

3. Complete multipart upload.

Call `$ossClient->completeMultipartUpload` to combine these parts into a complete object.

For the complete code of multipart upload, see [GitHub](#).

The following code is used as a complete example that describes the process of multipart upload:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssiClient ;
use OSS \ Core \ OssiExcepti on ;

```



```

use OSS \ Core \ OssUtil ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ uploadFile = "< yourLocalF ile >";

/**
 * Step 1 : Initiate a multipart upload event and
 * obtain the uploadId .
 */
try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeyS
    ecret , $ endpoint );

    // An uploadId is returned . It is the unique
    // identifier for a part upload event . You can initiate
    // related operations ( such as part upload cancelatio n
    // and query ) based on the uploadId .
    $ uploadId = $ ossClient -> initiateMu ltipartUpl oad ( $ bucket
    , $ object );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . " : initiateMu ltipartUpl oad
    FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . " : initiateMu ltipartUpl oad OK " . "\
n " );
/**
 * Step 2 : Upload parts .
 */
$ partSize = 10 * 1024 * 1024 ;
$ uploadFile Size = filesize ( $ uploadFile );
$ pieces = $ ossClient -> generateMu ltiuploadP arts ( $ uploadFile
    Size , $ partSize );
$ responseUp loadPart = array ();
$ uploadPosi tion = 0 ;
$ isCheckMd5 = true ;
foreach ( $ pieces as $ i => $ piece ) {
    $ fromPos = $ uploadPosi tion + ( integer ) $ piece [ $ ossClient
    :: OSS_SEEK_T 0 ];
    $ toPos = ( integer ) $ piece [ $ ossClient :: OSS_LENGTH ] + $
    fromPos - 1 ;
    $ upOptions = array (
        $ ossClient :: OSS_FILE_U PLOAD => $ uploadFile ,
        $ ossClient :: OSS_PART_N UM => ( $ i + 1 ),
        $ ossClient :: OSS_SEEK_T 0 => $ fromPos ,
        $ ossClient :: OSS_LENGTH => $ toPos - $ fromPos + 1 ,
        $ ossClient :: OSS_CHECK_ MD5 => $ isCheckMd5 ,
    );
    // MD5 check .

```

```

        if ( $ isCheckMd5 ) {
            $ contentMd5 = OssUtil :: getMd5SumF orFile ( $ uploadFile ,
$ fromPos , $ toPos );
            $ upOptions [ $ ossClient :: OSS_CONTEN T_MD5 ] = $ contentMd5
;
        }
        try {
            // Upload parts .
            $ responseUp loadPart [] = $ ossClient -> uploadPart ( $
bucket , $ object , $ uploadId , $ upOptions );
            } catch ( OssExcepti on $ e ) {
                printf ( __FUNCTION__ __ . ": initiateMu ltipartUpl oad ,
uploadPart - part #{ $ i } FAILED \ n " );
                printf ( $ e -> getMessage () . "\ n " );
                return ;
            }
            printf ( __FUNCTION__ __ . ": initiateMu ltipartUpl oad ,
uploadPart - part #{ $ i } OK \ n " );
        }
// $ uploadPart s is an array composed by the ETags
and part number ( PartNumber ) of each part .
$ uploadPart s = array ();
foreach ( $ responseUp loadPart as $ i => $ eTag ) {
    $ uploadPart s [] = array (
        ' PartNumber ' => ( $ i + 1 ),
        ' ETag ' => $ eTag ,
    );
}
/**
 * Step 3 : Complete multipart upload .
 */
try {
    // You must provide all valid $ uploadPart s when
you perform this operation . OSS verifies the validity
of all parts one by one after it receives $
uploadPart s . After part verificati on is successful ,
OSS combines these parts into a complete object .
    $ ossClient -> completeMu ltipartUpl oad ( $ bucket , $ object ,
$ uploadId , $ uploadPart s );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ __ . ": completeMu ltipartUpl oad
FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
printf ( __FUNCTION__ __ . ": completeMu ltipartUpl oad OK \ n
");

```

In the example above, \$options includes the following parameters:

Parameter	Description
\$ossClient::OSS_FILE_UPLOAD	File upload
OssClient::OSS_PART_NUM	Part number
OssClient::OSS_SEEK_TO	Specifies a start location.
OssClient::OSS_LENGTH	Object length
OssClient::OSS_PART_SIZE	Part size

Parameter	Description
OssClient::OSS_CHECK_MD5	Determines whether MD5 check is enabled. The MD5 check is enabled if the value is true.

Cancel a multipart upload event

You can call `$ossClient->abortMultipartUpload` to cancel a multipart upload event. If you cancel a multipart upload event, you are not allowed to perform any other operations with this uploadId anymore. The uploaded parts will be deleted.

Run the following code to cancel a multipart upload event:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ upload_id = "< yourUpload Id >";

try {
    $ ossClient = new OssClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> abortMulti partUpload ($ bucket , $ object , $
    upload_id );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
```

```
print ( __FUNCTION__ . ": OK " . "\n ");
```

Upload of a local file using multipart upload

Run the following code to upload a local file using multipart upload:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ file = "< yourLocalF ile >";

$ options = array (
    OsscClient :: OSS_CHECK_ MD5 => true ,
    OsscClient :: OSS_PART_S IZE => 1 ,
);
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> multiuploa dFile ($ bucket , $ object , $ file , $
    options );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
```

Upload a directory using multipart upload

Run the following code to upload a local directory (including all files in it) using multipart upload:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
```

```

if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM account , log on to https :// ram . console . aliyun
. com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ localDirec tory = ".";
$ prefix = " samples / codes ";
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    $ ossClient -> uploadDir ($ bucket , $ prefix , $ localDirec tory
);
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");

```

List uploaded parts

Run the following code to list uploaded parts:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";

```

```
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ uploadId = "< yourUpload Id >";

try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ listPartsInfo = $ ossClient -> listParts ( $ bucket , $ object , $ uploadId );
    foreach ( $ listPartsInfo -> getListPart () as $ partInfo )
    {
        print ( $ partInfo -> getPartNumber () . "\ t " . $ partInfo -> getSize () . "\ t " . $ partInfo -> getETag () . "\ t " . $ partInfo -> getLastModified () . "\ n " );
    }
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n " );
```

List all multipart upload events

Run the following code to list all multipart upload events:

```
<? php
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenance . To create
a RAM account , log on to https :// ram . console . aliyun
.com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ options = array (
    ' delimiter ' => '/',
    ' max - uploads ' => 100 ,
    ' key - marker ' => '',
```

```

        ' prefix ' => '',
        ' upload - id - marker ' => ''
    );
    try {
        $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

        $ listMultiPartUploadInfo = $ ossClient -> listMultiPartUploads ($ bucket , $ options );
    } catch ( OssException $ e ) {
        printf ( __FUNCTION__ . ": listMultiPartUploads FAILED \n ");
        printf ($ e -> getMessage () . "\n ");
        return ;
    }
    printf ( __FUNCTION__ . ": listMultiPartUploads OK \n ");
    $ listUploadInfo = $ listMultiPartUploadInfo -> getUploads ();
    var_dump ($ listUploadInfo );

```

Parameters of \$options are described as follows:

Parameter	Description
delimiter	Specifies a delimiter of a forward slash (/) used to group object names. The object between the specified prefix and the first occurrence of a delimiter of a forward slash (/) is commonPrefixes.
key-marker	Lists all part upload events with the object whose names start with a letter that comes after the key-marker value in the alphabetical order. You can use this parameter with the upload-id-marker parameter to specify the initial position for the specified returned result.
max-uploads	Specifies the maximum number of part upload events. The maximum value (also default value) you can set is 1,000.
prefix	Specifies the prefix that must be included in the returned object name. Note that if you use a prefix for query, the returned object name will contain the prefix.

Parameter	Description
upload-id-marker	You can use this parameter with the key-marker parameter to specify the initial position for the specified returned result. If you do not configure keyMarker, the uploadIdMarker parameter is invalid. If you configure key-marker, the query result contains: • All objects whose names start with a letter that comes after the keyMarker value. • All objects whose names start with a letter that is the same as the key-marker value in the alphabetical order and the value of uploadId greater than that of upload-id-marker.

6.6.5 Upload callback

Configure callback for simple upload

Run the following code to configure callback for simple upload.

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

$ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS ecret
, $ endpoint );

// Configure callback for uploads .
```



```
// callbackUrl is the IP address of the callback
server , such as http:// oss - demo . aliyuncs . com : 23450
or http:// 127 . 0 . 0 . 1 : 9090 .
// callbackHost is the value of the Host field
carried in the callback request , such as oss - cn -
hangzhou . aliyuncs . com .
$url =
    '{
        " callbackUrl ":"< yourCallbackServerUrl >",
        " callbackHost ":"< yourCallbackServerHost >",
        " callbackBody ":" bucket =${ bucket }& object =${ object }&
etag =${ etag }& size =${ size }& mimeType =${ mimeType }& imageInfo
. height =${ imageInfo . height }& imageInfo . width =${ imageInfo .
width }& imageInfo . format =${ imageInfo . format }& my_var1 =${ x :
var1 }& my_var2 =${ x : var2 }",
        " callbackBodyType ":" applicatio n / x - www - form -
urlencoded "
    }';

// Configure the custom parameters used to initiate a
callback request . Each parameter consists of a key
and a value . The key must start with x :.
$var =
    '{
        " x : var1 ":" value1 ",
        " x : var2 ":" value2 "
    }';
$options = array ( OssClient :: OSS_CALLBACK => $url ,
    OssClient :: OSS_CALLBACK_VAR => $var
);
$result = $ossClient -> putObject ( $bucket , $object ,
file_get_contents ( __FILE__ ), $options );
print_r ( $result [' body ']);
print_r ( $result [' info '][' http_code ']);
```

Configure callback for multipart upload

Run the following code to configure callback for multipart upload:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}
use OSS \ OssClient ;
use OSS \ Core \ OssException ;
use OSS \ Core \ OssUtil ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenance . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
$accessKeyId = "< yourAccess KeyId >";
$accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requirements .
$endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
```

```

$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ uploadFile = "< yourLocalFile >";

/**
 * Step 1 : Initiate a multipart upload event and get
 * the uploadId .
 */

$ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret
, $ endpoint );
// An uploadId is returned . It is the unique
identifier for a multipart upload event . You can
initiate related operations ( such as cancel or query
a multipart upload event ) based on the uploadId .
$ uploadId = $ ossClient -> initiateMultipartUpload ($ bucket , $
object );

print ( __FUNCTION__ . ": initiateMultipartUpload OK " . "\n" );
/**
 * Step 2 : Upload parts .
 */
$ partSize = 10 * 1024 * 1024 ;
$ uploadFileSize = filesize ($ uploadFile );
$ pieces = $ ossClient -> generateMultipartUploadParts ($ uploadFile
Size , $ partSize );
$ responseUploadPart = array ();
$ uploadPosition = 0 ;
$ isCheckMd5 = true ;
foreach ($ pieces as $ i => $ piece ) {
    $ fromPos = $ uploadPosition + ( integer )$ piece [$ ossClient
:: OSS_SEEK_T_0 ];
    $ toPos = ( integer )$ piece [$ ossClient :: OSS_LENGTH ] + $
fromPos - 1 ;
    $ upOptions = array (
        $ ossClient :: OSS_FILE_UPLOAD => $ uploadFile ,
        $ ossClient :: OSS_PART_NUMBER => ( $ i + 1 ),
        $ ossClient :: OSS_SEEK_T_0 => $ fromPos ,
        $ ossClient :: OSS_LENGTH => $ toPos - $ fromPos + 1 ,
        $ ossClient :: OSS_CHECK_MD5 => $ isCheckMd5 ,
    );
    // Perform the MD5 verification .
    if ( $ isCheckMd5 ) {
        $ contentMd5 = OssUtil :: getMd5SumFromFile ($ uploadFile ,
$ fromPos , $ toPos );
        $ upOptions [$ ossClient :: OSS_CONTENT_MD5 ] = $ contentMd5
;
    }
    try {
        // Upload the parts .
        $ responseUploadPart [] = $ ossClient -> uploadPart ($
bucket , $ object , $ uploadId , $ upOptions );
    } catch ( OssException $ e ) {
        printf ( __FUNCTION__ . ": initiateMultipartUpload ,
uploadPart - part #{$ i } FAILED \n " );
        printf ( $ e -> getMessage () . "\n " );
        return ;
    }
    printf ( __FUNCTION__ . ": initiateMultipartUpload ,
uploadPart - part #{$ i } OK \n " );
}
// $ uploadParts is an array that consists of the
ETags and the PartNumbers of each part .

```

```

$uploadParts = array();
foreach ($responseUploadParts as $i => $eTag) {
    $uploadParts[] = array(
        'PartNumber' => ($i + 1),
        'ETag' => $eTag,
    );
}

/**
 * Step 3 : Complete the upload .
 */
// The callbackUrl is the IP address of the
// callback server , such as http://oss-demo.aliyuncs.com
// : 23450 or http://127.0.0.1:9090 .
// callbackHost is the value of the Host field
// carried in the callback request , such as oss-cn-
// hangzhou.aliyuncs.com .
$json =
    '{
        "callbackUrl ":"< yourCallbackServerUrl >",
        "callbackHost ":"< yourCallbackServerHost >",
        "callbackBody ":"{" mimeType ":"{$ mimeType }"," size ":"
    ${ size }"," x : var1 ":"{$ x : var1 }"," x : var2 ":"{$ x : var2 }"}",
        "callbackBodyType ":"application/json "
    }';

// Configure the custom parameters used to initiate a
// callback request . Each parameter consists of a key
// and a value . The key must start with x :.
$var =
    '{
        " x : var1 ":" value1 ",
        " x : var2 ":" value2 "
    }';

$options = array ( OssClient::OSS_CALLBACK => $json ,
    OssClient::OSS_CALLBACK_VAR => $var );
// To perform the operation , all valid $uploadParts
// must be provided . OSS verifies the validity of all
// parts one by one after it receives the $uploadParts
// . After all the data parts are verified , the OSS
// combines these parts into a complete object .
$ossClient->completeMultipartUpload($bucket, $object, $
uploadId, $uploadParts, $options);

printf ( __FUNCTION__ . ": completeMultipartUpload OK \n
");

```

6.7 Download objects

6.7.1 Overview

This topic is the overview of object download in OSS PHP SDK.

For the complete code of object download, see [GitHub](#).

OSS PHP SDK provides the following download methods:

- [Download to your local file](#)

- [Download to local memory](#)
- [Range download](#)
- [Conditional download](#)

6.7.2 Download to your local file

You can run the following code to download the specified OSS object to a local file:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses the endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ localfile = "< yourLocalF ile >";
$ options = array (
    OsscClient :: OSS_FILE_D OWNLOAD => $ localfile
);

try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> getObject ($ bucket , $ object , $ options );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK , please check localfile : '
upload - test - object - name . txt ' . "\ n ");
```

6.7.3 Download to your local memory

Run the following code to download a specified object to local memory and print the content of the object.

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ content = $ ossClient -> getObject ($ bucket , $ object );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ($ content );
print ( __FUNCTION__ . " : OK " . "\ n ");
```

6.7.4 Range download

If you only need part of the data in an object, you can use range downloads to download specified content.

Run the following code to download the content of an object in a range of [0, 4] to local memories:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
```

```

if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses the endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
// Obtain data between the starting byte value of 0
// and the maximum byte value of 4 inclusive ( that
// is , a total of 5 bytes ). If the specified value
// range is invalid , for example , the specified range
// includes a negative number , or the specified value is
// larger than the object size , the entire object is
// downloaded .
$ options = array ( OsscClient :: OSS_RANGE => ' 0 - 4 ');
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ content = $ ossClient -> getObject ($ bucket , $ object , $
    options );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ($ content );
print ( __FUNCTION__ . ": OK " . "\ n ");

```

6.7.5 Conditional download

You can configure one or more conditions for downloads. If the specified conditions are met, the object can be downloaded. If the specified conditions are not met, an error code is returned and the object cannot not downloaded. You can configure the following conditions:

Parameter	Description	Configuration method
If-Modified-Since	If the specified time is earlier than the time the object is modified, the object can be downloaded . Otherwise, an error (304 Not modified) is returned.	OssClient::OSS_IF_MODIFIED_SINCE
If-Unmodified-Since	If the specified time is later than or equal to the time the object is modified, the object can be downloaded . Otherwise, an error (412 Precondition failed) is returned.	OssClient::OSS_IF_UNMODIFIED_SINCE
If-Match	If the specified ETag matches that of an object, the object can be downloaded. Otherwise, an error (412 Precondition failed) is returned.	OssClient::OSS_IF_MATCH
If-None-Match	If the specified ETag does not match that of an object, the object can be downloaded. Otherwise, an error (304 Not modified) is returned.	OssClient::OSS_IF_NONE_MATCH

**Note:**

- Both If-Modified-Since and If-Unmodified-Since can exist at the same time as object download conditions. Both If-Match and If-None-Match can exist at the same time as object download conditions.
- `$ossClient->getObjectMeta` can be used to obtain the ETag.

You can download an OSS object to local memories or a local file in conditional downloads. Run the following code for conditional download:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}
```

```

use OSS \ OssClient ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

try {
    $ options = array (
        OssClient :: OSS_HEADER_S => array (
            OssClient :: OSS_IF_MODIFIED_SINCE => " Fri , 13
Nov    2015    14 : 47 : 53    GMT " ),
    );

    $ ossClient = new    OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ content = $ ossClient -> getObject ( $ bucket , $ object , $
options );
} catch ( OssiException $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( $ content );
print ( __FUNCTION__ . " : OK " . "\ n " );

```

6.8 Manage objects

6.8.1 Overview

You can perform the following operations on objects in a bucket with APIs:

- [Determine whether an object exists](#)
- [Manage ACL for an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Delete an object](#)
- [Copy an object](#)
- [Restore an archive object](#)

- [Manage a symbolic link](#)
- [Enable MD5 verification](#)

6.8.2 Determine whether a specified object exists

Use the following code to determine whether a specified object exists:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

try {
    $ ossClient = new OssClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ exist = $ ossClient -> doesObject Exist ($ bucket , $ object
    );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . " : OK " . "\ n ");
var_dump ($ exist );
```

6.8.3 Manage ACL for an object

The following table describes the permissions included in the Access Control List (ACL) for an object.

Permission	Description	Value
default	The ACL of an object is the same with that of its bucket.	default
Private	Only the object owner and authorized users can read and write the object.	private
Public read	Only the object owner and authorized users can read and write the object. Other users can only read the object. Authorize this permission with caution.	public-read
Public read-write	All users can read and write the object. Authorize this permission with caution.	public-read-write

The ACL of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

Configure an ACL for an object

You can run the following code to configure an ACL for a specified object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM account , log on to https :// ram . console . aliyun
. com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
```

```
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
// Set the ACL for the object to public read , which
is inherited from the bucket by default .
$ acl = " public - read ";
try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    $ ossClient -> putObjectA cl ( $ bucket , $ object , $ acl );
} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n " );
```

Obtain the ACL for an object

You can run the following code to obtain the ACL for a specified object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssiExcepti on ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permission s on all APIs in OSS . We recommend that
you log on as a RAM user to access APIs or
perform routine operations and maintenanc e . To create
a RAM account , log on to https :// ram . console . aliyun
. com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
). Specify the actual endpoint based on your
requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    $ objectAcl = $ ossClient -> getObjectA cl ( $ bucket , $ object
);
} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
}
```

```
print ( __FUNCTION__ . ": OK " . "\n ");
var_dump ($ objectACL);
```

6.8.4 Manage Object Meta

For more information about Object Meta, see [Object Meta](#) in OSS Developer Guide.

Configure Object Meta

Run the following code to configure Object Meta:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ content = file_get_c ontents ( __FILE__ );
$ options = array (
    OsscClient :: OSS_HEADER S => array (
        ' Expires ' => ' 2012 - 10 - 01 08 : 00 : 00 ',
        ' Content - Dispositio n ' => ' attachment ; filename ="
xxxxxx "',
        ' x - oss - meta - self - define - title ' => ' user define
meta info ',
    ));
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    $ ossClient -> putObject ($ bucket , $ object , $ content , $
options );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
```

```
print ( __FUNCTION__ . ": OK " . "\n ");
```

Modify Object Meta

Run the following code to modify Object Meta:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ fromBucket = "< yourFromBu cketName >";
$ fromObject = "< yourFromOb jectName >";
$ toBucket = "< yourToBuck etName >";
$ toObject = "< yourToObje ctName >";
$ copyOption s = array (
    OsscClient :: OSS_HEADER S => array (
        ' Expires ' => ' 2018 - 10 - 01 08 : 00 : 00 ',
        ' Content - Dispositio n ' => ' attachment ; filename ="
xxxxxx "',
        ' x - oss - meta - location ' => ' location ',
    ),
);
try {
    $ ossClient = new OsscClient ($ accessKeyId , $ accessKeyS
ecret , $ endpoint );

    $ ossClient -> copyObject ($ fromBucket , $ fromObject , $
toBucket , $ toObject , $ copyOption s );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
```

Obtain Object Meta

Run the following code to obtain Object Meta:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
```

```

        require_once __DIR__ . '/../autoload.php';
    }
    if (is_file(__DIR__ . '/../vendor/autoload.php')) {
        require_once __DIR__ . '/../vendor/autoload.php';
    }

    use OSS \ OssClient ;
    use OSS \ Core \ OssException ;

    // It is highly risky to log on with AccessKey of
    // an Alibaba Cloud account because the account has
    // permission s on all APIs in OSS . We recommend that
    // you log on as a RAM user to access APIs or
    // perform routine operations and maintenance . To create
    // a RAM account , log on to https :// ram . console . aliyun
    // . com .
    $ accessKeyId = "< yourAccess KeyId >";
    $ accessKeySecret = "< yourAccess KeySecret >";
    // This example uses endpoint China East 1 ( Hangzhou
    // ). Specify the actual endpoint based on your
    // requirements .
    $ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
    $ bucket = "< yourBucket Name >";
    $ object = "< yourObject Name >";
    try {
        $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

        $ ObjectMeta = $ ossClient -> getObjectMeta ($ bucket , $
        object );
    } catch ( OssException $ e ) {
        printf ( __FUNCTION__ . " : FAILED \ n " );
        printf ($ e -> getMessage () . "\ n " );
        return ;
    }
    print ( __FUNCTION__ . " : OK " . "\ n " );
    if ( isset ($ objectMeta [ strtolower (' Content - Dispositio n
    ')])) &&
        ' attachment ; filename =" xxxxxx "' === $ objectMeta [
        strtolower (' Content - Dispositio n ')]
    ) {
        print ( __FUNCTION__ . " : ObjectMeta checked OK " . "\ n
        " );
    } else {
        print ( __FUNCTION__ . " : ObjectMeta checked FAILED " .
        "\ n " );
    }
}

```

6.8.5 List objects

List all objects in a bucket

Run the following code to list all objects in a bucket:

```

<? php
    if (is_file(__DIR__ . '/../autoload.php')) {
        require_once __DIR__ . '/../autoload.php';
    }
    if (is_file(__DIR__ . '/../vendor/autoload.php')) {
        require_once __DIR__ . '/../vendor/autoload.php';
    }

```

```

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ ossClient = new OssClient ($ accessKeyI d , $ accessKeyS ecret
, $ endpoint );

while ( true ) {
    try {
        $ listObject Info = $ ossClient -> listObject s ($ bucket ,
$ options );
    } catch ( OssExcepti on $ e ) {
        printf ( __FUNCTION__ . " : FAILED \ n " );
        printf ($ e -> getMessage () . "\ n " );
        return ;
    }
    // Obtain the nextMarker . Obtain the object list
    // from the object next to the last object read by
    // listObject s .
    $ nextMarker = $ listObject Info -> getNextMar ker ();
    $ listObject = $ listObject Info -> getObjectL ist ();
    $ listPrefix = $ listObject Info -> getPrefixL ist ();

    if ( ! empty ($ listObject )) {
        print ( " objectList : \ n " );
        foreach ($ listObject as $ objectInfo ) {
            print ($ objectInfo -> getKey () . "\ n " );
        }
    }
    if ( ! empty ($ listPrefix )) {
        print ( " prefixList : \ n " );
        foreach ($ listPrefix as $ prefixInfo ) {
            print ($ prefixInfo -> getPrefix () . "\ n " );
        }
    }
    if ( $ nextMarker === '' ) {
        break ;
    }
}

```

List objects with specified conditions

Run the following code to list objects with specified conditions:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}

```

```

if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS ecret
, $ endpoint );

$ prefix = ' dir /';
$ delimiter = '/';
$ nextMarker = '';
$ maxkeys = 10 ;
$ options = array (
    ' delimiter ' => $ delimiter ,
    ' prefix ' => $ prefix ,
    ' max - keys ' => $ maxkeys ,
    ' marker ' => $ nextMarker ,
);
try {
    $ listObject Info = $ ossClient -> listObject s ( $ bucket , $
options );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . "\ n " );
$ objectList = $ listObject Info -> getObjectL ist (); // object
list
$ prefixList = $ listObject Info -> getPrefixL ist (); //
directory list
if (! empty ( $ objectList )) {
    print ( " objectList : \ n " );
    foreach ( $ objectList as $ objectInfo ) {
        print ( $ objectInfo -> getKey () . "\ n " );
    }
}
if (! empty ( $ prefixList )) {
    print ( " prefixList : \ n " );
    foreach ( $ prefixList as $ prefixInfo ) {
        print ( $ prefixInfo -> getPrefix () . "\ n " );
    }
}
}

```

In the example above, \$options including the following parameters:

Parameter	Description	Required
delimiter	Specifies a delimiter used to group objects. CommonPrefixes indicates a set of objects which end with delimiter and have the same prefix.	No
prefix	Specifies the prefix you configure to list required objects.	No
max-keys	Specifies the maximum number of objects that can be listed. The default value is 100. The maximum value is 1,000.	No
marker	Specifies the initial object in the list.	No

6.8.6 Delete objects

Delete an object

Run the following code to delete a single object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

try {
```

```

    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ ossClient -> deleteObject ($ bucket , $ object );
} catch (OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");

```

Delete multiple objects

Run the following code to delete multiple objects simultaneously:

```

<? php
if ( is_file ( __DIR__ . '/../autoload . php ')) {
    require_once __DIR__ . '/../autoload . php ';
}
if ( is_file ( __DIR__ . '/../vendor / autoload . php ')) {
    require_once __DIR__ . '/../vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ objects = array ();
$ objects [] = "< yourObjectName1 >";
$ objects [] = "< yourObjectName2 >";
try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ ossClient -> deleteObjects ($ bucket , $ objects );
} catch (OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}

```

```
print ( __FUNCTION__ . ": OK " . "\n ");
```

6.8.7 Copy objects

Simple copy

Run the following code to copy objects smaller than 1 GB:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ from_bucke t = "< yourFromBu cketName >";
$ from_objec t = "< yourFromOb jectName >";
$ to_bucket = $ bucket ;
$ to_object = $ from_objec t . '. copy ';

try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> copyObject ($ from_bucke t , $ from_objec t , $
    to_bucket , $ to_object );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
```

Copy a large-sized object

To copy an object larger than 1 GB, you need to use multipart copy. To use multipart copy, perform the following steps:

1. Use `$ossClient->initiateMultipartUpload` to initiate an UploadPartCopy task.

2. Use `$ossClient->uploadPartCopy` to start a multipart copy. Aside from the last part, all parts must be larger than 100 KB.
3. Submit the multipart copy task with `$ossClient->completeMultipartUpload`.

Run the following code for multipart copy:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssiClient ;
use OSS \ Core \ OssiExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";

$ src_bucket = "< yourSource BucketName >";
$ src_object = "< yourSource ObjectName >";
$ dst_bucket = "< yourDestin ationBucke tName >";
$ dst_object = "< yourDestin ationObjec tName >";

try {
    $ ossClient = new OssiClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    // Initialize a multipart copy .
    $ upload_id = $ ossClient -> initiateMu ltipartUpl oad ($
    dst_bucket , $ dst_object );

    $ copyId = 1 ;
    // Copy the parts one - by - one .
    $ eTag = $ ossClient -> uploadPart Copy ( $ src_bucket , $
    src_object , $ dst_bucket , $ dst_object , $ copyId , $ upload_id );
    $ upload_par ts [] = array (
        ' PartNumber ' => $ copyId ,
        ' ETag ' => $ eTag ,
    );

    // Complete the multipart copy .
    $ result = $ ossClient -> completeMu ltipartUpl oad ($
    dst_bucket , $ dst_object , $ upload_id , $ upload_par ts );
} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
```

```
print ( __FUNCTION__ . ": OK " . "\n ");
```

6.8.8 Restore an archive object

You must restore an archive object before you read it. Do not call `restoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

1. An archive object is in the frozen state.
2. After you submit it for restoration, the server restores the object. The object is in the restoring state.
3. You can read the object after it is restored. The restored state of the object lasts one day by default. You can prolong this period to a maximum of seven days. Once this period expires, the object returns to the frozen state.

Run the following code to restore an object:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> restoreObj ect ($ bucket , $ object );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
```

```
print ( __FUNCTION__ . ": OK " . "\n ");
```

For more information about the archive storage classes, see [Introduction to storage classes](#).

6.8.9 Manage a symbolic link

Create a symbolic link

A symbolic link is a special object that maps to an object similar to a shortcut used in Windows. You can configure user-defined Object Meta for symbolic links.

Run the following code to create a symbolic link:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ symlink = "< yourSymlin k >";
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> putSymlink ($ bucket , $ symlink , $ object );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
```

Obtain object content for a symbolic link

To obtain a symbolic link, you must have the read permission on it. Run the following code to obtain the object content for a symbolic link:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console . aliyun
// . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ symlink = "< yourSymlin k >";
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ Symlinks = $ ossClient -> getSymlink ($ bucket , $ symlink );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");
var_dump ($ Symlinks );
```

6.8.10 Enable MD5 verification

MD5 verification is performed to ensure data integrity during transmission. MD5 verification brings performance loss. MD5 verification is disabled by default during file uploads.

Run the following code to enable MD5 verification during file uploads:

```
<? php
if ( is_file ( __DIR__ . '/../ Autopoload . php ')){
    require_on ce __DIR__ . '/../ autoload . php ';
```

```

}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

$ options = array ( OsscClient :: OSS_CHECK_ MD5 => true );
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> uploadFile ($ bucket , $ object , __FILE__ , $
    options );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");

```

The putObject, uploadFile, appendObject, appendFile and multiuploadFile methods support MD5 verification.

6.9 Authorized access

Sign a URL to authorize temporary access

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time for a URL to restrict the period of access from visitors. For the complete code of authorizing access, see [GitHub](#).

- Sign a URL to download an object

Run the following code to sign a URL to download an object:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {

```



```

        require_once __DIR__ . '/../vendor/autoload.php';
    }

    use OSS \ OssClient ;
    use OSS \ Core \ OssException ;
    use OSS \ Http \ RequestCore ;
    use OSS \ Http \ ResponseCore ;

    // It is highly risky to log on with AccessKey of
    // an Alibaba Cloud account because the account has
    // permission s on all APIs in OSS . We recommend that
    // you log on as a RAM user to access APIs or
    // perform routine operations and maintenance . To create
    // a RAM account , log on to https://ram.console.
    // aliyun.com .
    $ accessKeyId = "< yourAccess KeyId >";
    $ accessKeySecret = "< yourAccess KeySecret >";
    // This example uses endpoint China ( Hangzhou ). Specify
    // the actual endpoint based on your requirements .
    $ endpoint = "http://oss-cn-hangzhou.aliyuncs.com";
    $ bucket = "< yourBucket Name >";
    $ object = "< yourObject Name >";
    $ securityToken = "< yourSecurityToken >";

    // Set the expiration time of the URL to 3600
    // seconds .
    $ timeout = 3600 ;
    try {
        $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint , false , $ securityToken );

        // Generates a signed URL for access with
        // getObject .
        $ signedUrl = $ ossClient -> signUrl ($ bucket , $ object , $
        timeout );
    } catch ( OssException $ e ) {
        printf ( __FUNCTION__ . ": FAILED \n ");
        printf ($ e -> getMessage () . "\n ");
        return ;
    }
    print ( __FUNCTION__ . ": signedUrl : " . $ signedUrl . "\n
    ");

    // You can access a signed URL by running code or
    // input it in a browser .
    $ request = new RequestCore ($ signedUrl );
    // The signed URL is accessed with GET by default .
    $ request -> set_method (' GET ');
    $ request -> add_header (' Content - Type ', '');
    $ request -> send_request ();
    $ res = new ResponseCore ($ request -> get_response_header
    (), $ request -> get_response_body (), $ request -> get_response_code
    ());
    if ($ res -> isOK ()) {
        print ( __FUNCTION__ . ": OK " . "\n ");
    } else {
        print ( __FUNCTION__ . ": FAILED " . "\n ");
    }

```

```
};
```

- Sign a URL to upload a file

Run the following code to sign a URL to upload a file:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
use OSS \ Http \ RequestCor e ;
use OSS \ Http \ ResponseCo re ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all APIs in OSS . We recommend that
// you log on as a RAM user to access APIs or
// perform routine operations and maintenanc e . To create
// a RAM account , log on to https :// ram . console .
// aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ securityTo ken = "< yourSecuri tyToken >";

$ timeout = 3600 ;
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint , false , $ securityTo ken );

    // Generate a URL for access with PutObject .
    $ signedUrl = $ ossClient -> signUrl ($ bucket , $ object , $
    timeout , " PUT ");
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": signedUrl : " . $ signedUrl . "\ n
");

$ content = " Hello OSS .";
$ request = new RequestCor e ($ signedUrl );
// The signed URL is accessed with PUT .
$ request -> set_method (' PUT ');
$ request -> add_header (' Content - Type ', '');
$ request -> add_header (' Content - Length ', strlen ($ content
));
$ request -> set_body ($ content );
$ request -> send_reque st ();
$ res = new ResponseCo re ($ request -> get_respon se_header
()),
```

```

    $ request -> get_respon se_body (), $ request -> get_respon
se_code ());
    if ($ res -> isOK ()) {
        print ( __FUNCTION__ . ": OK " . "\ n ");
    } else {
        print ( __FUNCTION__ . ": FAILED " . "\ n ");
    }
};

```

Use STS for temporary access authorization

OSS supports Alibaba Cloud Security Token Service (STS) for temporary access authorization. For more information about STS, see [Introduction](#) of the access control API (STS) reference. For more information on accounts and authorization, see [STS temporary access authorization](#).

Run the following code to use STS for temporary file upload access authorization:

```

<? php
    if ( is_file ( __DIR__ . '/../ autoload . php ')) {
        require_on ce __DIR__ . '/../ autoload . php ';
    }
    if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
        require_on ce __DIR__ . '/../ vendor / autoload . php ';
    }

    use OSS \ OsscClient ;
    use OSS \ Core \ OssExcepti on ;

    // It is highly risky to log on with AccessKey of
    an Alibaba Cloud account because the account has
    permission s on all APIs in OSS . We recommend that
    you log on as a RAM user to access APIs or
    perform routine operations and maintenanc e . To create
    a RAM account , log on to https :// ram . console . aliyun
. com .
    $ accessKeyI d = "< yourAccess KeyId >";
    $ accessKeyS ecret = "< yourAccess KeySecret >";
    // This example uses endpoint China ( Hangzhou ). Specify
    the actual endpoint based on your requiremen ts .
    $ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
    $ bucket = "< yourBucket Name >";
    $ object = "< yourObject Name >";
    $ securityTo ken = "< yourSecuri tyToken >";

    $ content = " Hi , OSS .";

    try {
        $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
ecret , $ endpoint , false , $ securityTo ken );

        $ ossClient -> putObject ($ bucket , $ object , $ content );
    } catch ( OssExcepti on $ e ) {
        print $ e -> getMessage ();
    }

```

```
}
```

6.10 Static website hosting

You can set your bucket configuration to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

For more information about static website hosting, see [Static website hosting](#).

For the complete code of the following scenarios, see [GitHub](#).

Configure static website hosting

Run the following code to configure static website hosting:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
use OSS \ Model \ WebsiteCon fig ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

// Configure static website hosting : The index paga is
// index . html and the error page is error . html .
$ websiteCon fig = new WebsiteCon fig (" index . html ", " error
. html ");
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    $ ossClient -> putBucketW ebsite ($ bucket , $ websiteCon fig );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
```

```
print ( __FUNCTION__ . ": OK " . "\n");
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ websiteCon fig = null ;
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ websiteCon fig = $ ossClient -> getBucketW ebsite ($ bucket
    );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
print ($ websiteCon fig -> serializeT oXml () . "\n ");
```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
```

```
// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ ossClient -> deleteBucketWebsite ($ bucket );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
```

6.11 Manage lifecycle rules

You can use OSS to configure lifecycle rules to reduce storage costs. The lifecycle management rules can be used for automatic deletion of expired objects and fragments, or storage class conversion to Infrequent Access for objects that will expire soon. Each rule includes:

- **Rule ID:** It identifies a rule. Ensure that each rule ID in a bucket is unique.
- **Policy:** You can configure a policy by using the following configuration methods. Only one method can be configured for a bucket.
 - **Configure by Prefix:** You can create multiple rules with this method. Ensure that each prefix is unique.
 - **Configure for Entire Bucket:** You can configure only one rule with this method.
- **Expired:** You can configure the following configuration methods:
 - **Set by Number of Days:** It specifies the number of days from the last modification time of objects.
 - **Set by Date:** It specifies the date prior to which objects are created. If objects are created prior to the date, these objects are deleted.
- **Status:** This lifecycle rule takes effect or not.

For more information about lifecycle management, see [Manage object lifecycle](#). For the complete code of lifecycle management, see [GitHub](#).

Configure lifecycle rules

Run the following code to configure lifecycle rules:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
use OSS \ Model \ LifecycleC onfig ;
use OSS \ Model \ LifecycleR ule ;
use OSS \ Model \ LifecycleA ction ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

// Configure the rule ID and object name prefix .
$ ruleId0 = " rule0 ";
$ matchPrefi x0 = " A0 /";
$ ruleId1 = " rule1 ";
$ matchPrefi x1 = " A1 /";

$ lifecycleC onfig = new LifecycleC onfig ();
$ actions = array ();
// Set the expiration duration to 3 days after last
// modified date .
$ actions [] = new LifecycleA ction ( OsscClient :: OSS_LIFECY
    CLE_EXPIRA TION , OsscClient :: OSS_LIFECY CLE_TIMING _DAYS , 3
);
$ lifecycleR ule = new LifecycleR ule ($ ruleId0 , $ matchPrefi
    x0 , " Enabled ", $ actions );
$ lifecycleC onfig -> addRule ($ lifecycleR ule );
$ actions = array ();
// Configure expiration date to delete objects that are
// created before the date .
$ actions [] = new LifecycleA ction ( OsscClient :: OSS_LIFECY
    CLE_EXPIRA TION , OsscClient :: OSS_LIFECY CLE_TIMING _DATE , '
    2022 - 10 - 12T00 : 00 : 00 . 000Z ');
$ lifecycleR ule = new LifecycleR ule ($ ruleId1 , $ matchPrefi
    x1 , " Enabled ", $ actions );
$ lifecycleC onfig -> addRule ($ lifecycleR ule );
try {
```

```

    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ ossClient -> putBucketLifecycle ($ bucket , $ lifecycleConfig );
} catch (OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");

```

View lifecycle rules

Run the following code to view lifecycle rules:

```

<? php
if ( is_file ( __DIR__ . '/../autoload . php ')) {
    require_once __DIR__ . '/../autoload . php ';
}
if ( is_file ( __DIR__ . '/../vendor / autoload . php ')) {
    require_once __DIR__ . '/../vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ lifecycleConfig = null ;
try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    $ lifecycleConfig = $ ossClient -> getBucketLifecycle ($
    bucket );
} catch (OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
print ($ lifecycleConfig -> serializeToXml () . "\n ");

```

Clear lifecycle rules

Run the following code to clear lifecycle rules:

```

<? php

```



```

if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

try{
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> deleteBuck etLifecycl e ($ bucket );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");

```

6.12 Set logging

You can enable access logs to record bucket access to log files, which are stored in a specified bucket.

The log file format is as follows:

```

< TargetPref ix >< SourceBuck et >- YYYY - mm - DD - HH - MM - SS -
UniqueStri ng

```

For more information about access log files, see [Set access logging](#). For the complete code of access logging, see [GitHub](#).

Enable access logging

Run the following code to enable bucket access logging:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}

```

```

if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";

$ option = array ();
// Configure the bucket that stores log files .
$ targetBuck et = "< yourBucket Name >";
$ targetPref ix = " access . log ";
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    // Enable access logging .
    $ ossClient -> putBucketL ogging ($ bucket , $ targetBuck et , $
    targetPref ix , $ option );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");

```

View access logging configurations

Run the following code to view the access logging configurations for a bucket:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To

```

```

create a RAM account , log on to https :// ram . console
. aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ loggingCon fig = null ;
$ options = array ();
try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    // View access log configurat ions .
    $ loggingCon fig = $ ossClient -> getBucketL ogging ( $ bucket ,
$ options );
} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n " );
    printf ( $ e -> getMessage () . " \ n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . " \ n " );
print ( $ loggingCon fig -> serializeT oXml () . " \ n " );

```

Disable access logging

Run the following code to disable access logging for a bucket:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssiExcepti on ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
ecret , $ endpoint );

    // Disable access logging .
    $ ossClient -> deleteBuck etLogging ( $ bucket );
}

```

```

} catch ( OssiExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");

```

6.13 CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) in OSS Developer Guide. For the complete code of CORS, see [GitHub](#).

Configure CORS rules

Run the following code to configure CORS rules for the specified bucket:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssiClient ;
use OSS \ Core \ OssiExcepti on ;
use OSS \ Model \ CorsConfig ;
use OSS \ Model \ CorsRule ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ corsConfig = new CorsConfig ();
$ rule = new CorsRule ();
// AllowedHea ders and ExposeHead ers do not allow
// wildcards .
$ rule -> addAllowed Header ( " x - oss - header " );
// AllowedOlo wedMethods allows only one wildcard asterisk
// (*). Wildcard asterisks (*) indicate that all sources
// of the cross - origin requests and operations are
// allowed .

```

```

$ rule -> addAllowed Origin (" http :// www . b . com ");
$ rule -> addAllowed Method (" POST ");
$ rule -> setMaxAgeSeconds ( 10 );
// A maximum of 10 rules are allowed for each
bucket .
$ corsConfig -> addRule ( $ rule );

try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    // The existing rules will be replaced .
    $ ossClient -> putBucketCors ( $ bucket , $ corsConfig );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . " : FAILED \n " );
    printf ( $ e -> getMessage () . "\n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . "\n " );

```

Obtain CORS rules

Run the following code to obtain CORS rules:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccessKeyId >";
$ accessKeySecret = "< yourAccessKeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucketName >";

$ corsConfig = null ;
try {
    $ ossClient = new OssClient ( $ accessKeyId , $ accessKeySecret , $ endpoint );

    $ corsConfig = $ ossClient -> getBucketCors ( $ bucket );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . " : FAILED \n " );
    printf ( $ e -> getMessage () . "\n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . "\n " );

```

```
print ($ corsConfig -> serializeT oXml () . "\ n ");
```

Delete CORS rules

Run the following code to delete all CORS rules for a specified bucket:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> deleteBuck etCors ($ bucket );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");
```

6.14 Anti-leech

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leaching, see [Anti-leeching settings](#). For the complete code of anti-leech, see [GitHub](#).

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
use OSS \ Model \ RefererCon fig ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ refererCon fig = new RefererCon fig ();
// Allow empty referers .
$ refererCon fig -> setAllowEm ptyReferer ( true );
// Add the referer field . The referer field allows
// question marks (?) and asterisks (*) for wildcard use .
$ refererCon fig -> addReferer ( " www . aliiyun . com " );
$ refererCon fig -> addReferer ( " www . aliiyuncs . com " );
try {
    $ ossClient = new OsscClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ ossClient -> putBucketR eferer ( $ bucket , $ refererCon fig );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . "\ n " );
```

Obtain a referer whitelist

Run the following code to obtain a referer whitelist:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}
```

```

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
use OSS \ Model \ RefererCon fig ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

$ refererCon fig = null ;
try {
    $ ossClient = new OsscClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    $ refererCon fig = $ ossClient -> getBucketR eferer ($ bucket
    );
} catch ( OsscExcepti on $ e ) {
    printf ( __FUNCTION__ . " : FAILED \ n " );
    printf ($ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . " : OK " . "\ n " );
print ($ refererCon fig -> serializeT oXml () . "\ n " );

```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OsscClient ;
use OSS \ Core \ OsscExcepti on ;
use OSS \ Model \ RefererCon fig ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China ( Hangzhou ). Specify
// the actual endpoint based on your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

```



```

$ refererCon fig = new RefererCon fig ();
try {
    $ ossClient = new OssClient ($ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    // You cannot clear a referer whitelist directly . To
    clear a referer whitelist , you need to create the
    rule that allows an empty referer field and replace
    the original rule with the new rule .
    $ ossClient -> putBucketR eferer ($ bucket , $ refererCon fig );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n ");
    printf ($ e -> getMessage () . "\ n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n ");

```

6.15 Bind a custom domain name

You can bind a custom domain name to your OSS bucket and by adding a CNAME record in the Domain Name System (DNS). After you bind a custom domain name to a bucket, you can access the bucket with your own domain name. For example, if your domain name is "my-domain.com" and all your images were previously located at `http://img.my-domain.com/x.jpg`, after you migrate the image data to OSS, you can still access the images with the original IP address by binding a custom domain name to the bucket.

For more information, see [Bind a custom domain name](#) in the OSS Developer Guide.

Add a CNAME record

Run the following code to add a CNAME record for a bucket:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    Quire_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
an Alibaba Cloud account because the account has
permissions on all the APIs in OSS . We recommend
that you log on as a RAM user to access APIs
or perform routine operations and maintenanc e . To
create a RAM account , log on to https :// ram . console
. aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";

```

```
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

// Configure the custom domain name .
$ myDomain = '< yourDomain Name >';
try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    // Add a CNAME record .
    $ ossClient -> addBucketC name ( $ bucket , $ myDomain );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
print ( __FUNCTION__ . ": OK " . "\ n " );
```

View CNAME records

Run the following code to obtain CNAME records added to the bucket:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssExcepti on ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

try {
    $ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

    // View CNAME records .
    $ cnameConfi g = $ ossClient -> getBucketC name ( $ bucket );
} catch ( OssExcepti on $ e ) {
    printf ( __FUNCTION__ . ": FAILED \ n " );
    printf ( $ e -> getMessage () . "\ n " );
    return ;
}
```

```
var_dump ($ cnameConfig );
print ( __FUNCTION__ . ": OK " . "\n ");
```

Delete a CNAME record

Run the following code to delete a CNAME record added to a bucket:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;
use OSS \ Core \ OssException ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permissions on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenance . To
// create a RAM account , log on to https :// ram . console
// . aliyun . com .
$ accessKeyId = "< yourAccess KeyId >";
$ accessKeySecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";

// Configure the custom domain name .
$ myDomain = '< yourDomain Name >';
try {
    $ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

    // Delete a CNAME record .
    $ ossClient -> deleteBucketCname ($ bucket , $ myDomain );
} catch ( OssException $ e ) {
    printf ( __FUNCTION__ . ": FAILED \n ");
    printf ($ e -> getMessage () . "\n ");
    return ;
}
print ( __FUNCTION__ . ": OK " . "\n ");
```

6.16 Image processing

Image Processing (IMG) is a massive, secure, cost-effective and highly reliable image processing service. After source images are uploaded to OSS, you can process images on any Internet device at any anytime, from anywhere through simple RESTful APIs.

For more information about IMG, see [Image Processing](#). For the complete code of image processing, see [GitHub](#).

Basic features

OSS offers the following IMG features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Image processing](#)
- [Image processing access rules](#): calls multiple IMG functions.

Usage

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- [Anonymous access](#)

You can use the following format in level-3 domains to access a processed image:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >?  
>? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Specifies the name of a bucket.
endpoint	Specifies endpoint used to access a region.
object	Specifies the name of an image object.
image	Specifies the reserved identifier for IMG .
action	Specifies the operations on an image, such as scaling, cropping, and rotating.

Parameter	Description
param	Specifies the parameter that indicates the operation on an image.

- Basic operations

For example, scale an image to a width of 100 PX. Adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customized image styles

You can use the following format in level-3 domains to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

■ **style**: specifies a reserved identifier of a customized image style.

■ **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image sequentially.

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- HTTPS access

IMG supports access through HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

Run the following code to generate a signed URL for IMG:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;

// It is highly risky to log on with AccessKey of
// an Alibaba Cloud account because the account has
// permission s on all the APIs in OSS . We recommend
// that you log on as a RAM user to access APIs
// or perform routine operations and maintenanc e . To
// create a RAM account , log on to https:// ram .
// console . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 ( Hangzhou
// ). Specify the actual endpoint based on your
// requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";

$ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

// Generate a signed URL that is valid for 3600
// seconds and can be directly accessed by browsers .
$ timeout = 3600 ;
$ options = array (
    OossClient :: OSS_PROCES S => " image / resize , m_lfit ,
    h_100 , w_100 " );
$ signedUrl = $ ossClient -> signUrl ( $ bucket , $ object , $
    timeout , " GET " , $ options );
```

```
print (" rtmp   url : \ n " . $ signedUrl );
```

· Access with SDK

You can use SDK to access and process an image object.

SDK allows you to customize image styles, HTTPS access, and cascade operations.

- Basic operations

Run the following code to perform basic operations on an image:

```
<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;

// It is highly risky to log on with AccessKey
// of an Alibaba Cloud account because the account
// has permission s on all APIs in OSS . We
// recommend that you log on as a RAM user to
// access APIs or perform routine operations and
// maintenanc e . To create a RAM account , log on to
// https :// ram . console . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 (
// Hangzhou ). Specify the actual endpoint based on
// your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ download_f ile = " download . jpg ";

$ ossClient = new OssClient ( $ accessKeyI d , $ accessKeyS
    ecret , $ endpoint );

// Upload a sample image .
$ ossClient -> uploadFile ( $ bucket , $ object , " example . jpg
    " );

// Scale the image .
$ options = array (
    OossClient :: OSS_FILE_D OWNLOAD => $ download_f ile ,
    OossClient :: OSS_PROCES S => " image / resize , m_fixed ,
    h_100 , w_100 " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Crop the image .
$ options = array (
    OossClient :: OSS_FILE_D OWNLOAD => $ download_f ile ,
    OossClient :: OSS_PROCES S => " image / crop , w_100 , h_100
    , x_100 , y_100 , r_1 " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Rotate the image .
$ options = array (
```

```

        OssClient :: OSS_FILE_D  OWNLOAD => $ download_f ile ,
        OssClient :: OSS_PROCES S => " image / rotate , 90 " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Sharpen the image .
$ options = array (
    OssClient :: OSS_FILE_D  OWNLOAD => $ download_f ile ,
    OssClient :: OSS_PROCES S => " image / sharpen , 100 " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Add watermarks .
$ options = array (
    OssClient :: OSS_FILE_D  OWNLOAD => $ download_f ile ,
    OssClient :: OSS_PROCES S => " image / watermark ,
    text_SGVsb G8g5Zu - 54mH5pyN5Y qhIQ " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Convert the image format .
$ options = array (
    OssClient :: OSS_FILE_D  OWNLOAD => $ download_f ile ,
    OssClient :: OSS_PROCES S => " image / format , png " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Obtain image informatio n .
$ options = array (
    OssClient :: OSS_FILE_D  OWNLOAD => $ download_f ile ,
    OssClient :: OSS_PROCES S => " image / info " );
$ ossClient -> getObject ( $ bucket , $ object , $ options );

// Delete the sample image .
$ ossClient -> deleteObje ct ( $ bucket , $ object );

```

- Customize an image style

Run the following code to customize an image style:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_on ce __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_on ce __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;

// It is highly risky to log on with AccessKey
of an Alibaba Cloud account because the account
has permission s on all APIs in OSS . We
recommend that you log on as a RAM user to
access APIs or perform routine operations and
maintenanc e . To create a RAM account , log on to
https :// ram . console . aliyun . com .
$ accessKeyI d = "< yourAccess KeyId >";
$ accessKeyS ecret = "< yourAccess KeySecret >";
// This example uses endpoint China East 1 (
Hangzhou ). Specify the actual endpoint based on
your requiremen ts .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucket Name >";
$ object = "< yourObject Name >";
$ download_f ile = " download . jpg ";

```



```

$ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint );

// Upload a sample image .
$ ossClient -> uploadFile ($ bucket , $ object , " example . jpg
");

// Customize an image style
$ style = " style / oss - pic - style - w - 300 ";
$ options = array (
    OssClient :: OSS_FILE_DOWNLOAD => $ download_file ,
    OssClient :: OSS_PROCESS => $ style );
$ ossClient -> getObject ($ bucket , $ object , $ options );

// Delete the sample image .
$ ossClient -> deleteObject ($ bucket , $ object );

```

- Cascade operations

Use the following code to perform cascade operations on an image:

```

<? php
if ( is_file ( __DIR__ . '/../ autoload . php ')) {
    require_once __DIR__ . '/../ autoload . php ';
}
if ( is_file ( __DIR__ . '/../ vendor / autoload . php ')) {
    require_once __DIR__ . '/../ vendor / autoload . php ';
}

use OSS \ OssClient ;

// It is highly risky to log on with AccessKey
// of an Alibaba Cloud account because the account
// has permissions on all APIs in OSS . We
// recommend that you log on as a RAM user to
// access APIs or perform routine operations and
// maintenance . To create a RAM account , log on to
// https :// ram . console . aliyun . com .
$ accessKeyId = "< yourAccessKeyId >";
$ accessKeySecret = "< yourAccessKeySecret >";
// This example uses endpoint China East 1 (
// Hangzhou ) . Specify the actual endpoint based on
// your requirements .
$ endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";
$ bucket = "< yourBucketName >";
$ object = "< yourObjectName >";
$ download_file = " download . jpg ";

$ ossClient = new OssClient ($ accessKeyId , $ accessKeySecret , $ endpoint , false );

// Upload a sample image .
$ ossClient -> uploadFile ($ bucket , $ object , " example . jpg
");

// Perform cascade operations .
$ style = " image / resize , m_fixed , w_100 , h_100 / rotate ,
90 ";
$ options = array (
    OssClient :: OSS_FILE_DOWNLOAD => $ download_file ,
    OssClient :: OSS_PROCESS => $ style );
$ ossClient -> getObject ($ bucket , $ object , $ options );

```

```
// Delete the sample image .  
$ ossClient -> deleteObject ($ bucket , $ object );
```

IMG tools

You can use the IMG viewer [ImageStyleViewer](#) to directly view the IMG result.

6.17 Exception handling

OSS PHP SDK exceptions (OssException) include various errors, such as invalid parameters and file missing. You can use the getMessage method to obtain error messages.

For more information about OssException, see [GitHub](#).

Example for handling exceptions

The following code shows how to handle exceptions when creating a existing bucket and prints error messages.

```
try {  
    $ ossClient -> createBucket ($ bucket );  
} catch ( OssException $ e ) {  
    print ( " Exception : " . $ e -> getMessage () . "\ n " );  
}
```

You can also obtain the following information:

Parameter	Description
HTTPStatus	HTTP status code. You can get the status code using the getHTTPStatus method.
ErrorCode	Specifies the error code returned by OSS . You can get the error code using the getErrorCode method.
ErrorMessage	Specifies the error messages returned by OSS. You can get the error messages using the getErrorMessage method.
RequestId	Specifies the UUID. It is unique and used to identify a request. If a problem persists, send the RequestId to OSS developers for help. You can get the RequestId using the getRequestId method.

Parameter	Description
Details	Specifies detailed error messages returned by OSS. You can get the detailed error messages using the <code>getDetails</code> method.

Common OSS error codes

Error code	Description	HTTP status code
AccessDenied	Access denied	403
BucketAlreadyExists	The bucket already exists.	409
BucketNotEmpty	The bucket is not empty.	409
EntityTooLarge	The entity size exceeds the maximum limit.	400
EntityTooSmall	The entity size is below the minimum limit.	400
FileGroupTooLarge	The total file group size exceeds the maximum limit.	400
FilePartNotExist	No such part exists.	400
FilePartStale	The part has expired.	400
InvalidArgument	The parameter format is invalid.	400
InvalidAccessKeyId	No such AccessKeyId exists.	403
InvalidBucketName	The bucket name is invalid.	400
InvalidDigest	The digest is invalid.	400
InvalidObjectName	The object name is invalid.	400
InvalidPart	The part is invalid.	400
InvalidPartOrder	The part order is invalid.	400
InvalidTargetBucketForLogging	The buckets that store log files are invalid.	400
InternalError	Internal OSS error	500
MalformedXML	The XML format is invalid.	400

Error code	Description	HTTP status code
MethodNotAllowed	The method is not allowed.	405
MissingArgument	Parameters are not configured.	411
MissingContentLength	The content length is not configured.	411
NoSuchBucket	No such bucket exists.	404
NoSuchKey	No such object exists.	404
NoSuchUpload	No such uploadId exists.	404
NotImplemented	The methods are not implemented.	501
PreconditionFailed	Precondition error	412
RequestTimeTooSkewed	The local time set for OSSClient is deviated from the time set for the OSS server by over 15 minutes.	403
RequestTimeout	Request timeout	400
SignatureDoesNotMatch	Signature mismatch	403
InvalidEncryptionAlgorithmError	The specified encryption algorithm (the entropy encoding type) is invalid.	400

7 C

7.1 Preface

This document is written based on OSS C SDK 3.5.0.

Compatibility

- For SDK series of 3.*: compatible
- For SDK series of 2.*:
 - Windows: compatible
 - Linux API: compatible except for the linked list (`aos_list_t`) traversal interface.
 - `os_list_for_each_entry`
 - `aos_list_for_each_entry_reverse`
 - `aos_list_for_each_entry_safe`
 - `aos_list_for_each_entry_safe_reverse`
- For SDK series of 1.0.0: incompatible for only the following APIs
 - `oss_request_options_t`
 - `oss_get_object_to_buffer`
 - `oss_get_object_to_file`
 - `oss_get_object_to_buffer_by_url`
 - `oss_get_object_to_file_by_url`
 - `oss_init_multipart_upload`
 - `oss_complete_multipart_upload`
- For SDK series of 0.0.*: incompatible

SDK source code

For the Java SDK source code, see [GitHub](#).

Code samples

OSS SDK for C provides a variety of code samples for your reference or use. The following table describes the content of code samples:

Sample file	Content
oss_put_object_sample	Upload objects
oss_get_object_sample.c	Download objects
oss_append_object_sample.c	Append upload
oss_multipart_upload_sample.c	Multipart upload
oss_resumable_sample.c	Resumable upload, Resumable download
oss_head_object_sample.c	Manage the metadata of an object
oss_list_object_sample.c	List objects
oss_delete_object_sample.c	Delete an object
oss_callback_sample.c	Upload callback
oss_progress_sample.c	Progress bar, Progress bar
oss_crc_sample.c	CRC check during upload or download
oss_image_sample.c	Image processing

7.2 Installation

Install OSS C SDK in Linux

- Install CMake and third-party libraries

When installing OSS C SDK in Linux, you must install CMake and the following third-party libraries: curl, apr, apr-util and minixml.

The following parameters are required to install the environment:

Parameter	Description	Version requirement
CMake	Build and installation tool	Version 2.6.0 and later
curl	Focuses on network problems.	Version 7.32.0 and later
apr-util	Focuses on memory management and cross-platform operations.	Version 1.5.2 and later

Parameter	Description	Version requirement
minixml	Parses XML returned by a request.	We recommend you use version 2.9.

Select the operation system that you need to install CMake and third-party libraries

.

- Ubuntu/Debian

■ Install CMake

```
sudo apt - get install cmake
```

■ Install third-party libraries

```
sudo apt - get install libcurl4 - openssl - dev  
libapr1 - dev libaprutil 1 - dev libmxml - dev
```

If the mxml installation package is not included in the yum source, you can install the libraries using RPM. Download the rpm installation package based on your operation system:

■ 64-bit: [mxml-2.9-1.x86_64.rpm](#)

■ 32-bit: [mxml-2.9-1.i386.rpm](#)

Run the following code to install mxml with the rpm installation package:

```
rpm - ivh mxml - 2 . 9 - 1 . x86_64 . rpm
```

- RedHat/Aliyun/CentOS

■ Install CMake

```
sudo yum install cmake
```

■ Install third-party libraries

```
sudo yum install curl - devel apr - devel apr - util  
- devel mxml mxml - devel
```

- SuSE

■ Install CMake

```
zypper install cmake
```

■ Install third-party libraries

```
zypper install libcurl - devel libapr1 - devel libapr  
- util1 - devel mxml - devel
```

- Other Linux

■ Install CMake: [Download link](#)

The common installation method is as follows:

```
./ configure  
make  
make install
```



Note:

When running `./configure`, the default installation path is `/usr/local/`. If you want to specify another path, use the `./configure --prefix` option.

■ Install libcurl: [Download link](#)

The common installation method is as follows:

```
./ configure  
make  
make install
```

■ Install apr: [Download link](#)

The common installation method is as follows:

```
./ configure  
make  
make install
```

■ Install apr-util: [Download link](#)

The common installation method is as follows:

```
// You must specify the --with-apr option in  
the installation .  
./ configure --with-apr=/your/apr/install/path  
make  
make install
```

■ Install minixml: [Download link](#)

The common installation method is as follows:

```
./ configure
```



```
make
sudo make install
```

- Download SDK
 - [Download SDK directly](#)
 - [Download SDK from GitHub](#)
 - [Download historical SDK](#)
- Install SDK

Use `cmake`. If third-party libraries, such as `url`, `apr`, `apr-util` and `mxml` are not installed in the default method during the compiling process, you must specify the path of their header files and library files.

Install the SDK as follows:

```
cmake .
make
make install
```

An installation example in CentOS is as follows:

```
cmake -f CMakeLists.txt
// The compilation type is Release. Common
compilation types include: Debug, Release, RelWithDeb
Info, and MinSizeRel, in which Debug is the default
type.
- DCMAKE_BUI LD_TYPE = Release
// Customize the installation directory.
- DCMAKE_INS TALL_PREFI X =/usr/local/
// Specify the path where the header files and
library files of third party libraries, such as curl
, apr, apr-util, and xml, are located,
- DCURL_INCL UDE_DIR =/usr/include/curl
- DCURL_LIBR ARY =/usr/lib64/libcurl.so
- DAPR_INCLU DE_DIR =/usr/include/apr-1
- DAPR_LIBRA RY =/usr/lib64/libapr-1.so
- DAPR_UTIL_ INCLUDE_DI R =/usr/include/apr-1
- DAPR_UTIL_ LIBRARY =/usr/lib64/libaprutil-1.so
- DMINIXML_I NCLUDE_DIR =/usr/include
- DMINIXML_L IBRARY =/usr/lib64/libmxml.so
// If the "Could not find apr-config/apr-1-config
" error occurs during the compiling process, it
indicates that the apr-1-config file cannot be
found in the default path. Add this option.
- DAPR_CONFI G_BIN =/path/to/bin/apr-1-config
// If the "Could not find apu-config/apu-1-config
" error occurs, add this option.
```

```
- DAPU_CONFI  G_BIN =/ path / to / bin / apu - 1 - config
```

Install OSS C SDK in Windows

- Download SDK
 - [Download SDK directly](#)
 - [Download SDK from GitHub](#)
 - [Download historical SDK](#)
- Install SDK

For detailed steps and common problems when using Visual Studio to compile OSS C SDK, see [Compile and Use Alibaba Cloud OSS C SDK in Windows](#).



Note:

If you use Visual Studio 2012 and later versions, it prompts you to upgrade the project to the latest compiler and libraries. We recommended you to determine whether to upgrade based on your project. You can upgrade if your project uses the latest compiler and libraries.

7.3 Quick start

This topic describes how to use OSS C SDK to complete routine operations such as bucket creation, objects uploads, and object downloads.

Demo

- Linux demo

Linux-based demo: [aliyun-oss-c-sdk-demo.tar.gz](#).

1. Download and extract the demo project.
2. You must specify the installation directory for oss-c-sdk-demo-specified-installation: `/ home / your / oss / csdk` . You do not need to specify installation directories for other demo projects because they are automatically installed based on the OSS C SDK and other dependent third-party libraries.
3. Replace the `OSS_ENDPOINT`, `ACCESS_KEY_ID`, `ACCESS_KEY_SECRET`, and `BUCKET_NAME` in the demo with valid values. Use `LD_LIBRARY _PATH`

to specify the directory of the OSS C SDK and the dynamic libraries of the dependent libraries if they are not in the system directory.

4. Enter the directory of the project (oss-c-sdk-demo-xxx), run the `make` command to compile the demo project. Then, run the `./ main` command to run the executable program. If the project needs to be re-compiled, run the `make clean` command first before compiling.

- Windows demo

Windows-based demo: [aliyun-oss-c-sdk-sample.zip](#). You can download more demo projects from [GitHub](#).

1. Download and extract the demo project.
2. Use Visual Studio to open the demo project, and then replace the following parameters in the `oss_config . c` with valid values: `OSS_ENDPOINT`, `ACCESS_KEY_ID`, `ACCESS_KEY_SECRET`, `BUCKET_NAME`, `OBJECT_NAME`, `MULTIPART_UPLOAD_FILE_PATH`, and `DIR_NAME`. Compile and run the project. You can develop your own project based on the demo project.

Create a bucket

A bucket is a global namespace on OSS used to store objects. It is equivalent to a data container. You can run the following code to create a bucket:

```
# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess ` KeyId >";
const char * access_key _secret = "< yourAccess ` KeySecret >";
const char * bucket_nam e = "< yourBucket ` Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
```

```

/* Call the aos_http_initialize method in main
() to initialize global resources, such as the network
and memory. */
if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
    exit ( 1 );
}
/* Initialize a memory pool used to manage memories
, which is equivalent to apr_pool_t. The implementa
tion code is included in the apr library. */
aos_pool_t * pool ;
/* Re - create a new memory pool. The second
parameter is NULL, indicating that it does not
inherit from any other memory pools. */
aos_pool_create ( & pool , NULL );
/* Create and initialize options. This parameter
mainly includes global configuration information, such
as endpoint, access_key_id, access_key_secret, is_cname
, and curl. */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to options
. */
oss_client_options = oss_request_options_create ( pool );
/* Use oss_client_options to initialize client
options */
init_options ( oss_client_options );
/* Create the bucket. */
aos_status_t * resp_status ;
aos_table_t * resp_headers ;
oss_acl_t oss_acl = OSS_ACL_PRIVATE ;
aos_string_t bucket ;
/* Assign a char * to the bucket. */
aos_str_set ( & bucket , bucket_name );
resp_status = oss_create_bucket ( oss_client_options , &
bucket , oss_acl , & resp_headers );
/* Identify whether the request succeeds */
if ( aos_status_is_ok ( resp_status ) ) {
    printf ( " create bucket succeeded \n " );
} else {
    printf ( " create bucket failed \n " );
}
/* Release the memory pool, that is, memories
allocated to resources during the request. */
aos_pool_destroy ( pool );
/* Release allocated global resources. */
aos_http_initialize ();
return 0 ;
}

```

For more information about endpoints, see [Regions and endpoints](#).

Upload an object

You can run the following code to upload an object to OSS:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options ( oss_request_options_t * options )

```

```

{
    options -> config = oss_config_create ( options -> pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_str_set (& options -> config -> endpoint , endpoint );
    aos_str_set (& options -> config -> access_key_id ,
access_key_id );
    aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout */
    options -> ctl = aos_http_controller_create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) { =
AOSE_OK ) {
        exit ( 1 );
    }
    /* Initialize a memory pool used to manage memories
    , which is equivalent to apr_pool_t . The implementa
    tion code is included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pool . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configuration information , such
    as endpoint , access_key_id , access_key_secret , is_cname
    , and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memory in the memory pool to options .
    */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
    options */
    init_options ( oss_client_options );
    /* Initialization parameters */
    aos_string_t bucket ;
    aos_string_t object ;
    aos_status_t * resp_status ;
    aos_table_t * headers ;
    aos_table_t * resp_headers ;
    aos_list_t buffer ;
    aos_buf_t * content = NULL ;
    const char * data = " More than just cloud .";
    aos_str_set (& bucket , bucket_name );
    aos_str_set (& object , object_name );
    headers = aos_table_make ( pool , 0 );
    /* Convert the char * type data to the aos_list_t
    type data . */
    aos_list_init (& buffer );
    content = aos_buf_pack ( oss_client_options -> pool , data
, strlen ( data ));
    aos_list_add_tail (& content -> node , & buffer );
    /* Upload an object . */

```

```

    resp_status = oss_put_object_from_buffer ( oss_client_options , & bucket , & object , & buffer , headers , & resp_headers );
    /* Determine whether the upload request is successful . */
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " put file succeeded \n " );
    } else {
        printf ( " put file failed \n " );
    }
    /* Release the memory pool , that is , memories allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

Download an object

You can run the following code to download a specified object:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * download_filename = "< yourDownloadedFilename >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t . */
    aos_str_set ( & options->config->endpoint , endpoint );
    aos_str_set ( & options->config->access_key_id , access_key_id );
    aos_str_set ( & options->config->access_key_secret , access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as timeout */
    options->ctl = aos_http_controller_create ( options->pool , 0 );
}
int main ( int argc , char * argv [] )
{
    /* Call the aos_http_initialize method in main () to initialize global resources , such as networks and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Initialize a memory pool used to manage memories , which is equivalent to apr_pool_t . The implementation code is included in the apr library . */
    aos_pool_t * pool ;

```

```

/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_t create (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configuration information , such
as endpoint , access_key_id , access_key_secret , is_cname
, and curl . */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to options
. */
oss_client_options = oss_request_options_create ( pool );
/* Use oss_client_options to initialize client
options */
init_options ( oss_client_options );
/* Initialization parameters */
aos_string_t bucket ;
aos_string_t object ;
aos_string_t file ;
aos_status_t * resp_status ;
aos_table_t * headers = NULL ;
aos_table_t * params = NULL ;
aos_table_t * resp_headers = NULL ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );
aos_str_set (& file , download_filename );
headers = aos_table_make ( pool , 0 );
/* Download an object . */
resp_status = oss_get_object_to_file ( oss_client_options , & bucket , & object , headers , params , & file , & resp_headers );
/* Determine whether the download request is
successful . */
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get object to local file succeeded \n "
);
} else {
    printf ( " get object to local file failed \n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_info_deinitialize ();
return 0 ;
}

```

List objects in a bucket

You can run the following code to list objects stored in a specified bucket.

```

# include " oss_api . h "
# include " aos_http_info . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
}

```

```

    aos_str_set (& options -> config -> endpoint , endpoint );
    aos_str_set (& options -> config -> access_key _id ,
access_key _id );
    aos_str_set (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initia li ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initia li ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Initialize a memory pool used to manage memories
, which is equivalent to apr_pool_t . The implementa
tion code is included in the apr library . */
    aos_pool_t * pool = NULL ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pool . */ /*
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options = NULL ;
    /* Allocate memory in the memory pool to options .
*/
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters */
    aos_string _t bucket ;
    aos_status _t * resp_statu s = NULL ;
    oss_list_o bject_para ms_t * params = NULL ;
    oss_list_o bject_cont ent_t * content = NULL ;
    int size = 0 ;
    char * line = NULL ;
    aos_str_set (& bucket , bucket_nam e );
    params = oss_create _list_obje ct_params ( pool );
    /* List objects */
    resp_statu s = oss_list_o bject ( oss_client _options , &
bucket , params , NULL );
    /* Determine whether the list request is successful .
*/
    if (! aos_status _is_ok ( resp_statu s ))
    {
        printf (" list object failed \ n ");
    }
    else {
        /* Print objects . */
        printf (" Object \ tSize \ tLastModif ied \ n ");
        aos_list_f or_each_en try ( oss_list_o bject_cont ent_t
, content , & params -> object_lis t , node ) {
            ++ size ;

```



```

        line = apr_psprintf ( pool , "%.*s \t %.*s \t %.*s \n ", content -> key . len , content -> key . data ,
                                content -> size . len , content -> size . data ,
                                content -> last_modified . len , content ->
last_modified . data );
        printf ("%s ", line );
    }
    printf (" Total %d \n ", size );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_io_initialize ();
return 0 ;
}

```

Delete an object.

You can run the following code to delete an object:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options ( oss_request_options_t * options )
{
    options -> config = oss_config_create ( options -> pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_string_t (& options -> config -> endpoint , endpoint );
    aos_string_t (& options -> config -> access_key_id ,
access_key_id );
    aos_string_t (& options -> config -> access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout */
    options -> ctl = aos_http_controller_create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_io_initialize method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_io_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Initialize a memory pool used to manage memories
, which is equivalent to apr_pool_t . The implementa
tion code is included in the apr library . */
    aos_pool_t * pool = NULL ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pool . */
    aos_pool_create (& pool , NULL );
}

```

```

/* Create and initialize options. This parameter
mainly includes global configuration information, such
as endpoint, access_key_id, access_key_secret, is_cname
, and curl. */
oss_request_options_t *oss_client_options = NULL;
/* Allocate memory in the memory pool to options.
*/
oss_client_options = oss_request_options_create(pool);
/* Use oss_client_options to initialize client
options */
init_options(oss_client_options);
/* Initialization parameters */
aos_string_t bucket;
aos_string_t object;
aos_status_t *resp_status = NULL;
aos_table_t *resp_headers = NULL;
aos_str_set(&bucket, bucket_name);
aos_str_set(&object, object_name);
/* Delete an object. */
resp_status = oss_delete_object(oss_client_options, &
bucket, &object, &resp_headers);
/* Determine whether the delete request is successful
. */
if (aos_status_is_ok(resp_status)) {
    printf("delete object succeeded \n");
} else {
    printf("delete object failed \n");
}
/* Release the memory pool, that is, memories
allocated to resources during the request. */
aos_pool_destroy(pool);
/* Release allocated global resources. */
aos_http_info_deinitialize();
return 0;
}

```

7.4 Initialization

To use OSS C SDK, you must initialize request options (`oss_request_options_t`) and specify an endpoint. For more information about endpoint, see [Regions and endpoints](#) and [Bind a custom domain name](#).

Use the OSS domain name to initialize request options.

You can run the following code to initialize request options with the OSS domain name:

```

# include "oss_api.h"
# include "aos_http_info.h"
const char *endpoint = "< yourEndpoint >";
const char *access_key_id = "< yourAccessKeyId >";
const char *access_key_secret = "< yourAccessKeySecret >";
void init_options(oss_request_options_t *options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize the aos_string
    _t type. */
    aos_str_set(&options->config->endpoint, endpoint);
}

```

```

    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main () {
    aos_pool_t * p ;
    oss_reques t_options_ t * options ;
    /* Initialize global variables , which are run first
before the program starts . */
    if ( aos_http_i o_initia lize ( NULL , 0 ) != AOSE_OK ) {
        return - 1 ;
    }
    /* Initialize the memory pool and options . */
    aos_pool_c reate (& p , NULL );
    options = oss_reques t_options_ create ( p );
    init_optio ns ( options );
    /* Logical code , which is not listed here . */
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( p );
    /* Release the allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

**Note:**

For the configuration of request options, see related documents. <http://bbs.aliyun.com/read/261421.html?spm=5176.bbsr248504.0.0.NoFhRX>

Use a custom domain name to initialize request options.

You can run the following code to initialize request options with a domain name:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourCustom Endpoint >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
void init_optio ns ( oss_reques t_options_ t * options ) {
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize the aos_string
_t type . */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Enable the CNAME and bind the custom domain
name to the bucket . */
    options -> config -> is_cname = 1 ;
}

```

```

    options->ctl = aos_http_c ontroller_ create ( options->
pool , 0 );
}
int main () {
    aos_pool_t * p ;
    oss_reques t_options_ t * options ;
    /* Initialize global variables , which are run before
the program starts . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        return - 1 ;
    }
    /* Initialize the memory pool and options . */
    aos_pool_c reate (& p , NULL );
    options = oss_reques t_options_ create ( p );
    init_optio ns ( options );
    /* Logical code , which is not listed here . */
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( p );
    /* Release the allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Use STS to initialize request options

You can run the following code to initialize request options with STS:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * sts_token = "< yourSecuri tyToken >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options->config = oss_config _create ( options->pool );
    /* Use a char * string to initialize the aos_string
_t type . */
    aos_str_se t (& options->config->endpoint , endpoint );
    aos_str_se t (& options->config->access_key _id ,
access_key _id );
    aos_str_se t (& options->config->access_key _secret ,
access_key _secret );
    /* Configure STS */
    aos_str_se t (& options->config->sts_token , sts_token );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options->config->is_cname = 0 ;
    options->ctl = aos_http_c ontroller_ create ( options->
pool , 0 );
}
int main () {
    aos_pool_t * p ;
    oss_reques t_options_ t * options ;
    /* Initialize global variables , which are run first
before the program starts . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        return - 1 ;
    }
    /* Initialize the memory pool and options . */
    aos_pool_c reate (& p , NULL );
}

```

```

    options = oss_reques t_options_ create ( p );
    init_optio ns ( options );
    /* Logical code , which is not listed here . */
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( p );
    /* Release the allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Set the timeout value.

You can run the following code to set the timeout value:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize the aos_string
    _t type . */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
    access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
    access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout */
    options -> ctl = aos_http_c ontroller_ create ( options ->
    pool , 0 );
    /* Set the connection timeout value , which is 10
    seconds by default . */
    options -> ctl -> options -> connect_ti meout = 10 ;
    /* Set the DNS timeout value , which is 60 seconds
    by default . */
    options -> ctl -> options -> dns_cache_ timeout = 60 ;
    /* Set the request timeout value :
    Set the value of speed_limi t to specify the
    minimum speed that can be accepted , which is 1 , 024
    Bytes / s by default .
    Set the value of speed_time to specify the
    maximum time period that can be accepted , which is
    15 seconds by default .
    The following code indicates that if the
    trasmissio n speed is continuously lower than 1 , 024
    Bytes / s for 15 seconds , the request is time - out .
    */
    options -> ctl -> options -> speed_limi t = 1024 ;
    options -> ctl -> options -> speed_time = 15 ;
}

```

```
}
```

7.5 Bucket

A bucket serves as a container that stores objects. Objects belong to a bucket.

Create a bucket

Run the following code to create a bucket:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t. */
    aos_string_t endpoint_str;
    aos_string_t access_key_id_str;
    aos_string_t access_key_secret_str;
    aos_string_t bucket_name_str;
    /* Determine whether the CNAME is used. 0 indicates
       that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as
       timeout. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main()
       to initialize global resources, such as networks
       and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, which is
       equivalent to apr_pool_t. The implementation code is
       included in the apr library. */
    aos_pool_t * pool;
    /* Re-create a new memory pool. The second
       parameter is NULL, indicating that it does not
       inherit from any other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter
       mainly includes global configuration information, such
       as endpoint, access_key_id, access_key_secret, is_cname,
       and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Use oss_client_options to initialize client
       options. */
```

```

init_options ( oss_client_options );
/* Initialization parameters . */
aos_string_t bucket ;
oss_acl_t oss_acl = OSS_ACL_PRIVATE ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
/* Assign a char * to the aos_string_t bucket . */
aos_str_set (& bucket , bucket_name );
/* Create a bucket . */
resp_status = oss_create_bucket ( oss_client_options , &
bucket , oss_acl , & resp_headers );
/* Determine whether the request is successful . */
if ( aos_status_is_ok ( resp_status )) {
    printf ( " create bucket succeeded \n " );
} else {
    printf ( " create bucket failed \n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

For more information about bucket naming rules, see naming conventions in [Basic concepts](#). You can specify the ACL and the storage class when you create a bucket, as shown in [Bucket-level permissions](#) and [Storage class](#).

You can run the following code to create a bucket with a specified ACL:

```

/* Create a bucket and set the bucket ACL to
public read ( which is private by default ). */
resp_status = oss_create_bucket ( oss_client_options , &
bucket , OSS_ACL_PUBLIC_READ , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " create bucket succeeded \n " );
} else {
    Printf ( " Create Bucket failed \n " );
}

```

You can run the following code to creates a bucket of the archive class:

```

resp_status = oss_create_bucket_with_storage_class (
oss_client_options , & bucket , OSS_ACL_PRIVATE , OSS_STORAGE_
CLASS_ARCHIVE );
/* Determine whether the request is successful . */
if ( aos_status_is_ok ( resp_status )) {
    printf ( " create bucket succeeded \n " );
} else {
    printf ( " create bucket failed \n " );
}

```

You can run the followign code to create a bucket of the Infrequent Access class:

```

resp_status = oss_create_bucket_with_storage_class (
oss_client_options , & bucket , OSS_ACL_PRIVATE , OSS_STORAGE_
CLASS_IA );
/* Determine whether the request is successful . */

```

```

        if ( aos_status_is_ok ( resp_status )) {
            printf ( " create    bucket    succeeded \ n ");
        } else {
            printf ( " create    bucket    failed \ n ");
        }
    }
}

```

List buckets

Run the following code to list all buckets:

```

# include " oss_api . h "
# include " aos_http_interface . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize the aos_string_t type . */
    aos_string_t (& options->config->endpoint , endpoint );
    aos_string_t (& options->config->access_key_id , access_key_id );
    aos_string_t (& options->config->access_key_secret , access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as timeout .
    options->ctl = aos_http_controller_create ( options->pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_interface_initialize method in main () to initialize global resources , such as networks and memories . */
    if ( aos_http_interface_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is equivalent to apr_pool_t . The implementation code is included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second parameter is NULL , indicating that it does not inherit from any other memory pools . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter mainly includes global configuration information , such as endpoint , access_key_id , access_key_secret , is_cname , and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options . */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client options . */
    init_options ( oss_client_options );
}

```



```

/* Initialization parameters . */
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
oss_list_buckets_params_t * params = NULL ;
oss_list_bucket_content_t * content = NULL ;
int size = 0 ;
params = oss_create_list_buckets_params ( pool );
/* List all buckets . */
resp_status = oss_list_bucket ( oss_client_options ,
params , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " list buckets succeeded \n " );
} else {
    printf ( " list buckets failed \n " );
}
/* Print bucket information . */
aos_list_for_each_entry ( oss_list_bucket_content_t ,
content , & params-> bucket_list_t , node ) {
    printf ( " BucketName : %s \n " , content-> name . data );
    ++ size ;
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_info_deinitialize ();
return 0 ;
}

```

Configure an ACL for a bucket

The ACL of a bucket includes the following permissions.

Permission	Description	Value
Private	The bucket owner and the authorized users can read and write objects in the bucket. Other users cannot perform any operation on the objects.	OSS_ACL_PRIVATE
Public read	The bucket owner and the authorized users can read and write objects in the bucket. Other users can only read the objects in the bucket. Authorize this permission with caution.	OSS_ACL_PUBLIC_READ
Public read-write	All users can read and write objects in the bucket . Authorize this permission with caution.	OSS_ACL_PUBLIC_READ_WRITE

For more information about the ACL, see [Access control](#) in the developer guide.

You can run the following code to configure an ACL for a bucket:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates
    that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as
    timeout.
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_initialize method in main
    () to initialize global resources, such as
    networks and memories. */
    if (aos_http_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, which is
    equivalent to apr_pool_t. The implementation code is
    included in the apr library. */
    aos_pool_t * pool;
    /* Re-create a new memory pool. The second
    parameter is NULL, indicating that it does not
    inherit from any other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter
    mainly includes global configuration
    information, such as endpoint, access_key_id,
    access_key_secret, is_cname, and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create(pool);
    /* Use oss_client_options to initialize client
    options. */
    init_options(oss_client_options);
    /* Initialization parameters. */
    aos_string_t bucket;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_statuses = NULL;
    /* Assign a char * to the aos_string_t bucket. */
```

```

    aos_str_set (& bucket , bucket_name );
    /* Set the bucket ACL to public read .
    resp_status = oss_put_bucket_acl ( oss_client_options , &
    bucket , OSS_ACL_PUBLIC_READ , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " set bucket acl succeeded \n " );
    } else {
        printf ( " set bucket acl failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

Obtain an ACL for a bucket

Run the following code to obtain the ACL for a bucket:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id ,
    access_key_id );
    aos_str_set (& options->config->access_key_secret ,
    access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options->ctl = aos_http_controller_create ( options->
    pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
    () to initialize global resources , such as
    networks and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */

```

```

    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion
    informatio n , such as endpoint , access_key _id ,
    acces_key _secret , is_cname , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t oss_acl ;
    aos_table _t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    /* Assign a char * to the aos_string _t bucket */ /*
    aos_str_se t (& bucket , bucket_nam e );
    /* Obtain the ACL for a bucket . */
    resp_statu s = oss_get_bu cket_acl ( oss_client _options , &
    bucket , & oss_acl , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " get bucket acl succeeded : % s \ n " ,
    oss_acl . data );
    } else {
        printf ( " get bucket acl failed \ n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Obtain the region of a bucket

Run the following code to obtain the region (or location) for a bucket:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
    access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
    access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout .

```

```

    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as
networks and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion
informatio n , such as endpoint , access_key _id ,
acces_key_ secret , is_cname , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t oss_locati on ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    /* Assign a char * to the aos_string _t bucket . */
    aos_str_se t (& bucket , bucket_nam e );
    /* Obtain the region of the bucket . */
    resp_statu s = oss_get_bu cket_locat ion ( oss_client
_options , & bucket , & oss_locati on , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " get bucket location succeeded : % s \ n " ,
oss_locati on . data );
    } else {
        printf ( " get bucket location failed \ n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Obtain bucket information

Run the following code to obtain bucket information:

```
# include " oss_api . h "
```

```

# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as
networks and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion
informatio n , such as endpoint , access_key _id ,
acces_key_ secret , is_cname , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    oss_bucket _info_t bucket_inf o ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    /* Assign a char * to the aos_string _t bucket . */
    aos_str_se t (& bucket , bucket_nam e );
    /* Obtain the bucket inforamtio n . */
    resp_statu s = oss_get_bu cket_info ( oss_client _options ,
& bucket , & bucket_inf o , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {

```

```

        printf (" get    bucket    info    succeeded \ n ");
        printf (" location : % s \ n ", bucket_inf o . location .
data );
        printf (" owner_id : % s \ n ", bucket_inf o . owner_id .
data );
        printf (" owner_name : % s \ n ", bucket_inf o . owner_name
. data );
        printf (" created_da te : % s \ n ", bucket_inf o .
created_da te . data );
        printf (" location : % s \ n ", bucket_inf o . location .
data );
    } else {
        printf (" get    bucket    info    failed \ n ");
    }
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Delete a bucket

Before a bucket is deleted, ensure that all objects in the bucket and fragments that are generated from multipart upload are deleted.



Note:

To delete the fragments that are generated from multipart upload, use `oss_list_upload_part` to list all fragments, and then use `oss_abort_multipart_upload` to delete the fragments. For more information, see [Multipart upload](#).

Run the following code to delete a bucket:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
*/
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}

```

```

}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
    () to initialize global resources , such as
    networks and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_create ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configuration
    information , such as endpoint , access_key_id ,
    access_key_secret , is_cname , and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
    options . */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    /* Assign a char * to the aos_string_t bucket . */
    aos_str_set ( & bucket , bucket_name );
    /* Delete the bucket . */
    resp_status = oss_delete_bucket ( oss_client_options , &
    bucket , & resp_headers );
    if ( aos_status_is_ok ( resp_status ) ) {
        printf ( " delete bucket succeeded \n " );
    } else {
        printf ( " delete bucket failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

7.6 Upload objects

7.6.1 Overview

In OSS, an object is the basic unit for data operations. OSS C SDK provides the following file upload methods:

- **Simple upload**: supports the upload of a file up to 5 GB in size. It includes streaming upload and file upload.
- **Append upload**: supports the upload of a file up to 5 GB in size.
- **Resumable upload**: supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files. It also supports concurrent upload and you can define the size of each part.
- **Multipart upload**: supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files.



Note:

For more information about the usage scenarios of each upload method, see “Upload files” in OSS Developer Guide.

During the file upload, you can set **Object Meta**, and check the upload progress in **upload progress bar**. After you complete the file upload, you can perform **upload callback**.

7.6.2 Simple upload

This topic describes how to use simple upload.

For the complete code of simple upload, see [GitHub](#).

Upload data from memory to OSS

You can run the following code to upload data from memory to OSS:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * object_content = "More than just cloud.";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
        access_key_id);
```

```

    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . The value
0 indicates that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_list_t buffer ;
    aos_buf_t * content = NULL ;
    aos_table_ t * headers = NULL ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t (& bucket , bucket_nam e );
    aos_str_se t (& object , object_nam e );
    aos_list_i nit (& buffer );
    content = aos_buf_pa ck ( oss_client _options -> pool ,
object_con tent , strlen ( object_con tent ));
    aos_list_a dd_tail (& content -> node , & buffer );
    /* Upload a file . */
    resp_statu s = oss_put_ob ject_from_ buffer ( oss_client
_options , & bucket , & object , & buffer , headers , & resp_heade
rs );
    /* Determine whether the upload is successful . */
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " put object from buffer succeeded \ n " );
    } else {
        printf ( " put object from buffer failed \ n " );
    }
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( pool );

```

```

    /* Release allocated global resources . */
    aos_http_io_initialize ();
    return 0 ;
}

```

Upload local files

You can run the following code to upload a local file to OSS:

```

# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options (oss_request_t_options_t * options)
{
    options->config = oss_config_create (options->pool);
    /* Use a char * string to initialize aos_string_t . */
    aos_string_t (& options->config->endpoint, endpoint);
    aos_string_t (& options->config->access_key_id,
access_key_id);
    aos_string_t (& options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used . The value
0 indicates that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options->ctl = aos_http_controller_create (options->
pool , 0 );
}
int main (int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
() to initialize global resources , such as networks
and memories . */
    if (aos_http_initialize (NULL , 0 ) != AOSE_OK ) {
        exit (1);
    }
    /* Memory pool used to manage memories , equivalent
to apr_pool_t . The implementation code is included in
the apr library . */
    aos_pool_t * pool ;
    /* Re - create a memory pool . The second parameter
is NULL , indicating that the new pool does not
inherit any other memory pool . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter
includes global configuration information , such as
endpoint , access_key_id , access_key_secret , is_cname , and
curl . */
    oss_request_t_options_t * oss_client_options ;
    /* Allocate the memories in the memory pool to
options . */
    oss_client_options = oss_request_t_options_create (pool );
    /* Initialize oss_client_options . */
    init_options (oss_client_options);
    /* Initialization parameters . */
    aos_string_t bucket ;

```

```

aos_string_t object ;
aos_string_t file ;
aos_table_t * headers = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );
aos_str_set (& file , local_file_name );
/* Upload a file . */
resp_status = oss_put_object_from_file ( oss_client_options , & bucket , & object , & file , headers , & resp_headers );
/* Determine whether the upload is successful . */
if ( aos_status_is_ok ( resp_status )) {
    printf ( " put object from file succeeded \n " );
} else {
    printf ( " put object from file failed \n " );
}
/* Release the memory pool , that is , memories allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_info_deinitialize ();
return 0 ;
}

```

FAQ: Files with Chinese characters in their names cannot be uploaded.

For example, in Windows, you can upload a file with Chinese characters in its name after encoding its name in UTF-8. The example code is as follows:

```

# include " oss_api . h "
# include " aos_http_info . h "
# include < wchar . h >
const char endpoint = "< yourEndpoint >";
const char access_key_id = "< yourAccessKeyId >";
const char access_key_secret = "< yourAccessKeySecret >";
const char bucket_name = "< yourBucketName >";
const char object_name = "< yourObjectName >";
const char local_file_name_ch [] = " c :\\ test . txt ";
void init_options ( oss_request_options_t options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t . */
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id , access_key_id );
    aos_str_set (& options->config->access_key_secret , access_key_secret );
    /* Determine whether the CNAME is used . The value 0 indicates that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as timeout . */
    options->ctl = aos_http_controller_create ( options->pool , 0 );
}

char * multibyte_to_utf8 ( const char * ch , char * str , int size )
{

```

```

    if (! ch || ! str || size <= 0 )
        return NULL ;

    int chlen = strlen ( ch );
    int multi_byte _cnt = 0 ;
    for ( int i = 0 ; i < chlen - 1 ; i ++ ) {
        if (( ch [ i ] & 0x80 ) && ( ch [ i + 1 ] & 0x80 )) {
            i ++;
            multi_byte _cnt ++;
        }
    }
    if ( multi_byte _cnt == 0 )
        return ch ;

    int len = MultiByteT oWideChar ( CP_ACP , 0 , ch , - 1 ,
NULL , 0 );
    if ( len <= 0 )
        return NULL ;
    wchar_t * wch = malloc ( sizeof ( wchar_t ) * ( len + 1 ));
    if (! wch )
        return NULL ;
    wmemset ( wch , 0 , len + 1 );
    MultiByteT oWideChar ( CP_ACP , 0 , ch , - 1 , wch , len );

    len = WideCharTo MultiByte ( CP_UTF8 , 0 , wch , - 1 , NULL
, 0 , NULL , NULL );
    if (( len <= 0 ) || ( size < len + 1 )) {
        free ( wch );
        return NULL ;
    }
    memset ( str , 0 , len + 1 );
    WideCharTo MultiByte ( CP_UTF8 , 0 , wch , - 1 , str , len ,
NULL , NULL );
    free ( wch );
    return str ;
}

int main ( int argc , char argv [])
{
    /* Call the aos_http_i o_initialize method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_i o_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , equivalent
    to apr_pool_t . The implementa tion code is included in
    the apr library . */
    aos_pool_t pool ;
    /* Re - create a memory pool . The second parameter
    is NULL , indicating that the new pool does not
    inherit any other memory pool . */
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
    includes global configurat ion informatio n , such as
    endpoint , access_key _id , acces_key_ secret , is_cname , and
    curl . */
    oss_reques t_options_ t oss_client _options ;
    /* Allocate the memories in the memory pool to
    options . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Initialize oss_client _options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */

```

```

aos_string_t bucket ;
aos_string_t object ;
aos_string_t file ;
aos_table_t * headers = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );

/* Encode the file name with multiple Chinese
characters in UTF - 8 . */
char local_file_name_buf [ 1024 ];
char * local_file_name = multibyte_to_utf8 ( local_file_name_ch , local_file_name_buf , 1024 );
aos_str_set (& file , local_file_name );

/* Upload a file . */
resp_status = oss_put_object_from_file ( oss_client_options , & bucket , & object , & file , headers , & resp_headers );
/* Determine whether the upload is successful . */
if ( aos_status_is_ok ( resp_status )) {
    printf ( " put object from file succeeded \n " );
} else {
    printf ( " put object from file failed \n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

7.6.3 Append upload

You cannot perform copyObject for appended objects.

Append data from memory

Run the following code to append data from memories

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * object_content = " More than just cloud . ";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t . */
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id ,
access_key_id );
    aos_str_set (& options->config->access_key_secret ,
access_key_secret );
}

```

```

/* Determine whether the CNAME is used . The value
0 indicates that the CNAME is not used . */
options->config->is_cname = 0 ;
/* Configure network parameters , such as timeout . */
options->ctl = aos_http_c ontroller_ create ( options->
pool , 0 );
}
int main ( int argc , char * argv [])
{
/* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
exit ( 1 );
}
/* Memory pool used to manage memories , equivalent
to apr_pool_t . The implementa tion code is included in
the apr library . */
aos_pool_t * pool ;
/* Re - create a memory pool . The second parameter
is NULL , indicating that the new pool does not
inherit any other memory pool . */
aos_pool_c reate ( & p , NULL );
/* Create and initialize options . This parameter
includes global configurat ion informatio n , such as
endpoint , access_key _id , acces_key_ secret , is_cname , and
curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate the memories in the memory pool to
options . */
oss_client _options = oss_reques t_options_ create ( pool );
/* Initialize oss_client _options . */
init_optio ns ( oss_client _options );
/* Initialize parameters . */
aos_string _t bucket ;
aos_string _t object ;
aos_list_t buffer ;
int64_t position = 0 ;
char * next_appen d_position = NULL ;
aos_buf_t * content = NULL ;
aos_table_ t * headers1 = NULL ;
aos_table_ t * headers2 = NULL ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
aos_str_se t (& bucket , bucket_nam e );
aos_str_se t (& object , object_nam e );
headers1 = aos_table_ make ( pool , 0 );
/* Obtain the position where the object is appended
for the first time . */
resp_statu s = oss_head_o bject ( oss_client _options , &
bucket , & object , headers1 , & resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
next_appen d_position = ( char *) ( apr_table_ get (
resp_heade rs , " x - oss - next - append - position "));
position = atoi ( next_appen d_position );
}
/* Append the file . */
headers2 = aos_table_ make ( pool , 0 );
aos_list_i nit (& buffer );
content = aos_buf_pa ck ( pool , object_con tent , strlen (
object_con tent ));
aos_list_a dd_tail (& content -> node , & buffer );

```

```

    resp_status = oss_append_object_from_buffer ( oss_client_options , & bucket , & object , position , & buffer , headers2 , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " append object from buffer succeeded \n " );
    } else {
        printf ( " append object from buffer failed \n " );
    }
    /* Release the memory pool , that is , the memories allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

Append data from a local file

Run the following code to append data from a local file:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t . */
    aos_string_t (& options->config->endpoint , endpoint );
    aos_string_t (& options->config->access_key_id , access_key_id );
    aos_string_t (& options->config->access_key_secret , access_key_secret );
    /* Determine whether the CNAME is used . The value 0 indicates that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options->ctl = aos_http_controller_create ( options->pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main () to initialize global resources , such as networks and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , equivalent to apr_pool_t . The implementation code is included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a memory pool . The second parameter is NULL , indicating that the new pool does not inherit any other memory pool . */
    aos_pool_create (& p , NULL );
}

```



```

/* Create and initialize options . This parameter
includes global configurat ion informatio n , such as
endpoint , access_key _id , acces_key_ secret , is_cname , and
curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate the memories in the memory pool to
options . */
oss_client _options = oss_reques t_options_ create ( pool );
/* Initialize oss_client _options . */
init_optio ns ( oss_client _options );
/* Initialize parameters . */
aos_string _t bucket ;
aos_string _t object ;
aos_string _t file ;
int64_t position = 0 ;
char * next_appen d_position = NULL ;
aos_table_ t * headers1 = NULL ;
aos_table_ t * headers2 = NULL ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
aos_str_se t (& bucket , bucket_nam e );
aos_str_se t (& object , object_nam e );
aos_str_se t (& file , local_file name );
/* Obtain the position where the object is appended
for the first time . */
resp_statu s = oss_head_o bject ( oss_client _options , &
bucket , & object , headers1 , & resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
    next_appen d_position = ( char *) ( apr_table_ get (
resp_heade rs , " x - oss - next - append - position "));
    position = atoi ( next_appen d_position );
}
/* Append the file . */
resp_statu s = oss_append _object_fr om_file ( oss_client
_options , & bucket , & object , position , & file , headers2 , &
resp_heade rs );
/* Determine whether the append upload is successful
. */
if ( aos_status _is_ok ( resp_statu s )) {
    printf ( " append object from file succeeded \ n " );
} else {
    printf ( " append object from file failed \ n " );
}
/* Release the memory pool , that is , the memories
allocated to resources during the request . */
aos_pool_d estroy ( pool );
/* Release the allocated global resources . */
aos_http_i o_deinitia lize ();
return 0 ;
}

```

If the file already exists, the following may occur:

- If the file cannot be uploaded with append upload, an `ObjectNotAppendable` exception occurs.
- If the file can be uploaded with append upload, the file is upload by the Put Object method and overwrites the existing object. The object type is changed to Normal

Object. After the Head Object operation is performed, the system returns x-oss-object-type, which indicates the type of the object.

7.6.4 Resumable upload

In resumable upload, the object is divided into multiple parts, which are then uploaded separately. After all the parts are uploaded, they are merged into a complete object to finish the upload.

Current upload progress is recorded into a checkpoint file during the upload. If a part fails to be uploaded during the process, the object is uploaded from the part recorded in the checkpoint file when the upload restarts, that is, the resumable upload. After the upload is complete, the checkpoint is deleted.

You can use the `oss_resumable_upload` method to achieve resumable upload. Common parameters used in this method are listed as follows:

Parameter	Description
<code>thread_num</code>	The number of concurrent threads, which is 1 by default.
<code>enable_checkpoint</code>	Whether to enable resumable upload, which is disabled by default.
<code>partSize</code>	The part size, ranging from 100 KB to 5 GB.
<code>checkpoint_path</code>	The path that the checkpoint file is stored, which is {upload_file_path}.cp by default.

You can run the following code to perform resumable upload:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize the aos_string_t type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
}
```

```

/* Determnube whether the CNAME is used . 0 indicates
that it is not used . */
options -> config -> is_cname = 0 ;
/* Used to configure network parameters , such as
timeout . */
options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
/* Call the aos_http_i o_initialize method in main
() to initialize global resources , such as networks
and memories . */
if ( aos_http_i o_initialize ( NULL , 0 ) != AOSE_OK ) {
exit ( 1 );
}
/* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
aos_pool_t * pool ;
/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_c reate (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate memories in the memory pool to options
. */
oss_client _options = oss_reques t_options_ create ( pool );
/* Use oss_client _options to initialize client
options */
init_optio ns ( oss_client _options );
/* Initializa tion parameters . */
aos_string _t bucket ;
aos_string _t object ;
aos_string _t file ;
aos_list_t resp_body ;
aos_table_ t * headers = NULL ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
oss_resuma ble_clt_pa rams_t * clt_params ;
aos_str_se t (& bucket , bucket_nam e );
aos_str_se t (& object , object_nam e );
aos_str_se t (& file , local_file name );
aos_list_i nit (& resp_body );
/* Resumable upload . */
clt_params = oss_create _resumable _clt_param s_content (
pool , 1024 * 100 , 3 , AOS_TRUE , NULL );
resp_statu s = oss_resuma ble_upload _file ( oss_client
_options , & bucket , & object , & file , headers , NULL ,
clt_params , NULL , & resp_heade rs , & resp_body );
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " resumable upload succeeded \ n " );
} else {
printf ( " resumable upload failed \ n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_d estroy ( pool );
/* Release allocated global resources . */
aos_http_i o_deinitia lize ();

```

```
    return 0 ;
}
```

7.6.5 Multipart upload

To enable multipart upload, perform the following steps:

1. Initiate a multipart upload event.

You can call `ossClient.initiateMultipartUpload` to return the globally unique `uploadId` created in OSS.

2. Upload parts.

You can call `ossClient.uploadPart` to upload part data.



Notice:

- For parts with a same `uploadId`, parts are sequenced by their part numbers. If you have uploaded a part and use the same part number to upload another part, the later part will replace the former part.
- OSS places the MD5 value of part data in ETag and returns the MD5 value to the user.
- SDK automatically configures Content-MD5. OSS calculates the MD5 value of uploaded data and compares it with the MD5 value calculated by SDK. If the two values vary, the error code of `InvalidDigest` is returned.

3. Complete multipart upload.

After you have uploaded all parts, call `partossClient.completeMultipartUpload` to combine these parts into a complete object.

The following code is used as a complete example that describes the process of multipart upload:

```
# include " oss_api . h "
# include " aos_http_i o . h "
# include < sys / stat . h >
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_nam e = "< yourObject Name >";
const char * local_file name = "< yourLocalF ilename >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    ptions -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
```

```

    aos_str_set (& options -> config -> access_key_id ,
access_key_id );
    aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_controller_create ( options ->
pool , 0 );
}
int64_t get_file_size ( const char * file_path )
{
    int64_t filesize = - 1 ;
    struct stat statbuff ;
    if ( stat ( file_path , & statbuff ) < 0 ){
        return filesize ;
    } else {
        filesize = statbuff . st_size ;
    }
    return filesize ;
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementation code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configuration information , such
as endpoint , access_key_id , access_key_secret , is_cname ,
and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
options . */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_string_t object ;
    oss_upload_file_t * upload_file = NULL ;
    aos_string_t upload_id ;
    aos_table_t * headers = NULL ;
    aos_table_t * complete_headers = NULL ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    aos_str_set (& bucket , bucket_name );
    aos_str_set (& object , object_name );
    aos_strnull (& upload_id );
    headers = aos_table_make ( p , 1 );
    complete_headers = aos_table_make ( p , 1 );

```

```

    int part_num = 1;
    /* Initialize multipart upload and obtain an upload
    ID (upload_id). */
    resp_status s = oss_init_multipart_upload(oss_client
_options, & bucket, & object, & upload_id, headers, &
resp_headers);
    /* Determines whether the slice upload initialization
    is successful. */
    if (aos_status_is_ok(resp_status s)) {
        printf("Init multipart upload succeeded, upload_id
: %.*s \n",
            upload_id.len, upload_id.data);
    } else {
        printf("Init multipart upload failed, upload_id :
%.*s \n",
            upload_id.len, upload_id.data);
    }
    /* Upload the parts. */
    int64_t file_length = 0;
    int64_t pos = 0;
    file_length = get_file_size(local_file_name);
    while (pos < file_length) {
        upload_file = oss_create_upload_file(pool);
        aos_str_set(&upload_file->filename, local_file
name);
        upload_file->file_pos = pos;
        pos += 100 * 1024;
        upload_file->file_last = pos < file_length ? pos
: file_length;
        resp_status s = oss_upload_part_from_file(oss_client
_options, & bucket, & object, & upload_id, part_num++,
upload_file, & resp_headers);
        if (aos_status_is_ok(resp_status s)) {
            printf("Multipart upload part from file
succeeded \n");
        } else {
            printf("Multipart upload part from file
failed \n");
        }
    }
    oss_list_upload_part_params_t * params = NULL;
    aos_list_t complete_part_list;
    /* Obtain uploaded parts. */
    params = oss_create_list_upload_part_params(pool);
    params->max_ret = 1000;
    aos_list_init(&complete_part_list);
    resp_status s = oss_list_upload_part(oss_client_options,
& bucket, & object, & upload_id, params, & resp_headers);
    /* Determine whether the part list is successfully
    obtained. */
    if (aos_status_is_ok(resp_status s)) {
        printf("List multipart succeeded \n");
    } else {
        printf("List multipart failed \n");
    }
    oss_complete_part_content_t * complete_part_content =
NULL;
    oss_list_part_content_t * part_content = NULL;
    aos_list_for_each_entry(oss_list_part_content_t,
part_content, & params->part_list, node) {
        complete_part_content = oss_create_complete_
part_content(pool);
        aos_str_set(&complete_part_content->part_number,
part_content->part_number.data);

```

```

        aos_str_set (& complete_part_content -> etag ,
part_content -> etag . data );
        aos_list_append_tail (& complete_part_content -> node , &
complete_part_list );
    }
    /* Complete multipart upload . */
    resp_status = oss_complete_multipart_upload ( oss_client
_options , & bucket , & object , & upload_id ,
        & complete_part_list , complete_headers , &
resp_headers );
    /* Determine whether the multipart upload is complete
    . */
    if ( aos_status_is_ok ( resp_status )) {
        printf (" Complete multipart upload from file
succeeded , upload_id :%.16s \n " ,
            upload_id . len , upload_id . data );
    } else {
        printf (" Complete multipart upload from file
failed \n ");
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

**Note:**

You must provide all valid PartETags when you call `oss_complete_multipart_upload` to obtain uploaded parts. The ETags can be obtained in two methods: 1. The ETag of a part is included in the returned result when the part is uploaded. 2. You can call `oss_list_upload_part` to obtain the ETag of an uploaded part. In the example, the second method is used.

Cancel a multipart upload event

Run the following code to cancel a multipart upload event:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options ( oss_request_options_t * options )
{
    options -> config = oss_config_create ( options -> pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_str_set (& options -> config -> endpoint , endpoint );
    aos_str_set (& options -> config -> access_key_id ,
access_key_id );
    aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
}

```

```

/* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
options -> config -> is_cname = 0 ;
/* Used to configure network parameters , such as
timeout . */
options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
/* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as
networks and memories . */
if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
exit ( 1 );
}
/* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
aos_pool_t * pool ;
/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_c reate (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configurat ion
informatio n , such as endpoint , access_key _id ,
acces_key_ secret , is_cname , and curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate memories in the memory pool to options
. */
oss_client _options = oss_reques t_options_ create ( pool );
/* Use oss_client _options to initialize client
options . */
init_optio ns ( oss_client _options );
/* Initializa tion parameters . */
aos_string _t bucket ;
aos_string _t object ;
aos_string _t upload_id ;
aos_table_ t * headers = NULL ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
aos_str_se t (& bucket , bucket_nam e );
aos_str_se t (& object , object_nam e );
aos_str_nu ll (& upload_id );
/* Initialize multipart upload and obtain an upload
ID ( upload_id ) . */
resp_statu s = oss_init_m ultipart_u pload ( oss_client
_options , & bucket , & object , & upload_id , headers , &
resp_heade rs );
/* Determine whether the multipart upload is
initialize d successful ly . */
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " Init multipart upload succeeded , upload_id
: %.* s \ n " ,
upload_id . len , upload_id . data );
} else {
printf ( " Init multipart upload failed , upload_id :
%.* s \ n " ,
upload_id . len , upload_id . data );
}
}

```



```

/* Cancel the multipart upload . */
resp_status s = oss_abort_multipart_upload ( oss_client
_options , & bucket , & object , & upload_id , & resp_headers );
/* Determine whether the multipart upload is canceled
successfully . */
if ( aos_status_is_ok ( resp_status s )) {
    printf ( " Abort multipart upload succeeded , upload_id
::%.*s \n ",
            upload_id . len , upload_id . data );
} else {
    printf ( " Abort multipart upload failed \n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_info_deinitialize ();
return 0 ;
}

```

**Note:**

If you cancel a multipart upload event, you are not allowed to perform any other operations with this upload_id anymore. The uploaded parts will be deleted.

7.6.6 Upload progress bars

You can use progress bars to indicate the upload or download progress.

For the complete code of upload progress bar, see [GitHub](#).

The following code is used as an example to describe how to view progress information with PutObject:

```

# include " oss_api . h "
# include " aos_http_info . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_string_t (& options->config->endpoint , endpoint );
    aos_string_t (& options->config->access_key_id ,
    access_key_id );
    aos_string_t (& options->config->access_key_secret ,
    access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options->ctl = aos_http_controller_create ( options->
    pool , 0 );
}

```

```

}
void percentage ( int64_t consumed_bytes , int64_t
total_bytes )
{
    assert ( total_bytes >= consumed_bytes );
    printf ("%%%" APR_INT64_T_FMT "\n", consumed_bytes *
100 / total_bytes );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_create ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configuration information , such
    as endpoint , access_key_id , access_key_secret , is_cname
    , and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
    options . */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_string_t object ;
    aos_string_t file ;
    aos_list_t resp_body ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    aos_str_set ( & bucket , bucket_name );
    aos_str_set ( & object , object_name );
    aos_str_set ( & file , local_file_name );
    /* Upload with a progress bar displayed . */
    resp_status = oss_do_put_object_from_file ( oss_client
_options , & bucket , & object , & file , NULL , NULL , percentage
, & resp_headers , & resp_body );
    if ( aos_status_is_ok ( resp_status ) ) {
        printf ( "put object from file succeeded \n " );
    } else {
        printf ( "put object from file failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

```
}
```

`oss_put_object_from_file` and `oss_put_object_from_buffer` do not support progress bars.

7.6.7 Upload callback

Run the following code for upload callback:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * object_content = "More than just cloud.";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. The value
0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    aos_pool_t * p = NULL;
    aos_status_t * s = NULL;
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t * headers = NULL;
    oss_request_options_t * options = NULL;
    aos_table_t * resp_headers = NULL;
    aos_list_t resp_body;
    aos_list_t buffer;
    aos_buf_t * content;
    char * buf = NULL;
    int64_t len = 0;
    int64_t size = 0;
    int64_t pos = 0;
    char b64_buf[1024];
    int b64_len;
    /* Use the JSON format. */
    char * callback = "{
        \"callbackUrl\": \"http://oss-demo.aliyuncs.com:
23450\",
        \"callbackHost\": \"oss-cn-hangzhou.aliyuncs.com
\",
        \"callbackBody\": \"bucket=${bucket}&object=${
object}&size=${size}&mimeType=${mimeType}\"
    }";
```

```

        "\"" callbackBodyType "\":" application / x - www - form -
        urlencoded "\""
    }";
    /* Call the aos_http_initialize method in main
    () to initialize global resources, such as networks
    and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Initialize parameters . */
    aos_pool_create ( & p , NULL );
    options = oss_request_options_create ( p );
    init_options ( options );
    aos_str_set ( & bucket , bucket_name );
    aos_str_set ( & object , object_name );
    aos_list_init ( & resp_body );
    aos_list_init ( & buffer );
    content = aos_buf_pack ( options -> pool , object_content
, strlen ( object_content ));
    aos_list_append ( & content -> node , & buffer );
    /* Add callback into the header . */
    b64_len = aos_base64_encode ( ( unsigned char *) callback ,
strlen ( callback ), b64_buf );
    b64_buf [ b64_len ] = '\0';
    headers = aos_table_make ( p , 1 );
    apr_table_set ( headers , OSS_CALLBACK , b64_buf );
    /* Start upload callback . */
    s = oss_do_put_object_from_buffer ( options , & bucket , &
object , & buffer ,
    headers , NULL , NULL , & resp_headers , & resp_body );
    if ( aos_status_is_ok ( s )) {
        printf ( " put object from buffer succeeded \n " );
    } else {
        printf ( " put object from buffer failed \n " );
    }
    /* Obtain the buffer length . */
    len = aos_buf_list_len ( & resp_body );
    buf = ( char *) aos_palloc ( p , ( apr_size_t ) ( len + 1
));
    buf [ len ] = '\0';
    /* Copy the content in the buffer to the memory .
    */
    aos_list_foreach_entry ( aos_buf_t , content , & resp_body
, node ) {
        size = aos_buf_size ( content );
        memcpy ( buf + pos , content -> pos , ( size_t ) size );
        pos += size ;
    }
    /* Release the memory pool , that is , the memories
    allocated to resources during the request . */
    aos_pool_destroy ( p );
    /* Release the allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

For more information about upload callback, see [Upload callback](#). For detailed debugging methods and troubleshooting information, see [Upload callback error and exclusion](#).

7.7 Download objects

7.7.1 Overview

For the complete code of object download, see [GitHub](#).

OSS C SDK provides the following download methods:

- [Download to your local file](#)
- [Download to local memory](#)
- [Range download](#)
- [Resumable download](#)

You can view the download progress shown in the [download progress bar](#).

7.7.2 Download an object to a local file

For the complete code of downloading OSS object, see [GitHub](#).

You can run the following code to download the specified OSS object to a local file.

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize the aos_string_t type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates that the CNAME is used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as timeout */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources, such as networks and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
}
```

```

    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_create ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configuration information , such
    as endpoint , access_key_id , access_key_secret , is_cname
    , and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
    options */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_string_t object ;
    aos_string_t file ;
    aos_table_t * params ;
    aos_table_t * headers = NULL ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    aos_str_set_t ( & bucket , bucket_name );
    aos_str_set_t ( & object , object_name );
    aos_str_set_t ( & file , local_file_name );
    params = aos_table_make ( pool , 0 );
    /* Download the object . A new local file is
    created if the specified file does not exist . If
    the specified local file exists , it is overwritten .
    */
    resp_status = oss_get_object_to_file ( oss_client_options , & bucket , & object , headers , params , & file , & resp_headers );
    if ( aos_status_is_ok ( resp_status ) ) {
        printf ( " Get object from file succeeded \n " );
    } else {
        printf ( " Get object from file failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_io_deinitialize ();
    return 0 ;
}

```

**Note:**

Compared to the 1.0.0 version, the params parameter is added to the oss_get_object_to_file interface, and the value of the headers and params parameters can be NULL.

7.7.3 Download to your local memory

For the complete code of object download, see [GitHub](#).

Run the following code to download a specified object to local memory:

```
# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_nam e = "< yourObject Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_list_t buffer ;
```

```

aos_buf_t * content = NULL ;
aos_table_t * params = NULL ;
aos_table_t * headers = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
char * buf = NULL ;
int64_t len = 0 ;
int64_t size = 0 ;
int64_t pos = 0 ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );
aos_list_init (& buffer );
/* Download an object to local memory . */
resp_status = oss_get_object_to_buffer ( oss_client
_options , & bucket , & object ,
                                headers , params , & buffer , &
resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get object to buffer succeeded \n " );
    /* Copy the downloaded content to the buffer . */
    len = aos_buf_list_len (& buffer );
    buf = aos_palloc ( pool , len + 1 );
    buf [ len ] = '\0 ' ;
    aos_list_foreach_entry ( aos_buf_t , content , & buffer
, node ) {
        size = aos_buf_size ( content );
        memcpy ( buf + pos , content->pos , size );
        pos += size ;
    }
} else {
    printf ( " get object to buffer failed \n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

7.7.4 Range download

If you only need part of the data in an object, you can use range downloads to download specified content.

Run the following code for range download:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccess KeyId >";
const char * access_key_secret = "< yourAccess KeySecret >";
const char * bucket_name = "< yourBucket Name >";
const char * object_name = "< yourObject Name >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize the aos_string
    _t type . */
    aos_str_set (& options->config->endpoint , endpoint );
}

```



```

    aos_str_se_t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se_t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_list_t buffer ;
    aos_buf_t * content = NULL ;
    aos_table_ t * params = NULL ;
    aos_table_ t * headers = NULL ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    char * buf = NULL ;
    int64_t len = 0 ;
    int64_t size = 0 ;
    int64_t pos = 0 ;
    aos_str_se_t (& bucket , bucket_nam e );
    aos_str_se_t (& object , object_nam e );
    aos_list_i nit (& buffer );
    headers = aos_table_ make ( pool , 1 );
    /* Set the download range , which ranges from the
20th byte to the 100th byte . */
    apr_table_ set ( headers , " Range " , " bytes = 20 - 100 " );
    /* Download the object to memory . */
    resp_statu s = oss_get_ob ject_to_bu ffer ( oss_client
_options , & bucket , & object , headers , params , & buffer , &
resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {

```

```

        printf (" get object to buffer succeeded \n ");
        /* Obtain the buffer length . */
        aos_list_foreach (aos_buf_t, content, &buffer
, node) {
            len += aos_buf_size (content);
        }
        buf = aos_palloc (pool, (apr_size_t)(len + 1));
        buf[len] = '\0';
        /* Copy the content in the buffer to memory .
*/
        aos_list_foreach (aos_buf_t, content, &buffer
, node) {
            size = aos_buf_size (content);
            memcpy (buf + pos, content->pos, (size_t) size
);
            pos += size;
        }
    }
    else {
        printf (" get object to buffer failed \n ");
    }
    /* Release the memory pool, that is, memories
allocated to resources during the request . */
    aos_pool_destroy (pool);
    /* Release allocated global resources . */
    aos_http_io_deinitialize ();
    return 0;
}

```

7.7.5 Resumable download

Large-sized objects may fail to be downloaded because of unstable network or other exceptions, and the entire object needs to be downloaded again. However, the object may still fail to be downloaded after multiple attempts. Therefore, OSS provides resumable download.

To enable resumable download, you need to split the object you want to download into multiple parts and download them separately. After you have downloaded all parts, these parts will be combined into a complete object.

You can use `oss_resumable_client_params_t` to enable resumable download.

`oss_resumable_client_params_t` contains the following parameters:

Parameter	Description
thread_num	The number of concurrent threads, which is 1 by default.
enable_checkpoint	Determines whether to enable resumable download, which is disabled by default.
partSize	Specifies the size of a part. The value can be between 1 byte to 5 GB and the unit is byte.

Parameter	Description
<code>checkpoint_path</code>	The path where the checkpoint file is located, which is under <code>{upload_file_path}.cp</code> by default.

This `checkpoint` file stores information about the download progress. If you fail to download a part, the download will continue based on the progress recorded in the `checkpoint` file. After the object is downloaded, the `checkpoint` file will be deleted.

Run the following code for resumable download:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize the aos_string_t type. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources, such as networks and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, which is equivalent to apr_pool_t. The implementation code is included in the apr library. */
    aos_pool_t * pool;
    /* Re-create a new memory pool. The second parameter is NULL, indicating that it does not inherit from any other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter mainly includes global configuration information, such
```

```

    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_string _t file ;
    aos_table_ t * headers = NULL ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    oss_resuma ble_clt_pa rams_t * clt_params ;
    aos_str_se t (& bucket , bucket_nam e );
    aos_str_se t (& object , object_nam e );
    aos_str_se t (& file , local_file name );
    /* Download the object using resumable download . */
    clt_params = oss_create _resumable _clt_param s_content (
    pool , 1024 * 100 , 3 , AOS_TRUE , NULL );
    resp_statu s = oss_resuma ble_downlo ad_file ( oss_client
    _options , & bucket , & object , & file , headers , NULL ,
    clt_params , NULL , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " download succeeded \ n " );
    } else {
        printf ( " download failed \ n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

7.7.6 Download progress bars

You can use progress bars to indicate the upload or download progress.

For the complete code of download progress bar, see [GitHub](#).

The following code is used as an example to describe how to use the download progress bar.

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_nam e = "< yourObject Name >";
const char * local_file name = "< yourLocalF ilename >";
const char * download_f ilename = "< yourDownlo adFilename >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
}

```

```

/* Use a char * string to initialize aos_string_t .
*/
aos_str_se t (& options -> config -> endpoint , endpoint );
aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
/* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
options -> config -> is_cname = 0 ;
/* Used to configure network parameters , such as
timeout . */
options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
void percentage ( int64_t consumed_b ytes , int64_t
total_byte s )
{
    assert ( total_byte s >= consumed_b ytes );
    printf ("%%" APR_INT64_ T_FMT "\n " , consumed_b ytes *
100 / total_byte s );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_string _t file ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_list_t resp_body ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t (& bucket , bucket_nam e );
    aos_str_se t (& object , object_nam e );
    aos_str_se t (& file , local_file name );
    /* Upload with a progress bar displayed . */
    resp_statu s = oss_do_put _object_fr om_file ( oss_client
_options , & bucket , & object , & file , NULL , NULL , percentage
, & resp_heade rs , & resp_body );
    if ( aos_status _is_ok ( resp_statu s )) {

```

```

        printf (" put    object    from    file    succeeded \ n ");
    } else {
        printf (" put    object    from    file    failed \ n ");
    }
    /* Download with a progress bar displayed . */
    aos_pool_c reate (& pool , NULL );
    oss_client_options = oss_request_options_create ( pool );
    init_options ( oss_client_options );
    aos_str_set (& file , download_filename );
    resp_status = oss_do_get_object_to_file ( oss_client_options , & bucket , & object , NULL , NULL , & file , percentage , NULL );
    if ( aos_status_is_ok ( resp_status )) {
        printf (" get    object    to    file    succeeded \ n ");
    } else {
        printf (" get    object    to    file    failed \ n ");
    }
    /* Release the memory pool , that is , memories allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

`oss_get_object_from_file` and `oss_get_object_from_buffer` do not support progress bars.

7.8 Manage objects

7.8.1 Overview

You can perform the following operations on objects in a bucket with APIs:

- [Manage ACL for an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Copy objects](#)
- [Delete objects](#)
- [Restore an archive object](#)
- [Manage a symbolic link](#)

7.8.2 Manage ACL for an object

The following table describes the permissions included in the Access Control List (ACL) for an object.

Permission	Description	Value
default	The ACL of an object is the same with that of its bucket.	OSS_ACL_DEFAULT
Private	Only the object owner and authorized users can read and write the object.	OSS_ACL_PRIVATE
Public read	Only the object owner and authorized users can read and write the object. Other users can only read the object. Authorize this permission with caution.	OSS_ACL_PUBLIC_READ
Public read-write	All users can read and write the object. Authorize this permission with caution.	OSS_ACL_PUBLIC_READ_WRITE

The ACL of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

Configure an ACL for an object

You can run the following code to configure an ACL for an object:

```
# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_nam e = "< yourObject Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
```

```

    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_table _t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t ( & bucket , bucket_nam e );
    aos_str_se t ( & object , object_nam e );
    oss_acl_e oss_acl = OSS_ACL_PU BLIC_READ ;
    /* Configure an ACL for an object . */
    resp_statu s = oss_put_ob ject_acl ( oss_client _options , &
bucket , & object , oss_acl , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " put object acl success ! \ n " );
    } else {
        printf ( " put object acl failed ! \ n " );
    }
    /* Obtain the ACL for an object . */
    aos_string _t oss_acl_st ring ;
    resp_statu s = oss_get_ob ject_acl ( oss_client _options , &
bucket , & object , & oss_acl_st ring , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " get object acl success ! \ n " );
        printf ( " acl : % s \ n " , oss_acl_st ring . data );
    } else {
        printf ( " get object acl failed ! \ n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release the allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```



```
}
```

7.8.3 Manage Object Meta

Object Meta includes HTTP headers and user-defined Object Meta. For more information, see [Object Meta](#) in OSS Developer Guide.

Obtain Object Meta

After uploading files to OSS or before reading objects from OSS, you can use the

`oss_get_object_meta` interface to obtain some of meta information about the objects, such as the length and type. Run the following code to configure and obtain Object Meta:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. The value
0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
() to initialize global resources, such as networks
and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, equivalent
to apr_pool_t. The implementation code is included in
the apr library. */
    aos_pool_t * pool;
    /* Re-create a memory pool. The second parameter
is NULL, indicating that the new pool does not
inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter
includes global configuration information, such as
endpoint, access_key_id, access_key_secret, is_cname, and
curl. */
```

```

    oss_req_t options;
    /* Allocate the memories in the memory pool to
options */
    oss_client_options = oss_req_create ( pool );
    /* Initialize oss_client_options */
    init_options ( oss_client_options );
    /* Initialize parameters */
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t * headers;
    aos_list_t buffer;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_status = NULL;
    aos_buf_t * content = NULL;
    char * content_length_str = NULL;
    char * object_type = NULL;
    char * object_author = NULL;
    int64_t content_length = 0;
    aos_str_set (& bucket, bucket_name);
    aos_str_set (& object, object_name);
    headers = aos_table_make ( pool, 2 );
    // Configure custom Object Meta */
    apr_table_set ( headers, "Expires", "Fri, 28 Feb 2032
05:38:42 GMT");
    apr_table_set ( headers, "x-oss-meta-author", "oss
");
    aos_list_init (& buffer);
    content = aos_buf_pack ( oss_client_options->pool,
object_content, strlen ( object_content));
    aos_list_append (& content->node, & buffer);
    /* Set object permissions and upload the object
from the cache */
    resp_status = oss_put_object_from_buffer ( oss_client
_options, & bucket, & object,
    & buffer, headers, & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf (" put object from buffer with md5
succeeded \n ");
    } else {
        printf (" put object from buffer with md5 failed
\n ");
    }
    /* Obtain the object permission */
    resp_status = oss_get_object_meta ( oss_client_options,
& bucket, & object, & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        content_length_str = ( char *) apr_table_get (
resp_headers, OSS_CONTENT_LENGTH );
        if ( content_length_str != NULL ) {
            content_length = atol ( content_length_str );
        }
        object_author = ( char *) apr_table_get ( resp_headers,
OSS_AUTHORIZATION );
        object_type = ( char *) apr_table_get ( resp_headers,
OSS_OBJECT_TYPE );
        printf (" get object meta succeeded, object
author:%s, object type:%s, content_length:%ld \n ",
object_author, object_type, content_length );
    } else {
        printf (" req:%s, get object meta failed \n ",
resp_status->req_id );
    }
    /* Release the memory pool, that is, the memories
allocated to resources during the request */

```

```

aos_pool_d estroy ( pool );
/* Release the allocated global resources . */
aos_http_i o_deinitia lize ();
return 0 ;
}

```

**Note:**

For detailed information about HTTP header, see [RFC2616](#).

7.8.4 List objects

For the complete code for listing objects, see [GitHub](#).

OSS objects are listed alphabetically. You can use `oss_list_o bject` to list all objects in a bucket. The following table describes common parameters used to list objects.

Parameter	Description
delimiter	Specifies a delimiter used to group objects. All those objects whose names contain the specified prefix and after which the delimiter occurs for the first time, act as a group of elements, that is, commonPrefixes.
prefix	Specifies the prefix that the listed objects must contain.
max_ret	Specifies the maximum number of objects that can be listed. The default value is 1000.
marker	Specifies the initial object that is listed.

List all objects

You can run the following code to list all objects in the current bucket.

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize the aos_string
    _t type . */
    aos_str_se t (& options -> config -> endpoint , endpoint );
}

```

```

        aos_str_set (& options -> config -> access_key_id ,
        access_key_id );
        aos_str_set (& options -> config -> access_key_secret ,
        access_key_secret );
        /* Determine whether the CNAME is used . 0 indicates
        that the CNAME is not used . */
        options -> config -> is_cname = 0 ;
        /* Used to configure network parameters , such as
        timeout . */
        options -> ctl = aos_http_controller_create ( options ->
        pool , 0 );
    }
    int main ( int argc , char * argv [])
    {
        /* Call the aos_http_initialize method in main
        () to initialize global resources , such as networks
        and memories . */
        if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
            exit ( 1 );
        }
        /* Memory pool used to manage memories , which is
        equivalent to apr_pool_t . The implementation code is
        included in the apr library . */
        aos_pool_t * pool ;
        /* Re - create a new memory pool . The second
        parameter is NULL , indicating that it does not
        inherit from any other memory pools . */
        aos_pool_create (& pool , NULL );
        /* Create and initialize options . This parameter
        mainly includes global configuration information , such
        as endpoint , access_key_id , access_key_secret , is_cname
        , and curl . */
        oss_request_options_t * oss_client_options ;
        /* Allocate memories in the memory pool to options
        . */
        oss_client_options = oss_request_options_create ( pool );
        /* Use oss_client_options to initialize client
        options . */
        init_options ( oss_client_options );
        /* Initialization parameters . */
        aos_string_t bucket ;
        aos_status_t * resp_status = NULL ;
        oss_list_object_params_t * params = NULL ;
        oss_list_object_content_t * content = NULL ;
        int size = 0 ;
        char * line = NULL ;
        char * prefix = "" ;
        char * nextMarker = "" ;
        aos_str_set (& bucket , bucket_name );
        params = oss_create_list_object_params ( pool );
        params -> max_ret = 100 ;
        aos_str_set (& params -> prefix , prefix );
        aos_str_set (& params -> marker , nextMarker );
        printf ( " Object \ tSize \ tLastModified \ n " );
        /* List all objects . */
        do {
            resp_status = oss_list_object ( oss_client_options ,
            & bucket , params , NULL );
            if (! aos_status_is_ok ( resp_status ))
            {
                printf ( " list object failed \ n " );
                break ;
            }
        }
    }

```

```

        aos_list_for_each_entry ( oss_list_object_content_t
, content , & params -> object_list , node ) {
            ++ size ;
            line = apr_psprintf ( pool , "%.*s \t %.*s \t %.*s \n " ,
content -> key . len , content -> key . data ,
content -> size . len , content -> size . data ,
content -> last_modified . len , content ->
last_modified . data );
            printf ("%s " , line );
        }
        nextMarker = apr_psprintf ( pool , "%.*s " , params ->
next_marker . len , params -> next_marker . data );
        aos_str_set (& params -> marker , nextMarker );
        aos_list_init (& params -> object_list );
        aos_list_init (& params -> common_pre_fix_list );
    } while ( params -> truncated == AOS_TRUE );
    printf (" Total %d \n " , size );
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */
    aos_http_io_deinitialize ();
    return 0 ;
}

```

**Note:**

By default, if a bucket contains more than 1,000 objects, only the first 1,000 objects are returned and the truncated parameter in the returned results is true. Additionally, the next_marker is returned to serve as the start point of next read. To adjust the number of returned objects, you can modify the max_ret parameter or use the marker parameter for multiple reads.

List specified number of objects

You can run the following code to list specified number of objects.

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options -> config = oss_config_create ( options -> pool );
    /* aos_str_set uses a char * string to initialize
the aos_string_t type .*/
    aos_str_set (& options -> config -> endpoint , endpoint );
    aos_str_set (& options -> config -> access_key_id ,
access_key_id );
    aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used .*/
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout .*/
}

```

```

    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as networks
    and memories .*/
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library .*/
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pool .*/ */
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl .*/
    oss_reques t_options_ t * oss_client _options ;
    /* Memories for options are allocated by the memory
    pool and are released after the memory pool is
    released . Therefore , you do not need to release the
    memory for options individual ly .*/
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options .*/
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters */
    aos_string _t bucket ;
    aos_status _t * resp_statu s = NULL ;
    oss_list_o bject_para ms_t * params = NULL ;
    oss_list_o bject_cont ent_t * content = NULL ;
    int size = 0 ;
    char * line = NULL ;
    char * prefix = "" ;
    char * nextMarker = "" ;
    aos_str_se t ( & bucket , bucket_nam e );
    params = oss_create _list_obje ct_params ( pool );
    /* Set the maximum number of listed objects to
    200 .*/
    params -> max_ret = 200 ;
    aos_str_se t ( & params -> prefix , prefix );
    aos_str_se t ( & params -> marker , nextMarker );
    printf ( " Object \ tSize \ tLastModif ied \ n " );
    /* List all objects */
    resp_statu s = oss_list_o bject ( oss_client _options , &
bucket , params , NULL );
    if ( ! aos_status _is_ok ( resp_statu s ))
    {
        printf ( " list object failed \ n " );
    }
    aos_list_f or_each_en try ( oss_list_o bject_cont ent_t ,
content , & params -> object_lis t , node ) {
        ++ size ;
        line = apr_psprin tf ( pool , "%.* s \ t %.* s \ t %.* s \
n " , content -> key . len , content -> key . data ,
        content -> size . len , content -> size . data ,
        content -> last_modif ied . len , content ->
last_modif ied . data );
    }
}

```

```

        printf ("%s ", line );
    }
    printf (" Total %d \n ", size );
    /* Release the memory pool after a request is
    executed , which means that memories allocated for all
    resources during the request are released . */
    aos_pool_t destroy ( pool );
    /* Before the project ends , call the aos_http_i
    o_deinitia lize method to release allocated global
    resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

List objects with specified prefix

You can run the following code to list objects with specified prefix.

```

#include "oss_api.h"
#include "aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* aos_str_se t uses a char * string to initialize
    the aos_string _t data type . */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
    access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
    access_key _secret );
    /* Determine whether the CNAME is used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
    pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , including networks
    and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl . */
    oss_reques t_options_ t * oss_client _options ;

```

```

/* Memories for options are allocated by the memory
pool and are released after the memory pool is
released. Therefore, you do not need to release the
memory for options individually. */
oss_client_options = oss_request_options_create ( pool );
/* Use oss_client_options to initialize client
options. */
init_options ( oss_client_options );
/* Initialization parameters */
aos_string_t bucket;
aos_status_t * resp_status = NULL;
oss_list_object_params_t * params = NULL;
oss_list_object_content_t * content = NULL;
int size = 0;
char * line = NULL;
/* List objects with specified prefix. */
char * prefix = "< yourObject Prefix >";
char * nextMarker = "";
aos_str_set (& bucket, bucket_name);
params = oss_create_list_object_params ( pool );
params->max_ret = 100;
aos_str_set (& params->prefix, prefix);
aos_str_set (& params->marker, nextMarker);
printf (" Object \ tSize \ tLastModified \ n ");
/* List all objects. */
do {
    resp_status = oss_list_object ( oss_client_options,
& bucket, params, NULL );
    if (! aos_status_is_ok ( resp_status ))
    {
        printf (" list object failed \ n ");
        break;
    }
    aos_list_for_each_entry ( oss_list_object_content_t
, content, & params->object_list, node ) {
        ++ size;
        line = apr_psprintf ( pool, "%.5s \ t %.5s \ t %.5s \ n ",
content->key.len, content->key.data,
content->size.len, content->size.data,
content->last_modified.len, content->
last_modified.data );
        printf ("%s", line );
    }
    nextMarker = apr_psprintf ( pool, "%.5s", params->
next_marker.len, params->next_marker.data );
    aos_str_set (& params->marker, nextMarker);
    aos_list_init (& params->object_list);
    aos_list_init (& params->common_prefix_list);
} while ( params->truncated == AOS_TRUE );
printf (" Total %d \ n ", size );
/* Release the memory pool after a request is
executed, which means that memories allocated for all
resources during the request are released. */
aos_pool_destroy ( pool );
/* Before the project ends, call the aos_http_i
o_deinitialize method to release allocated global
resources. */
aos_http_io_deinitialize ();
return 0;

```



```
}
```

Specifies the initial object that is listed.

The value of marker is the name of the initial object. You can run the following code to list objects after marker in alphabetic order.

```
# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* aos_str_se t uses a char * string to initialize
    the aos_string _t type .*/
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used .*/
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout .*/
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as network and
    memories .*/
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library .*/
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools .*/
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl .*/
    oss_reques t_options_ t * oss_client _options ;
    /* Memories for options are allocated by the memory
    pool and are released after the memory pool is
    released . Therefore , you do not need to release the
    memory for options individual ly .*/
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options .*/
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters */
    aos_string _t bucket ;
    aos_status _t * resp_statu s = NULL ;
```

```

    oss_list_object_params_t * params = NULL ;
    oss_list_object_content_t * content = NULL ;
    int size = 0 ;
    char * line = NULL ;
    char * prefix = "" ;
    Specifies the initial object that is listed . */
    char * nextMarker = "< yourObject Marker >";
    aos_str_set (& bucket , bucket_name );
    params = oss_create_list_object_params ( pool );
    params->max_ret = 100 ;
    aos_str_set (& params->prefix , prefix );
    aos_str_set (& params->marker , nextMarker );
    printf (" Object \ tSize \ tLastModified \ n ");
    /* List all objects */
    do {
        resp_status = oss_list_object ( oss_client_options ,
        & bucket , params , NULL );
        if (! aos_status_is_ok ( resp_status ))
        {
            printf (" list object failed \ n ");
            break ;
        }
        aos_list_for_each_entry ( oss_list_object_content_t
        , content , & params->object_list , node ) {
            ++ size ;
            line = apr_psprintf ( pool , "%.*s \ t %.*s \ t %.*s \ n " ,
            content->key . len , content->key . data ,
            content->size . len , content->size . data ,
            content->last_modified . len , content->
            last_modified . data );
            printf ("% s " , line );
        }
        nextMarker = apr_psprintf ( pool , "%.*s " , params->
        next_marker . len , params->next_marker . data );
        aos_str_set (& params->marker , nextMarker );
        aos_list_init (& params->object_list );
        aos_list_init (& params->common_prefix_list );
    } while ( params->truncated == AOS_TRUE );
    printf (" Total % d \ n " , size );
    /* Release the memory pool after a request is
    executed , which means that memories allocated for all
    resources during the request are released . */
    aos_pool_destroy ( pool );
    /* Before the project ends , call the aos_http_i
    o_deinitialize method to release allocated global
    resources . */
    aos_http_io_deinitialize ();
    return 0 ;
}

```

7.8.5 Delete objects



Warning:

Delete objects with caution because deleted objects cannot be recovered.

For the complete code of deleting objects, see [GitHub](#).

Delete an object

Run the following code to delete a single object:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates
    that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as
    timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
    () to initialize global resources, such as networks
    and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, which is
    equivalent to apr_pool_t. The implementation code is
    included in the apr library. */
    aos_pool_t * pool;
    /* Re-create a new memory pool. The second
    parameter is NULL, indicating that it does not
    inherit from any other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter
    mainly includes global configuration information, such
    as endpoint, access_key_id, access_key_secret, is_cname
    , and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create(pool);
    /* Use oss_client_options to initialize client
    options. */
    init_options(oss_client_options);
    /* Initialization parameters. */
    aos_string_t bucket;
    aos_string_t object;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_status = NULL;
    /* Assign char * data to the aos_string_t bucket.
    */
}
```

```

    aos_str_set (& bucket , bucket_name );
    aos_str_set (& object , object_name );
    /* Delete an object . */
    resp_status = oss_delete_object ( oss_client_options , &
    bucket , & object , & resp_headers );
    /* Determine whether the object is deleted
    successfully . */
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " delete object succeed \n " );
    } else {
        printf ( " delete object failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_io_initialize ();
    return 0 ;
}

```

Delete multiple objects

Run the following code to delete multiple objects simultaneously:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name1 = "< yourObjectName1 >";
const char * object_name2 = "< yourObjectName2 >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id ,
    access_key_id );
    aos_str_set (& options->config->access_key_secret ,
    access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options->ctl = aos_http_controller_create ( options->
    pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_io_initialize method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_io_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool ;

```

```

/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_c reate (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate memories in the memory pool to options
. */
oss_client _options = oss_reques t_options_ create ( pool );
/* Use oss_client _options to initialize client
options . */
init_optio ns ( oss_client _options );
/* Initializa tion parameters . */
aos_string _t bucket ;
aos_string _t object1 ;
aos_string _t object2 ;
int is_quiet = 1 ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
aos_str_se t (& bucket , bucket_nam e );
aos_str_se t (& object1 , object_nam e1 );
aos_str_se t (& object2 , object_nam e2 );
/* Construct the list of objects to be deleted . */
aos_list_t object_lis t ;
aos_list_t deleted_ob ject_list ;
oss_object _key_t * content1 ;
oss_object _key_t * content2 ;
aos_list_i nit (& object_lis t );
aos_list_i nit (& deleted_ob ject_list );
content1 = oss_create _oss_objec t_key ( pool );
aos_str_se t (& content1 -> key , object_nam e1 );
aos_list_a dd_tail (& content1 -> node , & object_lis t );
content2 = oss_create _oss_objec t_key ( pool );
aos_str_se t (& content2 -> key , object_nam e2 );
aos_list_a dd_tail (& content2 -> node , & object_lis t );
/* Delete objects in the list . `is_quiet` is used
to determine whether the result of deletion is
returned . */
resp_statu s = oss_delete _objects ( oss_client _options
, & bucket , & object_lis t , is_quiet , & resp_heade rs , &
deleted_ob ject_list );
if ( aos_status _is_ok ( resp_statu s )) {
    printf ( " delete objects succeeded \ n " );
} else {
    printf ( " delete objects failed \ n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_d estroy ( pool );
/* Release the allocated global resources . */
aos_http_i o_deinitia lize ();
return 0 ;
}

```

Delete multiple objects with a specified prefix

You can delete a directory by deleting multiple objects with a specified prefix.

For example, you can delete the `dir` directory by deleting objects with the prefix

dir /. Run the following code to delete multiple objects with a specified prefix simultaneously:

```
# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_pre fix = "< yourObject NamePrefix >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t prefix ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t (& bucket , bucket_nam e );
    aos_str_se t (& prefix , object_pre fix );
    /* Delete objects with a specified prefix . */
```

```

    resp_status s = oss_delete_objects_by_prefix ( oss_client
_options , & bucket , & prefix );
    if ( aos_status_is_ok ( resp_status s )) {
        printf ( " delete objects by prefix succeeded \n " );
    } else {
        printf ( " delete objects by prefix failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```



Warning:

If the prefix is an empty string or null, all objects in the bucket will be deleted. Perform this operation with caution.

7.8.6 Copy objects

Copy a small-sized object

You must note the following limits before copy a small-sized object:

- The user must have the read and write permission on the source object.
- You cannot copy an object from a region to another region. For example, you cannot copy an object from a bucket in Hangzhou region to a bucket in Qingdao.
- The object size must be smaller than 1 GB.

Run the following code to copy a small-sized object:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * source_bucket_name = "< yourSourceBucketName >";
const char * source_object_name = "< yourSourceObjectName >";
const char * dest_bucket_name = "< yourDestBucketName >";
const char * dest_object_name = "< yourDestObjectName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_string_t (& options->config->endpoint , endpoint );
    aos_string_t (& options->config->access_key_id ,
    access_key_id );
    aos_string_t (& options->config->access_key_secret ,
    access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
}

```

```

    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
    pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
    options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t source_buc ket ;
    aos_string _t source_obj ect ;
    aos_string _t dest_bucke t ;
    aos_string _t dest_objec t ;
    aos_table_ t * headers = NULL ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t ( & source_buc ket , source_buc ket_name );
    aos_str_se t ( & source_obj ect , source_obj ect_name );
    aos_str_se t ( & dest_bucke t , dest_bucke t_name );
    aos_str_se t ( & dest_objec t , dest_objec t_name );
    headers = aos_table_ make ( pool , 0 );
    /* Copy an object . */
    resp_statu s = oss_copy_o bject ( oss_client _options , &
    source_buc ket , & source_obj ect , & dest_bucke t , & dest_objec
    t , headers , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " copy object succeeded \ n " );
    } else {
        printf ( " copy object failed \ n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release the allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```



```
}
```

Copy a large-sized object

For objects larger than 1 GB, we recommend that you should use

`oss_upload_part_copy` to perform multipart copy (`UploadPartCopy`). Run the following code for multipart copy:

```
# include "oss_api.h"
# include "aos_http_io.h"
# include <sys/stat.h>
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * source_bucket_name = "< yourSourceBucketName >";
const char * source_object_name = "< yourSourceObjectName >";
const char * dest_bucket_name = "< yourDestBucketName >";
const char * dest_object_name = "< yourDestObjectName >";
const char * local_file_name = "< yourLocalFilename >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates
    that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as
    timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int64_t get_file_size(const char * file_path)
{
    int64_t filesize = -1;
    struct stat statbuff;
    if (stat(file_path, &statbuff) < 0){
        return filesize;
    } else {
        filesize = statbuff.st_size;
    }
    return filesize;
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
    () to initialize global resources, such as networks
    and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK){
        exit(1);
    }
    /* Memory pool used to manage memories, which is
    equivalent to apr_pool_t. The implementation code is
    included in the apr library. */
    aos_pool_t * pool;
```

```

/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_t create (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configuration information , such
as endpoint , access_key_id , access_key_secret , is_cname
, and curl . */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to options
. */
oss_client_options = oss_request_options_create ( pool );
/* Use oss_client_options to initialize client
options . */
init_options ( oss_client_options );
/* Initialization parameters . */
aos_string_t dest_bucket ;
aos_string_t dest_object ;
aos_string_t upload_id ;
aos_table_t * headers = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
oss_list_upload_part_params_t * list_upload_part_params
;
oss_upload_part_copy_params_t * upload_part_copy_params1
;
oss_upload_part_copy_params_t * upload_part_copy_params2
;
oss_list_part_content_t * part_content ;
aos_list_t complete_part_list ;
oss_complete_part_content_t * complete_content ;
aos_table_t * list_part_resp_headers = NULL ;
aos_table_t * complete_resp_headers = NULL ;
int part1 = 1 ;
int part2 = 2 ;
int64_t range_start1 = 0 ;
int64_t range_end1 = 6000000 ;// not less than 5MB
int64_t range_start2 = 6000001 ;
int64_t range_end2 ;
int max_ret = 1000 ;
headers = aos_table_make ( pool , 0 );
aos_str_set (& dest_bucket , dest_bucket_name );
aos_str_set (& dest_object , dest_object_name );
resp_status = oss_init_multipart_upload ( oss_client_options , & dest_bucket , & dest_object , & upload_id , headers , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " init multipart upload succeeded , upload_id is %s \n " , upload_id . data );
} else {
    printf ( " init multipart upload failed \n " );
}
/* Copy the first part of the object . */
upload_part_copy_params1 = oss_create_upload_part_copy_params ( pool );
aos_str_set (& upload_part_copy_params1 -> source_bucket , source_bucket_name );
aos_str_set (& upload_part_copy_params1 -> source_object , source_object_name );
aos_str_set (& upload_part_copy_params1 -> dest_bucket , dest_bucket_name );
aos_str_set (& upload_part_copy_params1 -> dest_object , dest_object_name );

```

```

    aos_str_set (&upload_part_t_copy_part_ams1 -> upload_id ,
upload_id . data );
    upload_part_t_copy_part_ams1 -> part_num = part1 ;
    upload_part_t_copy_part_ams1 -> range_start = range_start1
;
    upload_part_t_copy_part_ams1 -> range_end = range_end1 ;
    headers = aos_table_make ( pool , 0 );
    resp_status = oss_upload_part_copy ( oss_client_options
, upload_part_t_copy_part_ams1 , headers , &resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf (" upload part 1 copy succeeded \n ");
    } else {
        printf (" upload part 1 copy failed \n ");
    }
    /* Copy the second part of the object . */
    range_end2 = get_file_size ( local_file_name ) - 1 ;
    upload_part_t_copy_part_ams2 = oss_create_upload_part_copy_params ( pool );
    aos_str_set (&upload_part_t_copy_part_ams2 -> source_bucket
, source_bucket_name );
    aos_str_set (&upload_part_t_copy_part_ams2 -> source_object
, source_object_name );
    aos_str_set (&upload_part_t_copy_part_ams2 -> dest_bucket ,
dest_bucket_name );
    aos_str_set (&upload_part_t_copy_part_ams2 -> dest_object ,
dest_object_name );
    aos_str_set (&upload_part_t_copy_part_ams2 -> upload_id ,
upload_id . data );
    upload_part_t_copy_part_ams2 -> part_num = part2 ;
    upload_part_t_copy_part_ams2 -> range_start = range_start2
;
    upload_part_t_copy_part_ams2 -> range_end = range_end2 ;
    headers = aos_table_make ( pool , 0 );
    resp_status = oss_upload_part_copy ( oss_client_options
, upload_part_t_copy_part_ams2 , headers , &resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf (" upload part 2 copy succeeded \n ");
    } else {
        printf (" upload part 2 copy failed \n ");
    }
    /* Lists the parts . */
    headers = aos_table_make ( pool , 0 );
    list_upload_part_params_ams = oss_create_list_upload_part_params ( pool );
    list_upload_part_params_ams -> max_ret = max_ret ;
    aos_list_init (&complete_part_list );
    resp_status = oss_list_upload_part ( oss_client_options
, &dest_bucket , &dest_object , &upload_id , list_upload_part_params_ams , &list_part_resp_headers );
    aos_list_for_each_entry ( oss_list_part_content_t ,
part_content , &list_upload_part_params_ams -> part_list , node )
{
        complete_content = oss_create_complete_part_content ( pool );
        aos_str_set (&complete_content -> part_number ,
part_content -> part_number . data );
        aos_str_set (&complete_content -> etag , part_content
-> etag . data );
        aos_list_append (&complete_content -> node , &
complete_part_list );
    }
    /* Complete the multipart copy . */

```

```

    resp_status s = oss_complete_multipart_upload( oss_client
_options, &dest_bucket, &dest_object, &upload_id, &
complete_part_list, headers, &complete_resp_headers );
    if ( aos_status_is_ok( resp_status s )) {
        printf (" complete multipart upload succeeded \n ");
    } else {
        printf (" complete multipart upload failed \n ");
    }
    /* Release the memory pool, that is, memories
    allocated to resources during the request. */
    aos_pool_destroy( pool );
    /* Release the allocated global resources. */
    aos_http_initialize ();
    return 0 ;
}

```

7.8.7 Restore an archive object

You must restore an archive object before you read it. Do not call `restoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

- An archive object is in the frozen state.
- After you submit it for restoration, the server restores the object. The object is in the restoring state.
- You can read the object after it is restored. The restored state of the object lasts one day by default. You can prolong this period to a maximum of seven days. Once this period expires, the object returns to the frozen state.

Run the following code to restore an object:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * object_content = " More than just cloud .";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_str_set (& options->config->endpoint, endpoint );
    aos_str_set (& options->config->access_key_id,
access_key_id );
    aos_str_set (& options->config->access_key_secret,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
}

```

```

    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_list_t buffer ;
    oss_acl_e oss_acl = OSS_ACL_PR IVATE ;
    aos_buf_t * content = NULL ;
    aos_table_ t * headers = NULL ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t ( & bucket , bucket_nam e );
    aos_str_se t ( & object , object_nam e );
    headers = aos_table_ make ( pool , 0 );
    /* Create a bucket of the archive storage class .
*/
    resp_statu s = oss_create _bucket_wi th_storage _class (
oss_client _options , & bucket , oss_acl , OSS_STORAG E_CLASS_AR
CHIVE , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " create bucket succeeded \ n " );
    } else {
        printf ( " create bucket failed \ n " );
    }
    aos_list_i nit ( & buffer );
    content = aos_buf_pa ck ( oss_client _options -> pool ,
object_con tent , strlen ( object_con tent ));
    aos_list_a dd_tail ( & content -> node , & buffer );
    /* Upload a file . */
    resp_statu s = oss_put_ob ject_from_ buffer ( oss_client
_options , & bucket , & object , & buffer , headers , & resp_heade
rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " put object from buffer succeeded \ n " );
    } else {
        printf ( " put object from buffer failed \ n " );
    }
}

```

```

    }
    /* Restore the object . */
    do {
        headers = aos_table_make ( pool , 0 );
        resp_status = oss_restore_object ( oss_client_options
, & bucket , & object , headers , & resp_headers );
        printf ( " restore object resp_status -> code : % d \ n
", resp_status -> code );
        if ( resp_status -> code != 409 ) {
            break ;
        } else {
            printf ( " restore object is already in progress
, resp_status -> code : % d \ n", resp_status -> code );
            apr_sleep ( 5000 );
        }
    } while ( 1 );
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

For more information about the archive storage classes, see [Introduction to storage classes](#). For more information about related status code, see [RestoreObject](#).

7.8.8 Manage a symbolic link

Create a symbolic link

A symbolic link is a special object that maps to an object similar to a shortcut used in Windows. You can configure user-defined object meta for symbolic links.

Run the following code to create a symbolic link:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * link_object_name = "< yourLinkObjectName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize the aos_string_t
type . */
    aos_string_t (& options->config->endpoint , endpoint );
    aos_string_t (& options->config->access_key_id ,
access_key_id );
    aos_string_t (& options->config->access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
}

```

```

    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate ( & pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_string _t object ;
    aos_string _t sym_object ;
    aos_table _t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t ( & bucket , bucket_nam e );
    aos_str_se t ( & object , object_nam e );
    aos_str_se t ( & sym_object , link_objec t_name );
    resp_statu s = oss_put_symlink ( oss_client _options , &
bucket , & sym_object , & object , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s ) ) {
        printf ( " put symlink succeeded \ n " );
    } else {
        printf ( " put symlink failed \ n " );
    }
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Obtain object content for a symbolic link

To obtain a symbolic link, you must have the read permission on it. Run the following code to obtain the object content for a symbolic link:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";

```

```

const char * access_key_id = "< yourAccess KeyId >";
const char * access_key_secret = "< yourAccess KeySecret >";
const char * bucket_name = "< yourBucket Name >";
const char * link_object_name = "< yourLinkObject Name >";
void init_options (oss_request_options_t * options)
{
    options->config = oss_config_create (options->pool);
    /* Use a char * string to initialize the aos_string
    _t type . */
    aos_string_t (& options->config->endpoint, endpoint);
    aos_string_t (& options->config->access_key_id,
    access_key_id);
    aos_string_t (& options->config->access_key_secret,
    access_key_secret);
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options->config->is_cname = 0;
    /* Used to configure network parameters , such as
    timeout . */
    options->ctl = aos_http_controller_create (options->
    pool, 0);
}
int main (int argc, char * argv[])
{
    /* Call the aos_http_initialize method in main
    () to initialize global resources , such as networks
    and memories . */
    if (aos_http_initialize (NULL, 0) != AOSE_OK) {
        exit (1);
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_create (& pool, NULL);
    /* Create and initialize options . This parameter
    mainly includes global configuration information , such
    as endpoint , access_key_id , access_key_secret , is_cname
    , and curl . */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create (pool);
    /* Use oss_client_options to initialize client
    options */
    init_options (oss_client_options);
    /* Initialization parameters . */
    aos_string_t bucket;
    aos_string_t link_object;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_status = NULL;
    char * target_object_name = NULL;
    aos_string_t (& bucket, bucket_name);
    aos_string_t (& link_object, link_object_name);
    resp_status = oss_get_symlink (oss_client_options, &
    bucket, & link_object, & resp_headers);
    if (aos_status_is_ok (resp_status)) {
        printf ("get symlink succeeded \n");
        target_object_name = (char *) (apr_table_get (
    resp_headers, OSS_CANNONICALIZED_HEADER_SYMLINK));
    }
}

```



```

        printf (" target_obj  ect_name : % s \ n ", target_obj
ect_name );
    } else {
        printf (" get    symlink    failed \ n ");
    }
    /* Release    the    memory    pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release    allocated    global    resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

7.9 Authorized access

Use STS for temporary access authorization

OSS supports Alibaba Cloud Security Token Service (STS) for temporary access authorization. STS is a web service that provides a temporary access token to a cloud computing user. Through the STS, you can assign a third-party application or a RAM user (you can manage the user ID) an access credential with a custom validity period and permissions. For more information about STS, see [STS introduction](#).

STS advantages:

- Your long-term key (AccessKey) is not exposed to a third-party application. You only need to generate an access token and send the access token to the third-party application. You can customize access permissions and the validity of this token.
- You do not need to keep track of permission revocation issues. The access token automatically becomes invalid when it expires.

For more information about the process of access to OSS with STS, see [RAM and STS scenario practices](#) in OSS Developer Guide.

Create a signature request with STS

Run the following code to create a signature request with STS:

```

# include " oss_api . h "
# include " aos_http_i o . h "

const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * sts_token = "< yourStsTok en >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_nam e = "< yourObject Name >";
const char * object_con tent = " More than just cloud .";

void init_optio ns ( oss_reques t_options_ t * options )
{

```

```

options -> config = oss_config _create ( options -> pool );
/* Use a char * string to initialize the aos_string _t
   type . */
aos_str_se t (& options -> config -> endpoint , endpoint );
aos_str_se t (& options -> config -> access_key _id , access_key
_id );
aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
aos_str_se t (& options -> config -> sts_token , sts_token );
/* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
options -> config -> is_cname = 0 ;
/* Used to configure network parameters , such as
timeout . */
options -> ctl = aos_http_c ontroller_ create ( options -> pool
, 0 );
}

int main ( int argc , char * argv [])
{
/* Call the aos_http_i o_initiali ze method in main ()
to initialize global resources , such as networks and
memories . */
if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
    exit ( 1 );
}

/* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
aos_pool_t * pool ;
/* Re - create a new memory pool . The second parameter
is NULL , indicating that it does not inherit from
any other memory pools . */
aos_pool_c reate (& pool , NULL );
/* Create and initialize options . This parameter mainly
includes global configurat ion informatio n , such as
endpoint , access_key _id , acces_key_ secret , is_cname , and
curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate memories in the memory pool to options .
*/
oss_client _options = oss_reques t_options_ create ( pool );
/* Use oss_client _options to initialize client options .
*/
init_optio ns ( oss_client _options );

/* Initializa tion parameters . */
aos_string _t bucket ;
aos_string _t object ;
aos_list_t buffer ;
aos_buf_t * content = NULL ;
aos_table_ t * headers = NULL ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
/* Assign the char * data to the bucket . */
aos_str_se t (& bucket , bucket_nam e );
aos_str_se t (& object , object_nam e );

aos_list_i nit (& buffer );
content = aos_buf_pa ck ( oss_client _options -> pool ,
object_con tent , strlen ( object_con tent ));
aos_list_a dd_tail (& content -> node , & buffer );

```

```

/* Upload an object . */
resp_status s = oss_put_object_from_buffer ( oss_client_options , & bucket , & object , & buffer , headers , & resp_headers );
/* Determine whether the upload is successful . */
if ( aos_status_is_ok ( resp_status s )) {
    printf ( " put object from buffer succeeded \ n " );
} else {
    printf ( " put object from buffer failed \ n " );
}

/* Release the memory pool , that is , memories allocated to resources during the request . */
aos_pool_destroy ( pool );

/* Release allocated global resources . */
aos_http_info_deinit ();

return 0 ;
}

```

Sign a URL to authorize temporary access

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time for a URL to restrict the period of access from visitors.

- Use a signed URL to upload an object

Run the following code to upload an object with a signed URL:

```

# include " oss_api . h "
# include " aos_http_info . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
const char * local_filename = "< yourLocalFilename >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize the
    aos_string_t type . */
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id ,
    access_key_id );
    aos_str_set (& options->config->access_key_secret ,
    access_key_secret );
    /* Determine whether the CNAME is used . 0
    indicates that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
    timeout . */
    options->ctl = aos_http_controller_create ( options->pool , 0 );
}
int main ( int argc , char * argv [])
{

```

```

/* Call the aos_http_initialize method in main
() to initialize global resources, such as networks
and memories. */
if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
    exit ( 1 );
}
/* Memory pool used to manage memories, which is
equivalent to apr_pool_t. The implementation code
is included in the apr library. */
aos_pool_t * pool ;
/* Re - create a new memory pool. The second
parameter is NULL, indicating that it does not
inherit from any other memory pools. */
aos_pool_create ( & pool , NULL );
/* Create and initialize options. This parameter
mainly includes global configuration information,
such as endpoint, access_key_id, access_key_secret,
is_cname, and curl. */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to
options. */
oss_client_options = oss_request_options_create ( pool
);
/* Use oss_client_options to initialize client
options. */
init_options ( oss_client_options );
/* Initialization parameters. */
aos_string_t bucket ;
aos_string_t object ;
aos_string_t file ;
aos_table_t * headers = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_http_request_t * req ;
apr_time_t now ;
char * url_str ;
aos_string_t url ;
int64_t expire_time ;
int one_hour = 3600 ;
aos_str_set ( & bucket , bucket_name );
aos_str_set ( & object , object_name );
aos_str_set ( & file , local_file_name );
expire_time = now / 1000000 + one_hour ;
headers = aos_table_make ( pool , 0 );
req = aos_http_request_create ( pool );
req -> method = HTTP_PUT ;
now = apr_time_now (); /* Unit: microseconds */
expire_time = now / 1000000 + one_hour ;
/* Generate the signed URL. */
url_str = oss_generate_url ( oss_client_options , &
bucket , & object , expire_time , req );
aos_str_set ( & url , url_str );
printf ( " Temporary upload URL : %s \n " , url_str );
/* Use the signed URL to upload an object. */
resp_status = oss_put_object_from_file_by_url (
oss_client_options , & url , & file , headers , & resp_headers
);
if ( aos_status_is_ok ( resp_status ) ) {
    printf ( " put objects by signed url succeeded \n
");
} else {
    printf ( " put objects by signed url failed \n
");
}
}

```

```

/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_d estroy ( pool );
/* Release allocated global resources . */
aos_http_i o_deinitia lize ();
return 0 ;
}

```

- Use a signed URL to download an object

Run the following code to upload an object with a signed URL:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
const char * object_nam e = "< yourObject Name >";
const char * local_file name = "< yourLocalF ilename >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize the
aos_string _t type . */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0
indicates that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code
is included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& p , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n ,
such as endpoint , access_key _id , acces_key_ secret ,
is_cname , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to
options . */
    oss_client _options = oss_reques t_options_ create ( pool
);
}

```

```

/* Use oss_client_options to initialize client
options . */
init_options ( oss_client_options );
/* Initialization parameters . */
aos_string_t bucket ;
aos_string_t object ;
aos_string_t file ;
aos_table_t * headers = NULL ;
aos_table_t * params = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_http_request_t * req ;
apr_time_t now ;
char * url_str ;
aos_string_t url ;
int64_t expire_time ;
int one_hour = 3600 ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );
aos_str_set (& file , local_file_name );
expire_time = now / 1000000 + one_hour ;
headers = aos_table_make ( pool , 0 );
params = aos_table_make ( pool , 0 );
req = aos_http_request_create ( pool );
req->method = HTTP_GET ;
now = apr_time_now (); /* Unit : microseconds */
expire_time = now / 1000000 + one_hour ;
/* Generate the signed URL . */
url_str = oss_generate_url ( oss_client_options , &
bucket , & object , expire_time , req );
aos_str_set (& url , url_str );
/* Use the signed URL to download an object . */
resp_status = oss_get_object_to_file_by_url (
oss_client_options , & url , headers , params , & file , &
resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get object succeeded \n " );
} else {
    printf ( " get object failed \n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

7.10 Manage lifecycle rules

You can use OSS to configure lifecycle rules to reduce storage costs. The lifecycle management rules can be used for automatic deletion of expired objects and fragments, or storage class conversion to Infrequent Access for objects that will expire soon. Each rule includes:

- **Rule ID:** It identifies a rule. Ensure that each rule ID in a bucket is unique.

- **Policy:** You can configure a policy by using the following configuration methods.
Only one method can be configured for a bucket.
 - **Configure by Prefix:** You can create multiple rules with this method. Ensure that each prefix is unique.
 - **Configure for Entire Bucket:** You can configure only one rule with this method.
- **Expired:** You can configure the following configuration methods:
 - **Set by Number of Days:** It specifies the number of days from the last modification time of objects.
 - **Set by Date:** It specifies the date prior to which objects are created. If objects are created prior to the date, these objects are deleted.
- **Status:** This lifecycle rule takes effect or not.

The parts uploaded by uploadPart method also support lifecycle rules. The last modification time of an object is the time the multipart upload event is initiated.

For more information about lifecycle management, see [Manage object lifecycle](#).

Configure lifecycle rules

Run the following code to configure lifecycle rules:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates
that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Used to configure network parameters, such as
timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
() to initialize global resources, such as networks
and memories. */
    if(aos_http_io_initialize(NULL, 0) != AOSE_OK) {
```

```

        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementation code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_create ( & pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configuration information , such
    as endpoint , access_key_id , access_key_secret , is_cname
    , and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
    options . */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    aos_str_set ( & bucket , bucket_name );
    aos_list_t lifecycle_rule_list ;
    /* Create a lifecycle rule for the bucket . */
    aos_str_set ( & bucket , bucket_name );
    aos_list_init ( & lifecycle_rule_list );
    /* Specify the expiration duration . */
    oss_lifecycle_rule_content_t * rule_content_days =
    oss_create_lifecycle_rule_content ( pool );
    aos_str_set ( & rule_content_days -> id , " rule - 1 " );
    aos_str_set ( & rule_content_days -> prefix , " obsoleted " );
    aos_str_set ( & rule_content_days -> status , " Enabled " );
    rule_content_days -> days = 3 ;
    aos_list_add_tail ( & rule_content_days -> node , &
    lifecycle_rule_list );
    /* Specify the expiration date . */
    oss_lifecycle_rule_content_t * rule_content_date =
    oss_create_lifecycle_rule_content ( pool );
    aos_str_set ( & rule_content_date -> id , " rule - 2 " );
    aos_str_set ( & rule_content_date -> prefix , " delete " );
    aos_str_set ( & rule_content_date -> status , " Enabled " );
    aos_str_set ( & rule_content_date -> date , " 2022 - 10 -
    11T00 : 00 : 00 . 000Z " );
    aos_list_add_tail ( & rule_content_date -> node , &
    lifecycle_rule_list );
    /* Configure a lifecycle rule . */
    resp_status = oss_put_bucket_lifecycle ( oss_client_options , & bucket , & lifecycle_rule_list , & resp_headers );
    if ( aos_status_is_ok ( resp_status ) ) {
        printf ( " put bucket lifecycle succeeded \n " );
    } else {
        printf ( " put bucket lifecycle failed , code : %d ,
        error_code : %s , error_msg : %s , request_id : %s \n " ,
        resp_status -> code , resp_status -> error_code ,
        resp_status -> error_msg , resp_status -> req_id );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release allocated global resources . */

```



```

    aos_http_initialize ();
    return 0 ;
}

```

View lifecycle rules

Run the following code to view lifecycle rules:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
    */
    aos_string_t (& options->config->endpoint , endpoint );
    aos_string_t (& options->config->access_key_id ,
access_key_id );
    aos_string_t (& options->config->access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options->ctl = aos_http_controller_create ( options->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementation code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configuration information , such
as endpoint , access_key_id , access_key_secret , is_cname
, and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client_options = oss_request_options_create ( pool );
    /* Use oss_client_options to initialize client
options . */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_table_t * resp_headers = NULL ;

```

```

aos_status_t * resp_status = NULL ;
aos_list_t lifecycle_rule_list ;
oss_lifecycle_rule_content_t * rule_content ;
char * rule_id ;
char * prefix ;
char * status ;
int days = INT_MAX ;
char * date = "" ;
aos_string_t (& bucket , bucket_name );
/* Obtain and print the lifecycle of the bucket .
*/
aos_list_init (& lifecycle_rule_list );
resp_status = oss_get_bucket_lifecycle ( oss_client_options , & bucket , & lifecycle_rule_list , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get bucket lifecycle succeeded \n " );
    aos_list_for_each_entry ( oss_lifecycle_rule_content_t , rule_content , & lifecycle_rule_list , node ) {
        rule_id = apr_psprintf ( pool , "%.s " ,
rule_content->id . len , rule_content->id . data );
        prefix = apr_psprintf ( pool , "%.s " , rule_content->prefix . len , rule_content->prefix . data );
        status = apr_psprintf ( pool , "%.s " , rule_content->status . len , rule_content->status . data );
        date = apr_psprintf ( pool , "%.s " , rule_content->date . len , rule_content->date . data );
        days = rule_content->days ;
    }
    printf ( " rule_id : %s \n " , rule_id );
    printf ( " prefix : %s \n " , prefix );
    printf ( " status : %s \n " , status );
    printf ( " date : %s \n " , date );
    printf ( " days : %d \n " , days );
} else {
    printf ( " get bucket lifecycle failed , code : %d ,
error_code : %s , error_msg : %s , request_id : %s \n " ,
resp_status->code , resp_status->error_code ,
resp_status->error_msg , resp_status->req_id );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

Clear lifecycle rules

Run the following code to clear lifecycle rules:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
*/

```

```

    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Used to configure network parameters , such as
timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementa tion code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Use oss_client _options to initialize client
options . */
    init_optio ns ( oss_client _options );
    /* Initializa tion parameters . */
    aos_string _t bucket ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    aos_str_se t (& bucket , bucket_nam e );
    /* Delete bucket lifecycle rules . */
    aos_str_se t (& bucket , bucket_nam e );
    resp_statu s = oss_delete _bucket_li fecycle ( oss_client
_options , & bucket , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " delete bucket lifecycle succeeded \ n " );
    } else {
        printf ( " delete bucket lifecycle failed , code :% d
, error_code :% s , error_msg :% s , request_id :% s \ n " ,
resp_statu s -> code , resp_statu s -> error_code ,
resp_statu s -> error_msg , resp_statu s -> req_id );
    }
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

```
}
```

7.11 Image processing

Image Processing (IMG) is a massive, secure, cost-effective and highly reliable image processing service. After source images are uploaded to OSS, you can process images on any Internet device at any anytime, from anywhere through simple RESTful APIs.

For more information about IMG, see [Image Processing](#).

For the complete code of IMG, see [GitHub](#).

Basic features

OSS offers the following IMG features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Image processing](#)
- [Image processing access rules](#): calls multiple IMG functions.

Usage

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- Anonymous access

You can use the following format in level-3 domains to access a processed image:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Specifies the name of a bucket.
endpoint	Specifies endpoint used to access a region.
object	Specifies the name of an image object.
image	Specifies the reserved identifier for IMG .
action	Specifies the operations on an image, such as scaling, cropping, and rotating.

Parameter	Description
param	Specifies the parameter that indicates the operation on an image.

- Basic operations

For example, scale an image to a width of 100 px. Adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customized image styles

You can use the following format in level-3 domains to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- **style**: specifies a reserved identifier of a customized image style.
- **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image sequentially.

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- HTTPS access

IMG supports access through HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

Use the following code to generate a signed URL for IMG:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options (oss_request_options_t * options)
{
    options->config = oss_config_create (options->pool);
    /* Use a char * string to initialize aos_string_t. */
    aos_str_set (& options->config->endpoint, endpoint);
    aos_str_set (& options->config->access_key_id,
        access_key_id);
    aos_str_set (& options->config->access_key_secret,
        access_key_secret);
    /* Determine whether the CNAME is used. The value
       0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create (options->
        pool, 0);
}
int main (int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
       () to initialize global resources, such as networks
       and memories. */
    if (aos_http_io_initialize (NULL, 0) != AOSE_OK) {
        exit (1);
    }
    /* Memory pool used to manage memories, equivalent
       to apr_pool_t. The implementation code is included
       in the apr library. */
    aos_pool_t * pool;
    /* Re-create a memory pool. The second parameter
       is NULL, indicating that the new pool does not
       inherit any other memory pool. */
    aos_pool_create (& pool, NULL);
    /* Create and initialize options. This parameter
       includes global configuration information, such as
       endpoint, access_key_id, access_key_secret, is_cname,
       and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate the memories in the memory pool to
       options. */
```

```

    oss_client_options = oss_request_options_create ( pool
);
/* Initialize oss_client_options . */
init_options ( oss_client_options );
/* Initialize parameters . */
aos_string_t bucket ;
aos_string_t object ;
aos_table_t * params = NULL ;
aos_http_request_t * req ;
char * url_str ;
apr_time_t now ;
int64_t expire_time ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );
/* Start image processing . */
params = aos_table_make ( pool , 1 );
apr_table_set ( params , OSS_PROCESS , " image / resize ,
m_fixed , w_100 , h_100 " );
req = aos_http_request_create ( pool );
req->method = HTTP_GET ;
req->query_params = params ;
/* Specify the expiration time ( expire_time ) in
seconds . */
now = apr_time_now ();
expire_time = now / 1000000 + 10 * 60 ;
/* Generate a signed URL . */
url_str = oss_generate_url ( oss_client_options , &
bucket , & object , expire_time , req );
printf ( " url : % s \ n " , url_str );
/* Release the memory pool , that is , the memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_initialize ();
return 0 ;
}

```

- Access with SDK

You can use SDK to access and process an image object.

SDK allows you to customize image styles, HTTPS access, and cascade operations.

- Basic operations

Run the following code to perform basic operations on an image:

```

# include " oss_api . h "
# include " aos_http_initialize . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool
);
    /* Use a char * string to initialize aos_string
    _t . */
    aos_str_set (& options->config->endpoint , endpoint );
}

```



```

        aos_str_set (& options -> config -> access_key_id ,
        access_key_id );
        aos_str_set (& options -> config -> access_key_secret ,
        access_key_secret );
        /* Determine whether the CNAME is used . The
        value 0 indicates that the CNAME is not used . */
        options -> config -> is_cname = 0 ;
        /* Configure network parameters , such as timeout .
        */
        options -> ctl = aos_http_controller_create ( options -
        > pool , 0 );
    }
    int main ( int argc , char * argv [])
    {
        /* Call the aos_http_initialize method in
        main () to initialize global resources , such as
        networks and memories . */
        if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK )
        {
            exit ( 1 );
        }
        /* Memory pool used to manage memories ,
        equivalent to apr_pool_t . The implementation code
        is included in the apr library . */
        aos_pool_t * pool ;
        /* Re - create a memory pool . The second
        parameter is NULL , indicating that the new pool
        does not inherit any other memory pool . */
        aos_pool_create (& pool , NULL );
        /* Create and initialize options . This parameter
        includes global configuration information , such as
        endpoint , access_key_id , access_key_secret , is_cname
        , and curl . */
        oss_request_options_t * oss_client_options ;
        /* Allocate memories in the memory pool to
        options . */
        oss_client_options = oss_request_options_create (
        pool );
        /* Initialize oss_client_options . */
        init_options ( oss_client_options );
        /* Initialize parameters . */
        aos_string_t bucket ;
        aos_string_t object ;
        aos_string_t file ;
        aos_table_t * headers = NULL ;
        aos_table_t * params = NULL ;
        aos_table_t * resp_headers = NULL ;
        aos_status_t * resp_statuses = NULL ;
        aos_str_set (& bucket , bucket_name );
        aos_str_set (& object , object_name );
        /* Scale the image .
        params = aos_table_make ( pool , 1 );
        apr_table_set ( params , OSS_PROCESSES , " image / resize
        , m_fixed , w_100 , h_100 " );
        /* Download processed images to a local file . */
        aos_str_set (& file , " example - resize . jpg " );
        resp_statuses = oss_get_object_to_file ( oss_client
       _options , & bucket , & object , headers , params , & file , &
        resp_headers );
        if ( aos_status_is_ok ( resp_statuses )) {
            printf ( " get object to file succeeded \ n " );
        } else {
            printf ( " get object to file failed \ n " );
        }
    }

```

```

/* Crop the image .
params = aos_table_ make ( pool , 1 );
apr_table_ set ( params , OSS_PROCES S , " image / crop ,
w_100 , h_100 , x_100 , y_100 , r_1 " );
/* Download processed images to a local file . */
aos_str_se t (& file , " example - crop . jpg " );
resp_statu s = oss_get_ob ject_to_fi le ( oss_client
_options , & bucket , & object , headers , params , & file , &
resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " get object to file succeeded \ n " );
} else {
printf ( " get object to file failed \ n " );
}
/* Rotate the image .
params = aos_table_ make ( pool , 1 );
apr_table_ set ( params , OSS_PROCES S , " image / rotate
, 90 " );
/* Download processed images to a local file . */
aos_str_se t (& file , " example - rotate . jpg " );
resp_statu s = oss_get_ob ject_to_fi le ( oss_client
_options , & bucket , & object , headers , params , & file , &
resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " get object to file succeeded \ n " );
} else {
printf ( " get object to file failed \ n " );
}
/* Sharpen the image .
params = aos_table_ make ( pool , 1 );
apr_table_ set ( params , OSS_PROCES S , " image / sharpen
, 100 " );
/* Download processed images to a local file . */
aos_str_se t (& file , " example - sharpen . jpg " );
resp_statu s = oss_get_ob ject_to_fi le ( oss_client
_options , & bucket , & object , headers , params , & file , &
resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " get object to file succeeded \ n " );
} else {
printf ( " get object to file failed \ n " );
}
/* Add watermarks .
params = aos_table_ make ( pool , 1 );
apr_table_ set ( params , OSS_PROCES S , " image /
watermark , text_SGVsb G8g5Zu - 54mH5pyN5Y qhIQ " );
/* Download processed images to a local file . */
aos_str_se t (& file , " example - watermark . jpg " );
resp_statu s = oss_get_ob ject_to_fi le ( oss_client
_options , & bucket , & object , headers , params , & file , &
resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " get object to file succeeded \ n " );
} else {
printf ( " get object to file failed \ n " );
}
/* Convert the image format .
params = aos_table_ make ( pool , 1 );
apr_table_ set ( params , OSS_PROCES S , " image / format
, png " );
/* Download processed images to a local file . */
aos_str_se t (& file , " example - format . png " );

```

```

    resp_status s = oss_get_object_to_file ( oss_client
_options, & bucket, & object, headers, params, & file, &
resp_headers );
    if ( aos_status_is_ok ( resp_status s )) {
        printf (" get object to file succeeded \n ");
    } else {
        printf (" get object to file failed \n ");
    }
    /* Obtain image information .
    params = aos_table_ make ( pool, 1 );
    apr_table_ set ( params, OSS_PROCES S, " image / info
");
    /* Download processed images to a local file . */
    aos_str_set (& file, " example - info . txt ");
    resp_status s = oss_get_object_to_file ( oss_client
_options, & bucket, & object, headers, params, & file, &
resp_headers );
    if ( aos_status_is_ok ( resp_status s )) {
        printf (" get object to file succeeded \n ");
    } else {
        printf (" get object to file failed \n ");
    }
    /* Release the memory pool, that is, the
    memories allocated to resources during the request .
    */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_io_deinitialize ();
    return 0 ;
}

```

- Customize an image style

Run the following code to customize an image style:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccess KeyId >";
const char * access_key_secret = "< yourAccess KeySecret
>";
const char * bucket_name = "< yourBucket Name >";
const char * object_name = "< yourObject Name >";
void init_options ( oss_request_options_t * options )
{
    options-> config = oss_config_create ( options-> pool
);
    /* Use a char * string to initialize aos_string
_t . */
    aos_str_set (& options-> config-> endpoint, endpoint );
    aos_str_set (& options-> config-> access_key_id,
access_key_id );
    aos_str_set (& options-> config-> access_key_secret,
access_key_secret );
    /* Determine whether the CNAME is used . The
value 0 indicates that the CNAME is not used . */
    options-> config-> is_cname = 0 ;
    /* Configure network parameters, such as timeout .
    */
    options-> ctl = aos_http_controller_create ( options -
> pool, 0 );
}
int main ( int argc, char * argv [])

```

```

{
    /* Call the aos_http_initialize method in
    main() to initialize global resources, such as
    networks and memories. */
    if (aos_http_initialize ( NULL , 0 ) != AOSE_OK )
    {
        exit ( 1 );
    }
    /* Memory pool used to manage memories,
    equivalent to apr_pool_t. The implementation code
    is included in the apr library. */
    aos_pool_t * pool ;
    /* Re - create a memory pool. The second
    parameter is NULL, indicating that the new pool
    does not inherit any other memory pool. */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options. This parameter
    includes global configuration information, such as
    endpoint, access_key_id, access_key_secret, is_cname
    , and curl. */
    oss_request_options_t * oss_client_options ;
    /* Allocate the memories in the memory pool to
    options. */
    oss_client_options = oss_request_options_create (
pool );
    /* Initialize oss_client_options. */
    init_options ( oss_client_options );
    /* Initialize parameters. */
    aos_string_t bucket ;
    aos_string_t object ;
    aos_string_t file ;
    aos_table_t * headers = NULL ;
    aos_table_t * params = NULL ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_statuses = NULL ;
    aos_str_set (& bucket , bucket_name );
    aos_str_set (& object , object_name );
    /* Customize the image style. */
    params = aos_table_make ( pool , 1 );
    apr_table_set ( params , OSS_PROCESS , " style /<
yourCustom StyleName >" );
    /* Download processed images to a local file. */
    aos_str_set (& file , " example - custom - style . jpg " );
    resp_statuses = oss_get_object_to_file ( oss_client
_options , & bucket , & object , headers , params , & file , &
resp_headers );
    if ( aos_status_is_ok ( resp_statuses )) {
        printf ( " get object to file succeeded \n " );
    } else {
        printf ( " get object to file failed \n " );
    }
    /* Release the memory pool, that is, the
    memories allocated to resources during the request.
    */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources. */
    aos_http_initialize ();
    return 0 ;
}

```

```
}

```

- Cascade operations

Run the following code to perform cascade operations on an image:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * object_name = "< yourObjectName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t. */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id, access_key_id);
    aos_str_set(&options->config->access_key_secret, access_key_secret);
    /* Determine whether the CNAME is used. The value 0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main() to initialize global resources, such as networks and memories. */
    if(aos_http_io_initialize(NULL, 0) != AOSE_OK)
    {
        exit(1);
    }
    /* Memory pool used to manage memories, equivalent to apr_pool_t. The implementation code is included in the apr library. */
    aos_pool_t * pool;
    /* Re-create a memory pool. The second parameter is NULL, indicating that the new pool does not inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter includes global configuration information, such as endpoint, access_key_id, access_key_secret, is_cname, and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate the memories in the memory pool to options. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialize parameters. */
    aos_string_t bucket;
```

```

aos_string_t object ;
aos_string_t file ;
aos_table_t * headers = NULL ;
aos_table_t * params = NULL ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_str_set (& bucket , bucket_name );
aos_str_set (& object , object_name );
/* Perform cascade operations . */
params = aos_table_make ( pool , 1 );
apr_table_set ( params , OSS_PROCESSES , " image / resize
, m_fixed , w_100 , h_100 ");
/* Download processed images to a local file . */
aos_str_set (& file , " example - cascade . jpg ");
resp_status = oss_get_object_to_file ( oss_client
_options , & bucket , & object , headers , params , & file , &
resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get object to file succeeded \ n ");
} else {
    printf ( " get object to file failed \ n ");
}
/* Release the memory pool , that is , the
memories allocated to resources during the request .
*/
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

IMG tools

You can use the IMG viewer [ImageStyleViewer](#) to directly view the IMG result.

7.12 CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) and [PutBucketcors](#) in OSS Developer Guide.

Configure CORS rules

Run the following code to configure CORS rules for a specified bucket:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
}

```

```

/* Use a char * string to initialize aos_string_t .
*/
aos_str_set (& options -> config -> endpoint , endpoint );
aos_str_set (& options -> config -> access_key_id ,
access_key_id );
aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
/* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
options -> config -> is_cname = 0 ;
/* Configure network parameters , such as timeout . */
options -> ctl = aos_http_controller_create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
/* Call the aos_http_initialize method in main
() to initialize global resources , such as networks
and memories . */
if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
exit ( 1 );
}
/* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementation code is
included in the apr library . */
aos_pool_t * pool ;
/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_create (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configuration information , such
as endpoint , access_key_id , access_key_secret , is_cname
, and curl . */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to options
. */
oss_client_options = oss_request_options_create ( pool );
/* Initialize oss_client_options . */
init_options ( oss_client_options );
/* Initialization parameters . */
aos_string_t bucket ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_list_t cors_rule_list ;
oss_cors_rule_t * cors_rule1 = NULL , * cors_rule2 = NULL
;
aos_str_set (& bucket , bucket_name );
aos_list_init (& cors_rule_list );
cors_rule1 = oss_create_cors_rule ( pool );
aos_list_add_tail (& cors_rule1 -> node , & cors_rule_list );
oss_create_sub_cors_rule ( pool , & cors_rule1 -> allowed_or
igin_list , " allowed_or_igin_1_1 " );
oss_create_sub_cors_rule ( pool , & cors_rule1 -> allowed_or
igin_list , " allowed_or_igin_1_1 " );
oss_create_sub_cors_rule ( pool , & cors_rule1 -> allowed_me
thod_list , " PUT " );
oss_create_sub_cors_rule ( pool , & cors_rule1 -> allowed_me
thod_list , " GET " );
oss_create_sub_cors_rule ( pool , & cors_rule1 -> allowed_he
ad_list , " Authorization " );
oss_create_sub_cors_rule ( pool , & cors_rule1 -> expose_he
ad_list , " expose_header_1_1 " );

```

```

    oss_create_sub_cors_rule ( pool , & cors_rule1 -> expose_header_list , " expose_header_1_1 " );
    cors_rule2 = oss_create_cors_rule ( pool );
    aos_list_append (& cors_rule2 -> node , & cors_rule_list );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> allowed_origin_list , " allowed_origin_2_1 " );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> allowed_origin_list , " allowed_origin_2_2 " );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> allowed_methods_list , " PUT " );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> allowed_methods_list , " GET " );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> allowed_headers_list , " Authorization " );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> expose_header_list , " expose_header_2_1 " );
    oss_create_sub_cors_rule ( pool , & cors_rule2 -> expose_header_list , " expose_header_2_2 " );
    /* Configure the CORS rule . */
    resp_status = oss_put_bucket_cors ( oss_client_options ,
    & bucket , & cors_rule_list , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " put bucket cors succeeded \n " );
    } else {
        printf ( " put bucket cors failed \n " );
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_io_deinitialize ();
    return 0 ;
}

```

Obtain CORS rules

Run the following code to obtain CORS rules:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options -> config = oss_config_create ( options -> pool );
    /* Use a char * string to initialize aos_string_t . */
    aos_str_set (& options -> config -> endpoint , endpoint );
    aos_str_set (& options -> config -> access_key_id ,
    access_key_id );
    aos_str_set (& options -> config -> access_key_secret ,
    access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options -> ctl = aos_http_controller_create ( options ->
    pool , 0 );
}
int main ( int argc , char * argv [])

```



```

{
    /* Call the aos_http_initialize method in main
    () to initialize global resources, such as networks
    and memories. */
    if (aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories, which is
    equivalent to apr_pool_t. The implementation code is
    included in the apr library. */
    aos_pool_t * pool;
    /* Re - create a new memory pool. The second
    parameter is NULL, indicating that it does not
    inherit from any other memory pools. */
    aos_pool_create (& pool, NULL);
    /* Create and initialize options. This parameter
    mainly includes global configuration information, such
    as endpoint, access_key_id, access_key_secret, is_cname
    , and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create ( pool );
    /* Initialize oss_client_options. */
    init_options ( oss_client_options );
    /* Initialization parameters. */
    aos_string_t bucket;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_status = NULL;
    aos_list_t cors_rule_list;
    oss_corss_rule_t * cors_rule = NULL;
    oss_sub_corss_rule_t * sub_corss_rule = NULL;
    aos_string_t (& bucket, bucket_name);
    /* Obtain CORS rules. */
    aos_list_init (& cors_rule_list);
    resp_status = oss_get_bucket_cors ( oss_client_options ,
    & bucket , & cors_rule_list , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf (" get bucket cors succeeded \n ");
        aos_list_for_each_entry ( oss_corss_rule_t , cors_rule
    , & cors_rule_list , node ) {
            printf (" max_age_seconds : % d \n ", cors_rule ->
    max_age_seconds );
            aos_list_for_each_entry ( oss_sub_corss_rule_t ,
    sub_corss_rule , & cors_rule -> allowed_or_igin_list , node ) {
                printf (" allowed_or_igin_list : % s \n ",
    sub_corss_rule -> rule . data );
            }
            aos_list_for_each_entry ( oss_sub_corss_rule_t ,
    sub_corss_rule , & cors_rule -> allowed_method_list , node ) {
                printf (" allowed_method_list : % s \n ",
    sub_corss_rule -> rule . data );
            }
            aos_list_for_each_entry ( oss_sub_corss_rule_t ,
    sub_corss_rule , & cors_rule -> allowed_header_list , node ) {
                printf (" allowed_header_list : % s \n ",
    sub_corss_rule -> rule . data );
            }
            aos_list_for_each_entry ( oss_sub_corss_rule_t ,
    sub_corss_rule , & cors_rule -> expose_header_list , node ) {
                printf (" expose_header_list : % s \n ",
    sub_corss_rule -> rule . data );
            }
        }
    }
}

```

```

    } else {
        printf (" get    bucket    cors    failed \n ");
    }
    /* Release the memory pool , that is , memories
    allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release the allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

Delete CORS rules

Run the following code to delete all CORS rules for the specified bucket:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
    access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
    access_key _secret );
    /* Determine whether the CNAME is used . 0 indicates
    that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
    pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_i o_initiali ze method in main
    () to initialize global resources , such as networks
    and memories . */
    if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
    equivalent to apr_pool_t . The implementa tion code is
    included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
    parameter is NULL , indicating that it does not
    inherit from any other memory pools . */
    aos_pool_c reate (& pool , NULL );
    /* Create and initialize options . This parameter
    mainly includes global configurat ion informatio n , such
    as endpoint , access_key _id , acces_key_ secret , is_cname
    , and curl . */
    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );

```

```

/* Initialize oss_client_options . */
init_options ( oss_client_options );
/* Initialization parameters . */
aos_string_t bucket ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_str_set (& bucket , bucket_name );
/* Delete all CORS rules . */
resp_status = oss_delete_bucket_cors ( oss_client_options , & bucket , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get bucket cors succeeded \n " );
} else {
    printf ( " get bucket cors failed \n " );
}
/* Release the memory pool , that is , memories allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_io_deinitialize ();
return 0 ;
}

```

7.13 Set logging

You can enable access logs to record bucket access to log files, which are stored in a specified bucket.

The log file format is as follows:

```

< TargetPrefix >< SourceBucket >- YYYY - mm - DD - HH - MM - SS -
UniqueString

```

For more information about access log files, see [Set access logging](#).

Enable access logging

Run the following code to enable bucket access logging:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
const char * target_bucket_name = "< yourTargetBucketName >";
const char * target_logging_prefix = "< yourTargetPrefix >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t . */
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id ,
        access_key_id );
}

```

```

    aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . The value
0 indicates that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options -> ctl = aos_http_controller_create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the maid method at the program portal to
initialize global resources such as network , memory ,
and so on . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , equivalent
to apr_pool_t . The implementation code is included in
the apr library . */
    aos_pool_t * pool ;
    /* Re - create a memory pool . The second parameter
is NULL , indicating that the new pool does not
inherit any other memory pool . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter
includes global configuration information , such as
endpoint , access_key_id , access_key_secret , is_cname , and
curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client_options = oss_request_options_create ( pool );
    /* Initialize oss_client_options . */
    init_options ( oss_client_options );
    /* Initialize parameters . */
    aos_string_t bucket ;
    oss_logging_config_content_t * content ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    aos_str_set (& bucket , bucket_name );
    content = oss_create_logging_rule_content ( pool );
    aos_str_set (& content -> target_bucket , target_bucket_name );
    aos_str_set (& content -> prefix , target_logging_prefix );
    /* Enable bucket access logging . */
    resp_status = oss_put_bucket_logging ( oss_client_options , & bucket , content , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " put symlink succeeded \n " );
    } else {
        printf ( " put symlink failed , code : %d , error_code : %s , error_msg : %s , request_id : %s \n " ,
resp_status -> code , resp_status -> error_code ,
resp_status -> error_msg , resp_status -> req_id );
    }
    /* Release the memory pool , that is , the memories
allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

```
}
```

View access logging configurations

Run the following code to view the access logging configurations for a bucket:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. The value
    0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
    () to initialize global resources, such as networks
    and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, equivalent
    to apr_pool_t. The implementation code is included in
    the apr library. */
    aos_pool_t * pool;
    /* Re-create a memory pool. The second parameter
    is NULL, indicating that the new pool does not
    inherit any other memory pool. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter
    includes global configuration information, such as
    endpoint, access_key_id, access_key_secret, is_cname, and
    curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options
    . */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialization parameters. */
    aos_string_t bucket;
    oss_logging_config_content_t * content;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_status = NULL;
    aos_str_set(&bucket, bucket_name);
    content = oss_create_logging_rule_content(pool);
```

```

/* View bucket access logging configurations. */
resp_status = oss_get_bucket_logging(oss_client
_options, &bucket, content, &resp_headers);
if (aos_status_is_ok(resp_status)) {
    printf("get bucket logging succeeded \n");
} else {
    printf("get bucket logging failed, code:%d,
error_code:%s, error_msg:%s, request_id:%s \n",
        resp_status->code, resp_status->error_code,
resp_status->error_msg, resp_status->req_id);
}
printf("bucket:%s, prefix:%s \n", content->
target_bucket.data, content->prefix.data);
/* Release the memory pool, that is, the memories
allocated to resources during the request. */
aos_pool_destroy(pool);
/* Release the allocated global resources. */
aos_http_io_deinitialize();
return 0;
}

```

Disable access logging

Run the following code to disable access logging for a bucket:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. The value
    0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
    () to initialize global resources, such as networks
    and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, equivalent
    to apr_pool_t. The implementation code is included in
    the apr library. */
    aos_pool_t * pool;

```

```

/* Re - create a memory pool . The second parameter
is NULL , indicating that the new pool does not
inherit any other memory pool . */
aos_pool_create (& pool , NULL );
/* Create and initialize options . This parameter
includes global configuration information , such as
endpoint , access_key_id , access_key_secret , is_cname , and
curl . */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to options
. */
oss_client_options = oss_request_options_create ( pool );
/* Initialize oss_client_options . */
init_options ( oss_client_options );
/* Initialization parameters . */
aos_string_t bucket ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
aos_str_set (& bucket , bucket_name );
/* Disable access logging . */
resp_status = oss_delete_bucket_logging ( oss_client_options , & bucket , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " delete bucket logging succeeded \n " );
} else {
    printf ( " delete bucket logging failed , code : %d ,
error_code : %s , error_msg : %s , request_id : %s \n " ,
resp_status -> code , resp_status -> error_code ,
resp_status -> error_msg , resp_status -> req_id );
}
/* Release the memory pool , that is , the memories
allocated to resources during the request . */
aos_pool_destroy ( pool );
/* Release the allocated global resources . */
aos_http_info_deinitialize ();
return 0 ;
}

```

7.14 Anti-leech

This topic describes how to use anti-leech.

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leeching, see [Anti-leeching settings](#).

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options(oss_request_options_t * options)
{
    options->config = oss_config_create(options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set(&options->config->endpoint, endpoint);
    aos_str_set(&options->config->access_key_id,
access_key_id);
    aos_str_set(&options->config->access_key_secret,
access_key_secret);
    /* Determine whether the CNAME is used. 0 indicates
that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create(options->
pool, 0);
}
int main(int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
() to initialize global resources, such as networks
and memories. */
    if (aos_http_io_initialize(NULL, 0) != AOSE_OK) {
        exit(1);
    }
    /* Memory pool used to manage memories, which is
equivalent to apr_pool_t. The implementation code is
included in the apr library. */
    aos_pool_t * pool;
    /* Re-create a new memory pool. The second
parameter is NULL, indicating that it does not
inherit from any other memory pools. */
    aos_pool_create(&pool, NULL);
    /* Create and initialize options. This parameter
mainly includes global configuration information, such
as endpoint, access_key_id, access_key_secret, is_cname
, and curl. */
    oss_request_options_t * oss_client_options;
    /* Allocate memories in the memory pool to options
. */
    oss_client_options = oss_request_options_create(pool);
    /* Initialize oss_client_options. */
    init_options(oss_client_options);
    /* Initialization parameters. */
    aos_string_t bucket;
    aos_table_t * resp_headers = NULL;
    aos_status_t * resp_status = NULL;
    oss_referer_config_t referer_config;
    aos_str_set(&bucket, bucket_name);
    aos_list_init(&referer_config.referer_list);
    oss_create_and_add_refer(pool, &referer_config, "http
://www.aliyun.com");
```



```

    oss_create_and_add_referer ( pool , & referer_config , "
https://www.aliyun.com");
    referer_config.allow_empty_referer = 0;
    /* Add the referer field. The referer field allows
question marks (?) and asterisks (*) for wildcard use.
*/
    resp_status = oss_put_bucket_referer ( oss_client
_options , & bucket , & referer_config , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " put bucket referer succeeded \n ");
    } else {
        printf ( " put bucket referer failed \n ");
    }
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_io_deinitialize ();
    return 0 ;
}

```

Obtain a referer whitelist

Run the following code to obtain a referer whitelist:

```

#include "oss_api.h"
#include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options->config = oss_config_create ( options->pool );
    /* Use a char * string to initialize aos_string_t .
*/
    aos_str_set (& options->config->endpoint , endpoint );
    aos_str_set (& options->config->access_key_id ,
access_key_id );
    aos_str_set (& options->config->access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options->config->is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options->ctl = aos_http_controller_create ( options->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_io_initialize method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_io_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementation code is
included in the apr library . */
    aos_pool_t * pool ;

```

```

/* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
aos_pool_c reate (& pool , NULL );
/* Create and initialize options . This parameter
mainly includes global configurat ion informatio n , such
as endpoint , access_key _id , acces_key_ secret , is_cname
, and curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate memories in the memory pool to options
. */
oss_client _options = oss_reques t_options_ create ( pool );
/* Initialize oss_client _options . */
init_optio ns ( oss_client _options );
/* Initializa tion parameters . */
aos_string _t bucket ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
oss_refere r_config_t referer_co nfig ;
oss_refere r_t * referer ;
aos_str_se t (& bucket , bucket_nam e );
aos_list_i nit (& referer_co nfig . referer_li st );
/* Obtain the referer list for a bucket . */
resp_statu s = oss_get_bu cket_refer er ( oss_client
_options , & bucket , & referer_co nfig , & resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
    printf ( " get bucket referer succeeded \ n " );
    aos_list_f or_each_en try ( oss_refere r_t , referer , &
referer_co nfig . referer_li st , node ) {
        printf ( " get referer % s \ n " , referer -> referer .
data );
    }
} else {
    printf ( " get bucket referer failed \ n " );
}
/* Release the memory pool , that is , memories
allocated to resources during the request . */
aos_pool_d estroy ( pool );
/* Release the allocated global resources . */
aos_http_i o_deinitia lize ();
return 0 ;
}

```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
}

```

```

    aos_str_set (& options -> config -> access_key_secret ,
access_key_secret );
    /* Determine whether the CNAME is used . 0 indicates
that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options -> ctl = aos_http_controller_create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
    /* Call the aos_http_initialize method in main
() to initialize global resources , such as networks
and memories . */
    if ( aos_http_initialize ( NULL , 0 ) != AOSE_OK ) {
        exit ( 1 );
    }
    /* Memory pool used to manage memories , which is
equivalent to apr_pool_t . The implementation code is
included in the apr library . */
    aos_pool_t * pool ;
    /* Re - create a new memory pool . The second
parameter is NULL , indicating that it does not
inherit from any other memory pools . */
    aos_pool_create (& pool , NULL );
    /* Create and initialize options . This parameter
mainly includes global configuration information , such
as endpoint , access_key_id , access_key_secret , is_cname
, and curl . */
    oss_request_options_t * oss_client_options ;
    /* Allocate memories in the memory pool to options
. */
    oss_client_options = oss_request_options_create ( pool );
    /* Initialize oss_client_options . */
    init_options ( oss_client_options );
    /* Initialization parameters . */
    aos_string_t bucket ;
    aos_table_t * resp_headers = NULL ;
    aos_status_t * resp_status = NULL ;
    oss_referer_config_t referer_config ;
    aos_str_set (& bucket , bucket_name );
    aos_list_init (& referer_config . referer_list );
    referer_config . allow_empty_referer = 1 ;
    /* You cannot clear a referer whitelist directly .
To clear a referer whitelist , you need to create
the rule that allows an empty referer field and
replace the original rule with the new rule . */
    resp_status = oss_put_bucket_referer ( oss_client
_options , & bucket , & referer_config , & resp_headers );
    if ( aos_status_is_ok ( resp_status )) {
        printf ( " delete bucket referer succeeded \n " );
    } else {
        printf ( " delete bucket referer failed \n " );
    }
    /* Release the memory pool , that is , memories
allocated to resources during the request . */
    aos_pool_destroy ( pool );
    /* Release the allocated global resources . */
    aos_http_initialize ();
    return 0 ;
}

```

```
}
```

7.15 Static website hosting

You can set your bucket to static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

For more information about static website hosting, see [Configure static website hosting](#).

Configure static website hosting

Run the following code to configure static website hosting:

```
# include "oss_api.h"
# include "aos_http_io.h"
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options (oss_request_options_t * options)
{
    options->config = oss_config_create (options->pool);
    /* Use a char * string to initialize aos_string_t.
    */
    aos_str_set (& options->config->endpoint, endpoint);
    aos_str_set (& options->config->access_key_id,
    access_key_id);
    aos_str_set (& options->config->access_key_secret,
    access_key_secret);
    /* Determine whether the CNAME is used. The value
    0 indicates that the CNAME is not used. */
    options->config->is_cname = 0;
    /* Configure network parameters, such as timeout. */
    options->ctl = aos_http_controller_create (options->
    pool, 0);
}
int main (int argc, char * argv[])
{
    /* Call the aos_http_io_initialize method in main
    () to initialize global resources, such as networks
    and memories. */
    if (aos_http_io_initialize (NULL, 0) != AOSE_OK) {
        exit (1);
    }
    /* Memory pool used to manage memories, equivalent
    to apr_pool_t. The implementation code is included in
    the apr library. */
    aos_pool_t * pool;
    /* Re-create a memory pool. The second parameter
    is NULL, indicating that the new pool does not
    inherit any other memory pool. */
    aos_pool_create (& p, NULL);
    /* Create and initialize options. This parameter
    includes global configuration information, such as
    endpoint, access_key_id, access_key_secret, is_cname, and
    curl. */
```

```

    oss_reques t_options_ t * oss_client _options ;
    /* Allocate memories in the memory pool to options
    . */
    oss_client _options = oss_reques t_options_ create ( pool );
    /* Initialize oss_client _options . */
    init_optio ns ( oss_client _options );
    /* Initialize parameters . */
    aos_string _t bucket ;
    aos_table_ t * resp_heade rs = NULL ;
    aos_status _t * resp_statu s = NULL ;
    oss_websit e_config_t website_co nfig ;
    aos_str_se t (& bucket , bucket_nam e );
    aos_str_se t (& website_co nfig . suffix_str , " index . html
");
    aos_str_se t (& website_co nfig . key_str , " error . html ");
    /* Configure static web hosting : The index page is
index . html and the error page is error . html . */
    resp_statu s = oss_put_bu cket_websi te ( oss_client
_options , & bucket , & website_co nfig , & resp_heade rs );
    if ( aos_status _is_ok ( resp_statu s )) {
        printf ( " put bucket website succeeded \ n " );
    } else {
        printf ( " put bucket website failed \ n " );
    }
    /* Release the memory pool , that is , the memories
allocated to resources during the request . */
    aos_pool_d estroy ( pool );
    /* Release allocated global resources . */
    aos_http_i o_deinitia lize ();
    return 0 ;
}

```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```

# include " oss_api . h "
# include " aos_http_i o . h "
const char * endpoint = "< yourEndpoi nt >";
const char * access_key _id = "< yourAccess KeyId >";
const char * access_key _secret = "< yourAccess KeySecret >";
const char * bucket_nam e = "< yourBucket Name >";
void init_optio ns ( oss_reques t_options_ t * options )
{
    options -> config = oss_config _create ( options -> pool );
    /* Use a char * string to initialize aos_string _t .
    */
    aos_str_se t (& options -> config -> endpoint , endpoint );
    aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
    aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
    /* Determine whether the CNAME is used . The value
0 indicates that the CNAME is not used . */
    options -> config -> is_cname = 0 ;
    /* Configure network parameters , such as timeout . */
    options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{

```

```

/* Call the aos_http_io_initialize method in main
() to initialize global resources, such as networks
and memories. */
if (aos_http_io_initialize ( NULL , 0 ) != AOSE_OK ) {
    exit ( 1 );
}
/* Memory pool used to manage memories, equivalent
to apr_pool_t. The implementation code is included in
the apr library. */
aos_pool_t * pool ;
/* Re - create a memory pool. The second parameter
is NULL, indicating that the new pool does not
inherit any other memory pool. */
aos_pool_create (& pool , NULL );
/* Create and initialize options. This parameter
includes global configuration information, such as
endpoint, access_key_id, access_key_secret, is_cname, and
curl. */
oss_request_options_t * oss_client_options ;
/* Allocate memories in the memory pool to options
. */
oss_client_options = oss_request_options_create ( pool );
/* Initialize oss_client_options. */
init_options ( oss_client_options );
/* Initialization parameters. */
aos_string_t bucket ;
aos_table_t * resp_headers = NULL ;
aos_status_t * resp_status = NULL ;
oss_website_config_t website_config ;
aos_str_set (& bucket , bucket_name );
/* View static web hosting configurations. */
resp_status = oss_get_bucket_website ( oss_client_options , & bucket , & website_config , & resp_headers );
if ( aos_status_is_ok ( resp_status )) {
    printf ( " get bucket website succeeded \n " );
    printf ( " website_config : %s %s \n " , website_config . suffix_str . data , website_config . key_str . data );
} else {
    printf ( " get bucket website failed \n " );
}
/* Release the memory pool, that is, the memories
allocated to resources during the request. */
aos_pool_destroy ( pool );
/* Release the allocated global resources. */
aos_http_io_deinitialize ();
return 0 ;
}

```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```

# include " oss_api . h "
# include " aos_http_io . h "
const char * endpoint = "< yourEndpoint >";
const char * access_key_id = "< yourAccessKeyId >";
const char * access_key_secret = "< yourAccessKeySecret >";
const char * bucket_name = "< yourBucketName >";
void init_options ( oss_request_options_t * options )
{
    options -> config = oss_config_create ( options -> pool );
}

```

```

/* Use a char * string to initialize aos_string_t .
*/
aos_str_se t (& options -> config -> endpoint , endpoint );
aos_str_se t (& options -> config -> access_key _id ,
access_key _id );
aos_str_se t (& options -> config -> access_key _secret ,
access_key _secret );
/* Determine whether the CNAME is used . The value
0 indicates that the CNAME is not used . */
options -> config -> is_cname = 0 ;
/* Configure network parameters , such as timeout . */
options -> ctl = aos_http_c ontroller_ create ( options ->
pool , 0 );
}
int main ( int argc , char * argv [])
{
/* Call the aos_http_i o_initiali ze method in main
() to initialize global resources , such as networks
and memories . */
if ( aos_http_i o_initiali ze ( NULL , 0 ) != AOSE_OK ) {
exit ( 1 );
}
/* Memory pool used to manage memories , equivalent
to apr_pool_t . The implementa tion code is included in
the apr library . */
aos_pool_t * pool ;
/* Re - create a memory pool . The second parameter
is NULL , indicating that the new pool does not
inherit any other memory pool . */
aos_pool_c reate (& pool , NULL );
/* Create and initialize options . This parameter
includes global configurat ion informatio n , such as
endpoint , access_key _id , acces_key_ secret , is_cname , and
curl . */
oss_reques t_options_ t * oss_client _options ;
/* Allocate the memories in the memory pool to
options . */
oss_client _options = oss_reques t_options_ create ( pool );
/* Initialize oss_client _options . */
init_optio ns ( oss_client _options );
/* Initialize parameters . */
aos_string _t bucket ;
aos_table_ t * resp_heade rs = NULL ;
aos_status _t * resp_statu s = NULL ;
aos_str_se t (& bucket , bucket_nam e );
/* Delete static website hosting configurat ions . */
resp_statu s = oss_delete _bucket_we bsite ( oss_client
_options , & bucket , & resp_heade rs );
if ( aos_status _is_ok ( resp_statu s )) {
printf ( " delete bucket website succeeded \ n " );
} else {
printf ( " delete bucket website failed \ n " );
}
/* Release the memory pool , that is , the memories
allocated to resources during the request . */
aos_pool_d estroy ( pool );
/* Release the allocated global resources . */
aos_http_i o_deinitia lize ();
return 0 ;

```

```
}
```

7.16 Error handling

Exception handling

If a request error occurs in the use of OSS C SDK, the corresponding error message is output in `aos_status_s`. The attributes of `aos_status_s` include:

Error code	Description	Type
<code>code</code>	The HTTP status code of the error request	Integer
<code>error_code</code>	Error code returned by OSS	Character string
<code>error_msg</code>	Error message returned by OSS	Character string
<code>req_id</code>	The UUID that identifies the request. You can provide the <code>req_id</code> to OSS development engineers for further help.	Character string

Timeout handling

If the code in the returned `aos_status_t` is not “2XX”, and the `error_code` is -992 or -995, it indicates a connection timeout error or a request timeout error occurs. You can try again.

You can use the `aos_should_retry(aos_status_t *)` in the `aos_status.h` to determine whether the returned error code requires a retry. If the returned value is 1, you must try again.

Common error codes

Error code	Description	HTTP status code
<code>AccessDenied</code>	Access denied	403
<code>BucketAlreadyExists</code>	The bucket already exists.	409
<code>BucketNotEmpty</code>	The bucket is not empty.	409
<code>EntityTooLarge</code>	The entity size exceeds the maximum limit.	400

Error code	Description	HTTP status code
EntityTooSmall	The entity size is below the minimum limit.	400
FileGroupTooLarge	The total file group size exceeds the maximum limit.	400
FilePartNotExist	The file part does not exist.	400
FilePartStale	The file part has expired.	400
InvalidArgument	The parameter format is invalid.	400
InvalidAccessKeyId	The AccessKeyId does not exist.	403
InvalidBucketName	The bucket name is invalid.	400
InvalidDigest	The digest is invalid.	400
InvalidObjectName	The object name is invalid.	400
InvalidPart	The part is invalid.	400
InvalidPartOrder	The part order is invalid.	400
InvalidTargetBucketForLogging	The buckets that store log files are invalid.	400
InternalError	Internal OSS error	500
MalformedXML	The XML format is invalid.	400
MethodNotAllowed	The method is not supported.	405
MissingArgument	Parameters are not configured.	411
MissingContentLength	The content length is not configured.	411
NoSuchBucket	The bucket does not exist.	404
NoSuchKey	The object does not exist	404
NoSuchUpload	The multipart upload ID does not exist.	404
NotImplemented	The method cannot be processed.	501

Error code	Description	HTTP status code
PreconditionFailed	Pre-processing error.	412
RequestTimeTooSkewed	The local time set for OSSClient is deviated from the time set for the OSS server by over 15 minutes.	403
RequestTimeout	Request timeout	400
SignatureDoesNotMatch	Signature mismatch	403
TooManyBuckets	The number of buckets exceed the limits,.	400

7.17 FAQ

This topic lists the FAQs about OSS C SDK.

- [How to configure the output of application logs in OSS C SDK.](#)
- [How to configure parameters for cURL-based communication in OSS C SDK.](#)
- [How to upload data using callback of cURL in OSS C SDK.](#)
- [How to upload data using callback of cURL in OSS C SDK.](#)
- [How to download data using callback of cURL in OSS C SDK.](#)
- [How to upload and download objects with names containing Chinese characters in OSS C SDK \(for Windows\).](#)

8 . NET

8.1 Preface

The OSS C# SDK supports .NET Framework 2.0 and later. This document is written based on OSS C# SDK 2.8.0.

Compatibility

- For SDK series of 2.x.x
 - API: compatible
 - Namespace: compatible
- For SDK series of 1.0.x
 - API: compatible
 - Namespace: incompatible. Aliyun.OpenServices.OpenStorageService is changed to Aliyun.OSS

SDK source code and API documents

For the SDK source code, see [GitHub](#). For more information, see [API documents](#).

Code samples

OSS SDK for C# provides a variety of code samples for your reference or use. You can obtain the code samples from [GitHub](#). The following table describes the content of code samples.

Sample file	Content
PutObjectSample.cs	Upload objects
AppendObjectSample.cs	Append upload
DoesObjectExistSample.cs	Determine whether an object exists
DeleteObjectsSample.cs	Delete objects
CopyObjectSample.cs	Copy objects
ModifyObjectMetaSample.cs	Manage Object Meta
MultipartUploadSample.cs	Multipart upload
ResumableSample.cs	Resumable upload

Sample file	Content
GetObjectSample.cs	Download objects
GetObjectByRangeSample.cs	Range download
GetObjectAclSample.cs	Manage ACL for an object
SetObjectAclSample.cs	Manage ACL for an object
ListObjectsSample.cs	List objects
UrlSignatureSample.cs	Authorized access
UploadCallbackSample.cs	Upload callback
ProgressSample.cs	Upload progress bars and Download progress bars
CNameSample.cs	Use custome domain name to access OSS (CNAME)
PostPolicySample.cs	Form upload
CreateBucketSample.cs	Create a bucket
DeleteBucketSample.cs	Delete a bucket
Doesbucketexistssample. CS	Determine whether a bucket exists
ListBucketsSample.cs	List buckets
SetBucketAclSample.cs	Configure an ACL for a bucket
SetBucketLifecycleSample.cs	Lifecycle management
SetBucketLoggingSample.cs	Access log files
SetBucketRefererSample.cs	Anti-leech
SetBucketWetbsiteSample.cs	Static website hosting
SetBucketCorsSample.cs	CORS
ImageProcessSample.cs	Image processing

8.2 Installation

This topic describes how to install OSS C# SDK.

Preparation

- Windows
 - Applicable to .NET 2 . 0 and later.
 - Applicable to Visual Studio 2010 and later.

- **Linux Mac**
 - Applicable to `Mono 3 . 12` and later.

Download SDK

- [Direct download](#)
- [Download from GitHub](#)
- [Download previous versions](#)

Install SDK

- **Install OSS C# SDK in Windows**
 - Install SDK through NuGet
 1. If your Visual Studio does not have NuGet installed, install [NuGet](#) first.
 2. Create a project or open an existing project in Visual Studio, select **Tools > NuGet Package Manager > Management Solution NuGet Package**.
 3. Search for `aliyun.oss.sdk` and find `Aliyun.OSS.SDK` in the search results. Select the latest version, and click **Install**.
 - Install SDK through DLL reference
 1. Download and unzip the C# SDK package.
 2. In the Visual Studio, access Solution Resource Manager, select your project, right click Project Name. Select **Reference > Add Reference**, and then select **Browse** in the prompt dialog box.
 3. Find the directory that the SDK package is unzipped to, select `Aliyun . OSS . dll` in the bin directory, and click **OK**.
 - Install SDK through project introduction

If you download SDK package or the source code from GitHub and you want to install SDK package using the source code, you can

1. right click Solution in Visual Studio and select **Add** in the dialog box menu to add an existing project.
2. In the prompt dialog box, select `aliyun - oss - sdk . csproj` , and click **Open**.
3. Right click Your Projects and select **Reference > Add Reference**. In the prompt dialog box, click the **Project** tab, select the `aliyun-oss-sdk` project, and then click **OK**.

- Install OSS C# SDK in Windows
 - Install SDK through NuGet
 1. In Xamarin, create a project or open an existing project, and select Add NuGet Packages from Tool.
 2. Search and find aliyun.oss.sdk, select the latest version, and click Add Package to add the package to project application.
 - Install SDK through DLL reference
 1. Download and unzip the C# SDK package.
 2. In Xamarin, select your project in Solution, right click Reference, and then select Edit References in the prompt menu.
 3. In the Edit References dialog box, select `. Net Assembly` , click Browse. Find the directory that SDK is unzipped to, select `Aliyun . OSS . dll` in the bin directory, and then click Open.

8.3 Quick start

This topic describes how to use OSS .NET SDK to perform routine operations such as bucket creation, object uploads, and object downloads.

Create a bucket

A bucket is a global namespace in OSS. It is similar to a data container that stores files .

Run the following code to create a bucket:

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Create a bucket .
    var bucket = client . CreateBucket ( bucketName );
    Console . WriteLine ( " Create bucket succeeded , { 0 } " ,
bucket . Name );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Create bucket failed , { 0 } " , ex .
Message );
}
```

```
}
```

For more information about bucket naming rules, see naming conventions in [Basic concepts](#).

For more information about endpoints, see [Regions and endpoints](#).

Upload objects

Run the following code to upload a file to OSS:

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Upload a file
    var result = client . PutObject ( bucketName , objectName ,
localFilename );
    Console . WriteLine ( " Put object succeeded , ETag : { 0 }
", result . ETag );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Put object failed , { 0 }", ex .
Message );
}
```

For more information, see [Upload an object](#).

Download objects

Run the following code to download specified object to a local file:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
var downloadFilename = "< yourDownloadFilename >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    client . PutObject ( bucketName , objectName , localFilename
);
    // Download the object.
    var result = client . GetObject ( bucketName , objectName );
```

```

        using ( var requestStream = result.Content )
        {
            using ( var fs = File.Open( downloadFilename,
                FileMode.OpenOrCreate ))
            {
                int length = 4 * 1024 ;
                var buf = new byte[ length ];
                do
                {
                    length = requestStream.Read( buf, 0,
                        length );
                    fs.Write( buf, 0, length );
                } while ( length != 0 );
            }
            Console.WriteLine( " Get object succeeded " );
        }
    }
    catch ( OssiException ex )
    {
        Console.WriteLine( " Failed with error code : { 0 };
            Error info : { 1 }. \nRequestID :{ 2 }\ tHostID :{ 3 }",
            ex.ErrorCode, ex.Message, ex.RequestId, ex.
            HostId );
    }
    catch ( Exception ex )
    {
        Console.WriteLine( " Failed with error info : { 0 }", ex
            .Message );
    }
}

```

List objects

Run the following code to list objects in a specified bucket:

```

using Aliyun.OSS;
using Aliyun.OSS.Common;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance.
var client = new OssClient( endpoint, accessKeyId,
    accessKeySecret );
try
{
    var objects = new List<string>();
    ObjectListing result = null;
    String nextmarker = string.Empty;
    do
    {
        var listObjectsRequest = new ListObjectsRequest(
            bucketName )
        {
            Marker = nextMarker,
        };
        // List objects.
        result = client.ListObjects( listObjectsRequest );
        foreach ( var summary in result.ObjectSummaries )
        {
            Console.WriteLine( summary.Key );
            objects.Add( summary.Key );
        }
    }
}

```



```

        nextMarker = result.NextMarker;
    } while (result.IsTruncated);
    Console.WriteLine("List objects of bucket:{0} succeeded ", bucketName);
}
catch (OssException ex)
{
    Console.WriteLine("Failed with error code:{0}; Error info:{1}.\nRequestId:{2}\tHostId:{3}", ex.ErrorCode, ex.Message, ex.RequestId, ex.HostId);
}
catch (Exception ex)
{
    Console.WriteLine("Failed with error info:{0}", ex.Message);
}
}

```

Delete objects

Run the following code to delete a specified object:

```

using Aliyun.OSS;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId, accessKeySecret);
try
{
    // Delete an object.
    client.DeleteObject(bucketName, objectName);
    Console.WriteLine("Delete object succeeded");
}
catch (Exception ex)
{
    Console.WriteLine("Delete object failed, {0}", ex.Message);
}
}

```

8.4 Initialization

OSSClient serves as the OSS C# client to manage OSS resources such as buckets and objects.

Create an OSSClient instance

To create an OSSClient instance, you need to specify an endpoint. For more information about endpoints, see [Regions and endpoints](#) and [Bind a custom domain name](#).

- Use an OSS domain to create an OSSClient instance

Run the following code to create an OSSClient instance with a domain assigned by OSS:

```
using Aliyun . OSS ;

const string accessKeyId = "< yourAccess KeyId >";
const string accessKeySecret = "< yourAccess KeySecret >";
const string endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

// Create a new OSSClient instance with the OSS
// access address specified by the user and the
// AccessKeyId / AccessKeySecret granted by Alibaba Cloud
var ossClient = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );
```

- Use a custom domain (CNAME) to create an OSSClient instance

Run the following code to create an OSSClient instance with CNAME:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

const string accessKeyId = "< yourAccess KeyId >";
const string accessKeySecret = "< yourAccess KeySecret >";
const string endpoint = "< yourDomain >";

// Create a ClientConfiguration instance . Modify
// parameters as required .
var conf = new ClientConfiguration ();

// Enable CNAME . CNAME indicates a custom domain
// bound to a bucket .
conf . IsCname = true ;

// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret , conf );
```



Note:

The `ossClient.listBuckets` method cannot be used when a CNAME is used.

Configure an OSSClient instance

`ClientConfiguration` is a configuration class of `OSSClient`. `ClientConfiguration` is used to configure parameters such as user agents, host proxies, connection timeout, and the maximum number of connections. You can configure the following parameters.

Parameter	Description	Configuration method
ConnectionLimit	Specifies the maximum number of HTTP connections that can be enabled.	512
MaxErrorRetry	Specifies the maximum number of retry attempts in the case of a request error.	3
ConnectionTimeout	Specifies the timeout time in milliseconds when connections are established. The default value is -1, indicating that the connections are not time-out.	-1
IsCname	Specifies whether CNAME can be used as an endpoint.	false
ProgressUpdateInterval	Specifies the update interval of the progress bar, which is calculated by bytes.	8096

A code example is given as follows:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var conf = new ClientConfiguration ();
conf . ConnectionLimit = 512 ;
conf . MaxErrorRetry = 3 ;
conf . ConnectionTimeout = 300 ;

var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret , conf );
```

• Data verification

Run the following code for MD5 data verification:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var conf = new ClientConfiguration ();
// Enable MD5 verification . Set EnableMD5Check to
perform MD5 verification on the uploaded or
```

```
downloaded data . MD5 verificati on is disabled by
default .
conf . EnalbeMD5C heck = true ;

var client = new OssClient ( endpoint , accessKeyI d ,
accessKeyS ecret , conf );
```

**Note:**

System performance reduces when MD5 verification is disabled.

- Proxy network

If you use proxy networks, you can configure the following parameters to access OSS:

Parameter	Description	Default value
ProxyHost	A proxy server, such as 8 . 8 . 8 . 8 or abc . def . com .	Null
ProxyPort	A proxy port, such as 3128 or 8080 .	Null
ProxyUserName	The proxy service account , which is optional	Null
ProxyPassword	The proxy service password, which is optional	Null

An example of proxy network access without using the account and password is as follows:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var conf = new ClientConf igation ();
conf . ProxyHost = " 8 . 8 . 8 . 8 ";
conf . ProxyPort = 3128 ;

var client = new OssClient ( endpoint , accessKeyI d ,
accessKeyS ecret , conf );
```

An example of proxy network access using the account and password is as follows:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var conf = new ClientConf igation ();
conf . ProxyHost = " 8 . 8 . 8 . 8 ";
conf . ProxyPort = 3128 ;
conf . ProxyUserN ame = " user ";
```

```
conf.ProxyPassword = "6666";

var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret , conf );
```

8.5 Manage a bucket

A bucket serves as a container that stores objects. Objects belong to a bucket.

Create a bucket

For the complete code of creating a bucket, see [GitHub](#).

You can run the following code to create a bucket:

```
using Aliyun.OSS;

// Initialize an OSSClient.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );

// Create a new bucket.
public void CreateBucket ( string bucketName )
{
    try
    {
        // Create a bucket. The bucketName is globally
        unique.
        client.CreateBucket ( bucketName );
        Console.WriteLine ( "Create bucket succeeded " );
    }
    catch ( Exception ex )
    {
        Console.WriteLine ( "Create bucket failed . { 0 }", ex
        . Message );
    }
}
```

List buckets

For the complete code of listing buckets, see [GitHub](#).

You can run the following code to list all buckets under an account:

```
using Aliyun.OSS;

// Initialize an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );

// List the information about all buckets under an
account.
public void ListBuckets ()
{
    try
    {
        var buckets = client.ListBuckets ();
    }
}
```

```

        Console.WriteLine(" List bucket succeeded ");
        foreach ( var bucket in buckets )
        {
            Console.WriteLine(" Bucket name : { 0 }, Location : {
1 }, Owner : { 2 }", bucket.Name, bucket.Location, bucket.
Owner );
        }
    }
    catch ( Exception ex )
    {
        Console.WriteLine(" List bucket failed . { 0 }", ex.
Message );
    }
}

```

Determine whether a bucket exists

For the complete code of determining whether a bucket exists, see [GitHub](#).

You can run the following code to determine whether a bucket exists:

```

using Aliyun.OSS;

// Initialize an OSSClient instance.
var client = new OssClient( endpoint, accessKeyId,
accessKeySecret );

// Determine whether a bucket exists.
public void DoesBucketExist( string bucketName )
{
    try
    {
        var exist = client.DoesBucketExist( bucketName );

        Console.WriteLine(" Check object Exist succeeded ");
        Console.WriteLine(" exist ? { 0 }", exist );
    }
    catch ( Exception ex )
    {
        Console.WriteLine(" Check object Exist failed . { 0
} ", ex.Message );
    }
}

```

Configure an ACL for a bucket

For the complete code of configuring ACL, see [GitHub](#).

The ACL of a bucket includes the following permissions.

Permission	Description	Value
Private	The bucket owner and the authorized users can read and write objects in the bucket. Other users cannot perform any operation on the objects.	oss.ACLPrivate
Public read	The bucket owner and the authorized users can read and write objects in the bucket. Other users can only read the objects in the bucket. Authorize this permission with caution.	oss.ACLPublicRead
Public read-write	All users can read and write objects in the bucket . Authorize this permission with caution.	oss.ACLPublicReadWrite

You can run the following code to configure an ACL for a bucket:

```
using Aliyun . OSS ;

// Initialize an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );

// Configure an access ACL for the bucket
public void SetBucketACL ( string bucketName )
{
    try
    {
        // Set the ACL of the bucket to public read .
        client . SetBucketACL ( bucketName , CannedAccessControlList . PublicRead );
        Console . WriteLine ( " Set bucket ACL succeeded " );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Set bucket ACL failed . { 0 }",
            ex . Message );
    }
}
```

Obtain the ACL for a bucket

For the complete code of obtaining the ACL for a bucket, see [GitHub](#).

You can run the following code to obtain the ACL for a bucket:

```
using Aliyun . OSS ;
```

```
// Initialize an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );

// Obtain the ACL for the bucket .
public void GetBucketAcl ( string bucketName )
{
    try
    {
        string bucketName = "your - bucket ";
        var acl = client . GetBucketAcl ( bucketName );

        Console . WriteLine ( " Get bucket ACL success ", acl .
ACL . ToString ());
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Get bucket ACL failed . { 0 }",
ex . Message );
    }
}
```

Delete a bucket

For the complete code of deleting a bucket, see [GitHub](#).

Before a bucket is deleted, ensure that all objects in the bucket and fragments that are generated from multipart upload are deleted.



Note:

To delete the fragments that are generated from multipart upload, use [Bucket.ListMultipartUploads](#) to list all fragments, and then use [Bucket.AbortMultipartUpload](#) to delete the fragments.

You can run the following code to delete a bucket:

```
using Aliyun . OSS ;

// Initialize an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );

// Delete the bucket .
public void DeleteBucket ( string bucketName )
{
    try
    {
        client . DeleteBucket ( bucketName );

        Console . WriteLine ( " Delete bucket succeeded ");
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Delete bucket failed . { 0 }", ex
. Message );
    }
}
```



```
}
```

8.6 Upload objects

8.6.1 Overview

In OSS, an object is the basic unit for data operations. OSS .NET SDK provides the following file upload methods:

- **Simple upload**: supports the upload of a file up to 5 GB in size.
- **Append upload**: supports the upload of a file up to 5 GB in size.
- **Resumable upload**: supports concurrent upload and you can define the size of each part. It applies to the upload of large files. Resumable upload supports the upload of a file up to 48.8 TB in size.
- **Multipart upload**: supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files.



Note:

For more information about the usage scenarios of each upload method, see “Upload files” in OSS Developer Guide.

During the file upload, you can set **Object Meta**, and view the **upload progress**. After you complete the file upload, you can perform **upload callback**.

8.6.2 Simple upload

Upload a string

For the complete code of uploading a string, see [GitHub](#).

You can run the following code to upload a string:

```
using System . Text ;
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var objectContent = "More than just cloud ." ;
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );
try
{
    byte [] binaryData = Encoding . ASCII . GetBytes ( objectContent );
}
```

```

        MemoryStream requestContent = new MemoryStream (
            binaryData );
        // Upload a file .
        client . PutObject ( bucketName , objectName , requestContent
    );
        Console . WriteLine ( " Put object succeeded " );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Put object failed , { 0 } " , ex .
            Message );
    }
}

```

Upload a specified local file

For the complete code of uploading a specified local file, see [Git Hub](#).

You can run the following code to upload a specified local file:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );
try
{
    // Upload a local file .
    client . PutObject ( bucketName , objectName , localFilename
    );
    Console . WriteLine ( " Put object succeeded " );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Put object failed , { 0 } " , ex .
        Message );
}
}

```

Upload a file with MD5 verification

To ensure the consistency between the data sent by the SDK and the data received by the OSS client, you can add a Content-MD5 value in ObjectMeta. Then OSS checks the received data against the MD5 value.

For the complete code of uploading a file with MD5 verification, see [Git Hub](#).

```

using Aliyun . OSS ;
using Aliyun . OSS . Util ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
// Create an OSSClient instance .

```

```

var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Calculate the MD5 value .
    string md5 ;
    using ( var fs = File . Open ( localFilename , FileMode .
Open ))
    {
        md5 = OssUtils . ComputeContentMd5 ( fs , fs . Length );
    }
    var objectMeta = new ObjectMetadata
    {
        ContentMd5 = md5
    };
    // Upload a file .
    client . PutObject ( bucketName , objectName , localFilename
, objectMeta );
    Console . WriteLine ( " Put object succeeded " );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Put object failed , { 0 }" , ex .
Message );
}

```

**Note:**

MD5 verification may cause a reduction in performance.

Asynchronous upload

For the complete code of asynchronous upload, see [GitHub](#).

You can run the following code to perform asynchronous upload:

```

using System ;
using System . IO ;
using System . Threading ;

using Aliyun . OSS ;
using Aliyun . OSS . Common ;

// Upload a file asynchronously .
namespace AsyncPutObject
{
    class Program
    {
        static string endpoint = "< yourEndpoint >";
        static string accessKeyId = "< yourAccessKeyId >";
        static string accessKeySecret = "< yourAccessKeySecret >";
        static string bucketName = "< yourBucketName >";
        static string objectName = "< yourObjectName >";
        static string localFilename = "< yourLocalFilename >";

        static AutoResetEvent _event = new AutoResetEvent
( false );

        // Create an OSSClient instance .

```

```

        static OssClient client = new OssClient ( endpoint ,
accessKeyId , accessKeySecret );

        private static void PutObjectCallback ( IAsyncResult
lt ar )
        {
            try
            {
                client . EndPutObject ( ar );
                Console . WriteLine ( ar . AsyncState as string
);
                Console . WriteLine ( " Put object succeeded " );
            }
            catch ( Exception ex )
            {
                Console . WriteLine ( ex . Message );
            }
            finally
            {
                _event . Set ();
            }
        }

        public static void AsyncPutObject ()
        {
            try
            {
                using ( var fs = File . Open ( localFilename
, FileMode . Open ))
                {
                    var metadata = new ObjectMetadata ();
                    // Add custom meta information .
                    metadata . UserMetadata . Add ( " mykey1 " , "
myval1 " );
                    metadata . UserMetadata . Add ( " mykey2 " , "
myval2 " );
                    metadata . CacheControl = " No - Cache ";
                    metadata . ContentType = " text / html ";
                    string result = " Notice user : put
object finish ";

                    // Upload a file asynchronously .
                    client . BeginPutObject ( bucketName ,
objectName , fs , metadata , PutObjectCallback , result .
ToCharArray () );

                    _event . WaitOne ();
                }
            }
            catch ( OssException ex )
            {
                Console . WriteLine ( " Failed with error code :
{ 0 } ; Error info : { 1 } . \n RequestID : { 2 } \ tHostID : { 3 } " ,
ex . ErrorCode , ex . Message , ex . RequestId
, ex . HostId );
            }
            catch ( Exception ex )
            {
                Console . WriteLine ( " Failed with error info :
{ 0 } " , ex . Message );
            }
        }

        static void Main ( string [] args )

```

```

        {
            Program . AsyncPutObject ();
        }
    }
}

```

**Note:**

When using asynchronous upload, you must implement your own callback handler.

8.6.3 Append upload

For the complete code of append upload, see [GitHub](#).

You cannot perform copyObject for appended objects. When you call AppendObject to upload data, if the target object does not exist, an appendable object is created. If the object already exists, data is appended to the object.

Run the following code for append upload:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
// If the object is appended for the first time ,
the append position is 0 , and the returned value is
the position for the next append . The position for
the next append is the length of object before
the append .
long position = 0 ;
try
{
    var metadata = client . GetObjectMetadata ( bucketName ,
objectName );
    position = metadata . ContentLength ;
}
catch ( Exception ) { }
try
{
    using ( var fs = File . Open ( localFilename , FileMode .
Open ))
    {
        var request = new AppendObjectRequest ( bucketName ,
objectName )
        {
            ObjectMetadata = new ObjectMetadata (),
            Content = fs ,
            Position = position
        };
        // Append the object .
        var result = client . AppendObject ( request );
    }
}

```

```

        // Configure the position where the object is
        appended
        position = result . NextAppend Position ;
        Console . WriteLine (" Append object succeeded , next
append    position :{ 0 }", position );
    }
    // Obtain the start position for the next append .
    using ( var fs = File . Open ( localFile name , FileMode .
Open ))
    {
        var request = new AppendObjectRequest ( bucketName ,
objectName )
        {
            ObjectMetadata = new ObjectMetadata (),
            Content = fs ,
            Position = position
        };
        var result = client . AppendObject ( request );
        position = result . NextAppend Position ;
        Console . WriteLine (" Append object succeeded , next
append    position :{ 0 }", position );
    }
}
catch ( Exception ex )
{
    Console . WriteLine (" Append object failed , { 0 }", ex .
Message );
}

```

8.6.4 Resumable upload

You can use resumable upload to split a file you want to upload into several parts and upload them simultaneously. After you have uploaded all parts, you can combine these parts into a complete object. Thus, you can complete uploading the entire object.

Current upload progress is recorded into a checkpoint file during the upload. If a part fails to be uploaded during the process, the object is uploaded from the part recorded in the checkpoint file when the upload restarts, that is, the resumable upload. After the upload is complete, the checkpoint is deleted.

For the complete code of resumable upload, see [GitHub](#).

Run the following code for resumable upload:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
string checkpointDir = "< yourCheckpointDir >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );

```

```

try
{
    // Configure parameters with UploadFile Request .
    UploadObjectRequest request = new UploadObjectRequest (
        bucketName , objectName , localFilename )
    {
        // Specify the size of a part you want to
        upload .
        PartSize = 8 * 1024 * 1024 ,
        // Specify the number of threads for simultaneous
        upload .
        ParallelThreadCount = 3 ,
        // The progress information of a resumable
        upload is stored in the checkpoint directory .
        If a part fails to be uploaded , the progress
        information is used to continue the upload . If the
        checkpoint directory is null , the resumable upload
        does not take effect , and the file that failed to
        be uploaded is uploaded all over again .
        CheckpointDir = checkpointDir ,
    };
    // Start resumable upload .
    client.ResumableUploadObject ( request );
    Console.WriteLine ( " Resumable upload object : { 0 }
        succeeded " , objectName );
}
catch ( OssException ex )
{
    Console.WriteLine ( " Failed with error code : { 0 };
        Error info : { 1 }. \ n RequestID : { 2 } \ t HostID : { 3 } " ,
        ex.ErrorCode , ex.Message , ex.RequestId , ex .
        HostId );
}
catch ( Exception ex )
{
    Console.WriteLine ( " Failed with error info : { 0 } " , ex
        . Message );
}

```

8.6.5 Multipart upload

For the complete code of multipart upload, see [GitHub](#).

To enable multipart upload, perform the following steps:

1. Initiate a multipart upload event.

You can call `OssClient.initiateMultipartUpload` to return the globally unique uploadId created in OSS.

2. Upload parts.

You can call `OssClient.uploadPart` to upload part data.



Notice:

- For parts with a same uploadId, parts are sequenced by their part numbers. If you have uploaded a part and use the same part number to upload another part, the later part will replace the former part.
- OSS places the MD5 value of part data in ETag and returns the MD5 value to the user.
- SDK automatically configures Content-MD5. OSS calculates the MD5 value of uploaded data and compares it with the MD5 value calculated by SDK. If the two values vary, the error code of InvalidDigest is returned.

3. Complete multipart upload.

After you have uploaded all parts, call `partossClient.completeMultipartUpload` to combine these parts into a complete object.

The following code is used as a complete example that describes the process of multipart upload:

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
// Initiate the multipart upload .
var uploadId = "";
try
{
    // Specify the name and the bucket of the
    // uploaded object . You can configure ObjectMeta when
    // using InitiateMultipartUploadRequest . However , you do
    // not need to specify the ContentLength .
    var request = new InitiateMultipartUploadRequest (
    bucketName , objectName );
    var result = client . InitiateMultipartUpload ( request
    );
    uploadId = result . UploadId ;
    // Print the UploadId
    Console . WriteLine ( " Init multipart upload succeeded
    " );
    Console . WriteLine ( " Upload Id :{ 0 }", result . UploadId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Init multipart upload failed , {
    0 }", ex . Message );
}
// Calculate the total number of parts .
var partSize = 100 * 1024 ;
var fi = new FileInfo ( localFilename );
var fileSize = fi . Length ;
```



```

var partCount = fileSize / partSize ;
if ( fileSize % partSize != 0 )
{
    partCount ++;
}
// Start the multipart upload . partETags is a list of
partETags . OSS verifies the validity of all parts
one by one after it receives the partETags . After
part verification is successful , OSS combines these
parts into a complete object .
var partETags = new List < PartETag >();
try
{
    using ( var fs = File . Open ( localFilename , FileMode .
Open ))
    {
        for ( var i = 0 ; i < partCount ; i ++ )
        {
            var skipBytes = ( long ) partSize * i ;
            // Locate to the start position of the
upload .
            fs . Seek ( skipBytes , 0 );
            // Calculate the size of the uploaded part
. The size of the last part is the size of the
data remained after being split by the part size .
            var size = ( partSize < fileSize - skipBytes ) ?
partSize : ( fileSize - skipBytes );
            var request = new UploadPart Request ( bucketName
, objectName , uploadId )
            {
                InputStream = fs ,
                PartSize = size ,
                PartNumber = i + 1
            };
            // Call the UploadPart interface to upload
the object . The returned results contain the ETag
value of the uploaded part .
            var result = client . UploadPart ( request );
            partETags . Add ( result . PartETag );
            Console . WriteLine ( " finish { 0 }/{ 1 }", partETags .
Count , partCount );
        }
        Console . WriteLine ( " Put multi part upload
succeeded " );
    }
}
catch ( Exception ex )
{
    Console . WriteLine ( " Put multi part upload failed , { 0
}", ex . Message );
}
// List uploaded parts .
try
{
    var listPartsRequest = new ListPartsRequest (
bucketName , objectName , uploadId );
    var listPartsResult = client . ListParts ( listPartsR
quest );
    Console . WriteLine ( " List parts succeeded " );
    // Upload each part simultaneously until all parts
are uploaded .
    var parts = listPartsResult . Parts ;
    foreach ( var part in parts )
    {

```

```

        Console.WriteLine(" partNumber : { 0 }, ETag : { 1 },
        Size : { 2 }", part.PartNumber, part.ETag, part.Size);
    }
}
catch (Exception ex)
{
    Console.WriteLine(" List parts failed , { 0 }", ex.
    Message);
}
// Complete the multipart upload .
try
{
    var completeMultipartUploadRequest = new CompleteMultipartUploadRequest(bucketName, objectName, uploadId);
    foreach (var partETag in partETags)
    {
        completeMultipartUploadRequest.PartETags.Add(partETag);
    }
    var result = client.CompleteMultipartUpload(completeMultipartUploadRequest);
    Console.WriteLine(" complete multi part succeeded ");
}
catch (Exception ex)
{
    Console.WriteLine(" complete multi part failed , { 0 }", ex.Message);
}

```

Cancel a multipart upload event

For the complete code of canceling a multipart upload event, see [GitHub](#).

Run the following code to cancel a multipart upload event:

```

using Aliyun.OSS;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var uploadId = "";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId, accessKeySecret);
// Initiate the multipart upload.
try
{
    var request = new InitiateMultipartUploadRequest(bucketName, objectName);
    var result = client.InitiateMultipartUpload(request);
    uploadId = result.UploadId;
    // Print the UploadId.
    Console.WriteLine(" Init multi part upload succeeded ");
    Console.WriteLine(" Upload Id :{ 0 }", result.UploadId);
}
catch (Exception ex)
{
    Console.WriteLine(" Init multi part upload failed , { 0 }", ex.Message);
}

```

```

}
// Cancel the multipart upload .
try
{
    var request = new AbortMulti partUpload Request (
        bucketName , objectName , uploadId );
    client . AbortMulti partUpload ( request );
    Console . WriteLine ( " Abort multi part succeeded , { 0 }", uploadId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Abort multi part failed , { 0 }", ex . Message );
}

```

List uploaded parts

For the complete code of listing uploaded parts, see [GitHub](#).

Run the following code to list uploaded parts:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var uploadId = "< yourUploadId >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );
try
{
    PartListing listPartsResult = null ;
    var nextMarker = 0 ;
    do
    {
        var listPartsRequest = new ListPartsRequest (
            bucketName , objectName , uploadId )
        {
            PartNumberMarker = nextMarker ,
        };
        // List uploaded parts .
        listPartsResult = client . ListParts ( listPartsRequest );
        Console . WriteLine ( " List parts succeeded " );
        // Upload each part simultaneously until all
        parts are uploaded .
        var parts = listPartsResult . Parts ;
        foreach ( var part in parts )
        {
            Console . WriteLine ( " partNumber : { 0 } , ETag : { 1 } , Size : { 2 }", part . PartNumber , part . ETag , part . Size );
        }
        nextMarker = listPartsResult . NextPartNumberMarker ;
    } while ( listPartsResult . IsTruncated );
}
catch ( Exception ex )
{
}

```

```

        Console.WriteLine(" List parts failed , { 0 }", ex .
        Message );
    }

```

List multipart upload events

Call `ossClient.listMultipartUploads` to list all ongoing part upload events (events that have been initiated but not completed or have been canceled). You can configure the following parameters:

Parameter	Description	Configuration method
prefix	Specifies the prefix that must be included in the returned object name . Note that if you use a prefix for query, the returned object name will contain the prefix.	<code>ListMultipartUploadsRequest.setPrefix(String prefix)</code>
delimiter	Specifies a delimiter of a forward slash (/) used to group object names . The object between the specified prefix and the first occurrence of a delimiter of a forward slash (/) is <code>commonPrefixes</code> .	<code>ListMultipartUploadsRequest.setDelimiter(String delimiter)</code>
maxUploads	Specifies the maximum number of part upload events. The maximum value (also default value) you can set is 1,000.	<code>ListMultipartUploadsRequest.setMaxUploads(Integer maxUploads)</code>
keyMarker	Lists all part upload events with the object whose names start with a letter that comes after the <code>keyMarker</code> value in the alphabetical order. You can use this parameter with the <code>uploadIdMarker</code> parameter to specify the initial position for the specified returned result.	<code>ListMultipartUploadsRequest.setKeyMarker(String keyMarker)</code>

Parameter	Description	Configuration method
uploadIdMarker	You can use this parameter with the keyMarker parameter to specify the initial position for the specified returned result. If you do not configure keyMarker, the uploadIdMarker parameter is invalid. If you configure keyMarker, the query result contains: · All objects whose names start with a letter that comes after the keyMarker value. · All objects whose names start with a letter that is the same as the keyMarker value in the alphabetical order and the value of uploadId greater than that of uploadIdMarker.	ListMultipartUploads Request.setUploadIdMarker(String uploadIdMarker)

For the complete code of listing multipart upload events, see [GitHub](#).

Run the following code to list all multipart upload events:

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    MultipartUploadListing multipartUploadListing = null
;
    var nextMarker = string . Empty ;
    do
    {
        // List multipart upload events .
        var request = new ListMultipartUploads Request (
bucketName )
        {
            KeyMarker = nextMarker ,
        };
    }
```

```

        multipartUploadListings = client.ListMultipartUploads ( request );
        Console.WriteLine ( " List multipart succeeded " );
        // List the information about multipart upload events .
        foreach ( var mu in multipartUploadListings .
MultipartUploads )
        {
            Console.WriteLine ( " Key : { 0 }, UploadId : { 1 }",
mu . Key , mu . UploadId );
        }
        // If the returned result shows that the value
of isTruncated is false , the values of nextKeyMarker
and nextUploadIdMarker are returned and used as
the initial position for the next object reading .
        nextMarker = multipartUploadListings . NextKeyMarker ;
    } while ( multipartUploadListings . IsTruncated );
}
catch ( Exception ex )
{
    Console.WriteLine ( " List multipart uploads failed , {
0 }", ex . Message );
}

```

Specify the prefix and maximum number of returned results

Run the following code to perform multipart upload with a specified prefix and maximum number of returned results:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccess KeyId >";
var accessKeySecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
// Define prefix .
var prefix = "< yourObject Prefix >";
// Set the maximum number of returned result to 100 .
var maxUploads = 100 ;
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    MultipartUploadListings multipartUploadListings = null
;
    var nextMarker = string . Empty ;
    do
    {
        // List multipart upload events . By default , 1 ,
000 parts are listed .
        var request = new ListMultipartUploads Request (
bucketName )
        {
            KeyMarker = nextMarker ,
            // Specify the prefix .
            Prefix = prefix ,
            // Specify the maximum number of returned
result .
            MaxUploads = maxUploads ,
        };
        multipartUploadListings = client . ListMultipartUploads ( request );
    }
}
catch ( Exception ex )
{
    Console.WriteLine ( " List multipart uploads failed , {
0 }", ex . Message );
}

```

```

        Console.WriteLine(" List multi part succeeded ");
        // List the information about each multipart
        upload event .
        foreach ( var mu in multipartUploadList .
        MultipartUploads )
        {
            Console.WriteLine(" Key : { 0 }, UploadId : { 1 }",
            mu . Key , mu . UploadId );
        }
        nextMarker = multipartUploadList . NextKeyMarker ;
    } while ( multipartUploadList . IsTruncated );
}
catch ( Exception ex )
{
    Console.WriteLine(" List multi part uploads failed , {
    0 }", ex . Message );
}

```

8.6.6 Upload progress bars

You can use progress bars to indicate the upload or download progress.

For the complete code of progress bars, see [GitHub](#).

The following code is used as an example to describe how to view progress information with PutObject:

```

using System ;
using System . IO ;
using System . Text ;
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
namespace PutObjectProgress
{
    class Program
    {
        static void Main ( string [] args )
        {
            Program . PutObjectProgress () ;
            Console . ReadKey () ;
        }
        public static void PutObjectProgress ()
        {
            var endpoint = "< yourEndpoint >";
            var accessKeyId = "< yourAccessKeyId >";
            var accessKeySecret = "< yourAccessKeySecret >";
            var bucketName = "< yourBucketName >";
            var objectName = "< yourObjectName >";
            var localFilename = "< yourLocalFilename >";
            // Create an OSSClient instance .
            var client = new OssClient ( endpoint ,
            accessKeyId , accessKeySecret );
            // Upload with a progress bar displayed .
            try
            {
                using ( var fs = File . Open ( localFilename ,
                FileMode . Open ))
                {
                    var putObjectRequest = new PutObjectRequest
                    ( bucketName , objectName , fs );

```

```

        putObjectRequest.StreamTransferProgress
+= streamProgressCallback;
        client.PutObject(putObjectRequest);
    }
    Console.WriteLine("Put object:{0} succeeded",
objectName);
}
catch (OssException ex)
{
    Console.WriteLine("Failed with error code :
{0}; Error info:{1}.\nRequestID:{2}\tHostID:{3}",
ex.ErrorCode, ex.Message, ex.RequestId
, ex.HostId);
}
catch (Exception ex)
{
    Console.WriteLine("Failed with error info :
{0}", ex.Message);
}
}
// Obtain the upload progress.
private static void streamProgressCallback(object
sender, StreamTransferProgressArgs args)
{
    System.Console.WriteLine("ProgressCallback -
Progress:{0}%, TotalBytes:{1}, TransferredBytes:{2}",
args.TransferredBytes * 100 / args.
TotalBytes, args.TotalBytes, args.TransferredBytes);
}
}
}

```

8.6.7 Upload callback

For the complete code of upload callback, see [GitHub](#).

Run the following code for upload callback:

```

using System;
using System.IO;
using System.Text;
using Aliyun.OSS;
using Aliyun.OSS.Common;
using Aliyun.OSS.Util;
namespace Callback
{
    class Program
    {
        static void Main(string[] args)
        {
            Program.PutObjectCallback();
            Console.ReadKey();
        }
        public static void PutObjectCallback()
        {
            var endpoint = "< yourEndpoint >";
            var accessKeyId = "< yourAccessKeyId >";
            var accessKeySecret = "< yourAccessKeySecret >";
            var bucketName = "< yourBucketName >";
            var objectName = "< yourObjectName >";
            var localFilename = "< yourLocalFilename >";

```



```

        // Specify the IP address of the server you
        want to send the callback request to .
        const string callbackUrl = "http://oss-demo.
aliyuncs.com:23450";
        # Configure the value of the body field
        carried in the callback request .
        const string callbackBody = "bucket=${ bucket }&
object=${ object }& etag=${ etag }& size=${ size }& mimeType=${
mimeType }&" +
                                " my_var1=${ x : var1 }&
my_var2=${ x : var2 }";
        // Create an OSSClient instance .
        var client = new OssClient ( endpoint ,
accessKeyId , accessKeySecret );
        try
        {
            string responseContent = "";
            var metadata = BuildCallbackMetadata (
callbackUrl , callbackBody );
            using ( var fs = File . Open ( localFilename ,
FileMode . Open ))
            {
                var putObjectRequest = new PutObjectR
equest ( bucketName , objectName , fs , metadata );
                var result = client . PutObject ( putObjectR
equest );
                responseContent = GetCallbackResponse (
result );
            }
            Console . WriteLine ( " Put object :{ 0 } succeeded
, callback response content :{ 1 }", objectName , responseCo
ntent );
        }
        catch ( OssException ex )
        {
            Console . WriteLine ( " Failed with error code :
{ 0 }; Error info : { 1 }. \n RequestID : { 2 } \ tHostID : { 3 }",
ex . ErrorCode , ex . Message , ex . RequestId
, ex . HostId );
        }
        catch ( Exception ex )
        {
            Console . WriteLine ( " Failed with error info :
{ 0 }", ex . Message );
        }
        // Configure upload callback .
        private static ObjectMetadata BuildCallbackMetadata
a ( string callbackUrl , string callbackBody )
        {
            string callbackHeaderBuilder = new CallbackHe
aderBuilder ( callbackUrl , callbackBody ). Build ();
            string CallbackVariableHeaderBuilder = new
CallbackVariableHeaderBuilder ().
                AddCallbackVariable ( " x : var1 ", " x : value1
"). AddCallbackVariable ( " x : var2 ", " x : value2 "). Build ();
            var metadata = new ObjectMetadata ();
            metadata . AddHeader ( HttpHeaders . Callback ,
callbackHeaderBuilder );
            metadata . AddHeader ( HttpHeaders . CallbackVa r ,
CallbackVariableHeaderBuilder );
            return metadata ;
        }
    }

```

```

        // Read the message returned from upload callback
        private static string GetCallbackResponse (
            PutObjectResult result)
        {
            string callbackResponse = null;
            using (var stream = result.ResponseStream)
            {
                var buffer = new byte[4 * 1024];
                var bytesRead = stream.Read(buffer, 0,
                    buffer.Length);
                callbackResponse = Encoding.Default.GetString(
                    buffer, 0, bytesRead);
            }
            return callbackResponse;
        }
    }
}

```

**Note:**

For more information about upload callback, see [Upload callback](#) in the Developer Guide.

8.7 Download objects

8.7.1 Overview

OSS .NET SDK provides the following download methods:

- [Streaming download](#)
- [Range download](#)
- [Resumable download](#)

You can view the download progress shown in [download progress bars](#).

8.7.2 Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use streaming download. Streaming download enables you to download part of the object each time until you have downloaded the entire object.

For the complete code of streaming download, see [GitHub](#).

Run the following code to download a specified OSS object to a stream:

```

using Aliyun.OSS;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";

```

```

var bucketName = "< yourBucket Name >";
var objectName = "< yourObject Name >";
var downloadFilename = "< yourDownloadFilename >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Download an object to a stream. OssObject
    includes object information, such as the bucket that
    contains the object, object name, metadata, and an
    input stream.
    var obj = client.GetObject ( bucketName , objectName );
    using ( var requestStream = obj.Content )
    {
        byte [] buf = new byte [ 1024 ];
        var fs = File.Open ( downloadFilename , FileMode .
OpenOrCreate );
        var len = 0 ;
        // Use the input stream to read the object
        content into a file or the memory .
        while ( ( len = requestStream.Read ( buf , 0 , 1024
)) != 0 )
        {
            fs . Write ( buf , 0 , len );
        }
        fs . Close ();
    }
    Console . WriteLine ( " Get object succeeded " );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Get object failed . { 0 }" , ex .
Message );
}

```

8.7.3 Range download

If you only need part of the data in an object, you can use range downloads to download specified content.

For the complete code of range download, see [GitHub](#).

Run the following code for range download:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucket Name >";
var objectName = "< yourObject Name >";
var downloadFilename = "< yourDownloadFilename >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    var getObjectRequest = new GetObjectRequest (
bucketName , objectName );
    // Set the range, which is from the 20th byte
    to the 100th byte .

```

```

        getObjectRequest.SetRange(20, 100);
        // Start range download. The setRange of getObjectRequest
        // can be used to implement multipart download
        // and resumable download.
        var obj = client.GetObject(getObjectRequest);
        // Download the data and write it into a file.
        using (var requestStream = obj.Content)
        {
            byte[] buf = new byte[1024];
            var fs = File.Open(downloadFilename, FileMode.OpenOrCreate);
            var len = 0;
            while ((len = requestStream.Read(buf, 0, 1024)) != 0)
            {
                fs.Write(buf, 0, len);
            }
            fs.Close();
        }
        Console.WriteLine("Get object succeeded");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Get object failed. {0}", ex.Message);
    }
}

```

You can configure the following parameters of `GetObjectRequest`.

Parameter	Description
Range	Specifies the transmission range of an object.
ModifiedSinceConstraint	If the specified time is earlier than the actual modification time of the file, the file is transmitted normally. Otherwise, a 304 Not Modified exception is thrown.
UnmodifiedSinceConstraint	If the time passed to the parameter is equal to or later than the actual modification time of the file, the file is transmitted normally. Otherwise, a 412 precondition failed exception is thrown.
MatchingETagConstraints	Passes in a group of ETags. If the ETags match the ETag of the file, the file is transmitted normally. Otherwise, a 412 precondition failed exception is thrown.
NonmatchingETagConstraints	Passes in a group of ETags. If the ETags do not match the ETag of the file, the file is transmitted normally. Otherwise, a 304 Not Modified exception is thrown.

Parameter	Description
ResponseHeaderOverrides	Customizes some headers in requests returned by OSS.

8.7.4 Resumable download

Large-sized objects may fail to be downloaded because the network is unstable or the Java program exits abnormally and the entire object needs to be downloaded again. However, the object may still fail to be downloaded after multiple attempts. Therefore, OSS provides resumable download.

Run the following code for resumable download:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
var downloadFilename = "< yourDownloadFilename >";
var checkpointDir = "< yourCheckpointDir >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Configure parameters with DownloadObjectRequest.
    DownloadObjectRequest request = new DownloadObject
    Request ( bucketName , objectName , downloadFilename )
    {
        // Specify the size of a part you want to
        download .
        PartSize = 8 * 1024 * 1024 ,
        // Specifies the number of concurrent download
        threads .
        ParallelThreadCount = 3 ,
        // Specify the checkpointDir to store the
        download progress information . If the download fails ,
        it can be continued based on the progress informatio
        n . If the checkpointDir directory is null , the
        resumable upload is not enabled , which means that
        objects that fails to be downloaded are downloaded
        all over again .
        CheckpointDir = checkpointDir ,
    };
    // Start resumable download .
    client . ResumableDownloadObject ( request );
    Console . WriteLine ( " Resumable download object : { 0 }
    succeeded " , objectName );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error code : { 0 } ;
    Error info : { 1 } . \n RequestID : { 2 } \n tHostID : { 3 } " ,
```

```

        ex . ErrorCode , ex . Message , ex . RequestId , ex .
        HostId );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Failed with error info : { 0 } " , ex
        . Message );
    }
}

```

For more information about resumable download, see [Resumable download](#) in OSS Developer Guide.

8.7.5 Download progress bars

You can use progress bars to indicate the upload or download progress.

For the complete code of download progress bar, see [GitHub](#).

The following code is used as an example to describe how to view progress information with GetObject:

```

using System ;
using System . IO ;
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
namespace GetObjectP rogress
{
    class Program
    {
        static void Main ( string [] args )
        {
            Program . GetObjectP rogress () ;
            Console . ReadKey () ;
        }
        public static void GetObjectP rogress ()
        {
            var endpoint = "< yourEndpoi nt >";
            var accessKeyI d = "< yourAccess KeyId >";
            var accessKeyS ecret = "< yourAccess KeySecret >";
            var bucketName = "< yourBucket Name >";
            var objectName = "< yourObject Name >";
            // Create an OSSClient instance .
            var client = new OssClient ( endpoint ,
accessKeyI d , accessKeyS ecret );
            try
            {
                var getObjectR equest = new GetObjectR equest
( bucketName , objectName );
                getObjectR equest . StreamTran sferProgre ss +=
streamProg ressCallba ck ;
                // Download an object .
                var ossObject = client . GetObject ( getObjectR
equest );
                using ( var stream = ossObject . Content )
                {
                    var buffer = new byte [ 1024 * 1024 ];
                    var bytesRead = 0 ;
                    while (( bytesRead = stream . Read ( buffer ,
0 , buffer . Length )) > 0 )
                    {

```

```

// Process the read data ( detailed
code is not listed here ).
    }
    Console.WriteLine(" Get object :{ 0 } succeeded
", objectName );
    }
    catch ( OssExcepti on ex )
    {
        Console.WriteLine(" Failed with error code :
{ 0 }; Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
ex.ErrorCode , ex.Message , ex.RequestId
, ex.HostId );
    }
    catch ( Exception ex )
    {
        Console.WriteLine(" Failed with error info :
{ 0 }", ex.Message );
    }
}
private static void streamProg ressCallba ck ( object
sender , StreamTran sferProgre ssArgs args )
{
    System.Console.WriteLine(" ProgressCa llback -
Progress : { 0 }%, TotalBytes :{ 1 }, Transferre dBytes :{ 2 } ",
args.Transferre dBytes * 100 / args.
TotalBytes , args.TotalBytes , args.Transferre dBytes );
}
}
}

```

8.8 Manage objects

8.8.1 Overview

This topic is the overview of object management in OSS .NET SDK.

You can perform the following operations on objects in a bucket with APIs:

- [Determine whether an object exists](#)
- [Manage ACL for an object](#)
- [Manage Object Meta](#)
- [List objects](#)
- [Delete an object](#)
- [Copy an object](#)
- [Restore an archive object](#)
- [Manage a symbolic link](#)

8.8.2 Determine whether a specified object exists

Run the following code to determine whether a specified object exists:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";

// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Determine whether a specified object exists .
    var exist = client . DoesObjectExist ( bucketName ,
objectName );
    Console . WriteLine ( " Object exist ? " + exist );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error code : { 0 } ;
Error info : { 1 } . \ nRequestID : { 2 } \ tHostID : { 3 } " ,
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " , ex
. Message );
}
```

8.8.3 Manage ACL for an object

The following table describes the permissions included in the Access Control List (ACL) for an object.

Permission	Description	Value
default	The ACL of an object is the same with that of its bucket.	CannedAccessControlList.Default
Private	Only the object owner and authorized users can read and write the object.	CannedAccessControlList.Private

Permission	Description	Value
Public read	Only the object owner and authorized users can read and write the object. Other users can only read the object. Authorize this permission with caution.	CannedAccessControlList.PublicRead
Public read-write	All users can read and write the object. Authorize this permission with caution.	CannedAccessControlList.PublicReadWrite

The ACL of objects take precedence over that of buckets. For example, if the ACL of a bucket is private, while the object ACL is public read-write, all users can read and write the object. If an object is not configured with an ACL, its ACL is the same as that of its bucket by default.

For the complete code of configuring an ACL for an object, see [GitHub](#). For the complete code of obtaining the ACL for an object, see [GitHub](#).

You can run the following code to configure an ACL for an object and obtain the ACL for an object:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;

var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";

// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
// Configure an ACL for an object .
try
{
    // Use SetObjectACL to configure an ACL for an
    object .
    client . SetObjectACL ( bucketName , objectName , CannedAccessControlList.PublicRead );
    Console . WriteLine ( " Set Object : { 0 } ACL succeeded ",
objectName );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Set Object ACL failed with error
info : { 0 } ", ex . Message );
}
// Obtain the ACL for an object .
```

```

try
{
    // Use GetObjectACL to obtain the ACL for an
    object .
    var result = client . GetObjectACL ( bucketName ,
    objectName );
    Console . WriteLine ( " Get Object ACL succeeded , Id : { 0 }
    ACL : { 1 } ",
        result . Owner . Id , result . ACL . ToString ());
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error code : { 0 };
    Error info : { 1 }. \ nRequestID : { 2 } \ tHostID : { 3 } ",
        ex . ErrorCode , ex . Message , ex . RequestId , ex .
    HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } ", ex
    . Message );
}

```

For details about object ACL, see [Access control](#).

8.8.4 Manage Object Meta

Object Meta includes HTTP headers and user-defined Object Meta. For more information, see [Object Meta](#) in OSS Developer Guide.

For the complete code of configuring Object Meta, see [GitHub](#). For the complete code of modifying Object Meta, see [GitHub](#).

Run the following code to configure, modify, and obtain Object Meta:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var localFilename = "< yourLocalFilename >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    using ( var fs = File . Open ( localFilename , FileMode .
    Open ))
    {
        // Create Object Meta . You can configure HTTP
        headers with Object Meta .
        var metadata = new ObjectMeta data ()
        {
            // Specify the object type .
            ContentType = " text / html ",
            // Configure the expiration time ( GMT is used
        ).
    }
}

```

```

        Expiration Time = DateTime . Parse ( " 2025 - 10 -
12T00 : 00 : 00 . 000Z " ),
    };
    // Configure the length of the object you want
    to upload . If the object length is greater than
    the configured length , only the configured length of
    the object is uploaded . If the configured length is
    greater than the actual object length , the entire
    object is uploaded .
    metadata . ContentLen gth = fs . Length ;
    // Configure the cache action of a webpage when
    an object is downloaded .
    metadata . CacheContr ol = " No - Cache " ;
    // Set the value of metadata mykey1 to myval1 .
    metadata . UserMetada ta . Add ( " mykey1 " , " myval1 " );
    // Set the value of metadata mykey2 to myval2 .
    metadata . UserMetada ta . Add ( " mykey2 " , " myval2 " );
    var saveAsFile name = " Filetest12 3 . txt " ;
    var contentDis position = string . Format ( " attachment
; filename *= utf - 8 '{ 0 }' " , HttpUtils . EncodeUri ( saveAsFile
name , " utf - 8 " ));
    // Provide a default file name when saving the
    requested content as a file .
    metadata . ContentDis position = contentDis position ;
    // Upload a file and configure the Object Meta
    .
    client . PutObject ( bucketName , objectName , fs ,
metadata );
    Console . WriteLine ( " Put object succeeded " );
    // Obtain Object Meta .
    var oldMeta = client . GetObjectM etadata ( bucketName
, objectName );
    // Configure new Object Meta .
    var newMeta = new ObjectMeta data ()
    {
        ContentType = " applicatio n / octet - stream " ,
        Expiration Time = DateTime . Parse ( " 2035 - 11 -
11T00 : 00 : 00 . 000Z " ),
        // Specify the encoding format of a download
        object .
        ContentEnc oding = null ,
        CacheContr ol = ""
    };
    // Add custom Object Meta .
    newMeta . UserMetada ta . Add ( " author " , " oss " );
    newMeta . UserMetada ta . Add ( " flag " , " my - flag " );
    newMeta . UserMetada ta . Add ( " mykey2 " , " myval2 -
modified - value " );
    // Use the ModifyObje ctMeta method to modify
    Object Meta .
    client . ModifyObje ctMeta ( bucketName , objectName ,
newMeta );
    }
}
catch ( Exception ex )
{
    Console . WriteLine ( " Put object failed , { 0 } " , ex .
Message );
}

```



Note:

- For detailed information about HTTP header, see [RFC2616](#).
- The size of Object Meta with custom HTTP headers is no larger than 8 KB.

8.8.5 List objects

For the complete code of listing objects, see [GitHub](#).

Objects are listed alphabetically. You can use `ListObjects` to list objects in a bucket.

The following table describes the parameters of `ListObjects`

Parameter	Description
Delimiter	Specifies a delimiter of a forward slash (/) used to group object names. <code>CommonPrefixes</code> is a set of objects which are ended with the delimiter and have the same prefix.
Marker	Specifies the initial object in the list.
MaxKeys	Specifies the maximum number of objects that can be listed. The default value of <code>MaxKeys</code> is 100. The maximum value of <code>MaxKeys</code> is 1,000.
Prefix	Specifies the prefix you configure to list required objects.

Simple list

For the complete code of simple list, see [GitHub](#).

Run the following code to list objects in a specified bucket.

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    var listObjectRequest = new ListObjectRequest (
bucketName );
    // Simply list the objects in a specified bucket .
    100 records are returned by default .
    var result = client . ListObject ( listObjectRequest );
    Console . WriteLine ( " List objects succeeded " );
    foreach ( var summary in result . ObjectSummaries )
    {
        Console . WriteLine ( " File name :{ 0 }", summary . Key );
    }
}
```

```

}
catch ( Exception ex )
{
    Console . WriteLine ( " List objects failed . { 0 }" , ex .
        Message );
}

```

List a specified number of objects

Use the following code to list a specified number of objects:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );
try
{
    var listObjectsRequest = new ListObjectsRequest (
        bucketName )
    {
        // A maximum of 200 records can be returned .
        MaxKeys = 200 ,
    };
    var result = client . ListObjects ( listObjectsRequest );
    Console . WriteLine ( " List objects succeeded " );
    foreach ( var summary in result . ObjectSummaries )
    {
        Console . WriteLine ( summary . Key );
    }
}
catch ( Exception ex )
{
    Console . WriteLine ( " List objects failed , { 0 }" , ex .
        Message );
}

```

List objects by a specified prefix

Use the following code to list objects with a specified prefix:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var prefix = "< yourObjectPrefix >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
    accessKeySecret );
try
{
    var keys = new List < string > ();
    ObjectListing result = null ;
    string nextMarker = string . Empty ;
    do
    {

```

```

        var listObject sRequest = new ListObject sRequest (
        bucketName )
        {
            Marker = nextMarker ,
            MaxKeys = 100 ,
            Prefix = prefix ,
        };
        result = client . ListObject s ( listObject sRequest );
        foreach ( var summary in result . ObjectSumm aries )
        {
            Console . WriteLine ( summary . Key );
            keys . Add ( summary . Key );
        }
        nextMarker = result . NextMarker ;
    } while ( result . IsTruncate d );
    Console . WriteLine ( " List objects of bucket :{ 0 }
succeeded ", bucketName );
}
catch ( OssExcepti on ex )
{
    Console . WriteLine ( " Failed with error code : { 0 };
Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
        ex . ErrorCode , ex . Message , ex . RequestId , ex .
        HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }", ex
. Message );
}
}

```

List objects by marker

The marker parameter indicates the name of an object from which the listing begins . Run the following code to specify an object (specified with marker) after which the listing begins:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoi nt >";
var accessKeyI d = "< yourAccess KeyId >";
var accessKeyS ecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
var marker = "< yourObject Marker >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyI d ,
accessKeyS ecret );
try
{
    var keys = new List < string >();
    ObjectList ing result = null ;
    string nextMarker = marker ;
    do
    {
        var listObject sRequest = new ListObject sRequest (
        bucketName )
        // To increase the number of returned objects
        , you can modify the MaxKeys parameter or use the
        Marker parameter for multiple reads .
        {
            Marker = nextMarker ,

```

```

        MaxKeys = 100 ,
    };
    result = client . ListObject s ( listObject sRequest );
    foreach ( var summary in result . ObjectSumm aries )
    {
        Console . WriteLine ( summary . Key );
        keys . Add ( summary . Key );
    }
    nextMarker = result . NextMarker ;
    // If the value of IsTruncate d is true , the
    next read starts from NextMarker .
    } while ( result . IsTruncate d );
    Console . WriteLine ( " List objects of bucket :{ 0 }
succeeded " , bucketName );
}
catch ( OssExcepti on ex )
{
    Console . WriteLine ( " Failed with error code : { 0 } ;
Error info : { 1 } . \ nRequestID :{ 2 } \ tHostID :{ 3 } " ,
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " , ex
. Message );
}
}

```

List objects in an asynchronous mode

For the complete code of listing objects in an asynchronous mode, see [GitHub](#).

Run the following code to list objects in an asynchronous mode:

```

using System ;
using System . IO ;
using System . Threading ;
using Aliyun . OSS ;
namespace AsyncListO bjects
{
    class Program
    {
        static string endpoint = "< yourEndpoi nt >";
        static string accessKeyI d = "< yourAccess KeyId >";
        static string accessKeyS ecret = "< yourAccess
KeySecret >";
        static string bucketName = "< yourBucket Name >";
        static AutoResetE vent _event = new AutoResetE vent
( false );
        // Create an OSSClient instance .
        static OssClient client = new OssClient ( endpoint ,
accessKeyI d , accessKeyS ecret );
        static void Main ( string [] args )
        {
            Program . AsyncListO bjects ();
            Console . ReadKey ();
        }
        public static void AsyncListO bjects ()
        {
            try
            {

```

```

        var listObject sRequest = new ListObject
sRequest ( bucketName );
        client . BeginListO bjects ( listObject sRequest ,
ListObject Callback , null );
        _event . WaitOne ();
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Async list objects
failed . { 0 }" , ex . Message );
    }
}
// The ListObject Callback method is implemente d
after an asynchrono us call . Similar interfaces must
be used when an asynchrono us interface is called to
list objects .
private static void ListObject Callback ( IAsyncResult
lt ar )
{
    try
    {
        var result = client . EndListObj ects ( ar );
        foreach ( var summary in result . ObjectSumm
aries )
        {
            Console . WriteLine ( " Object name : 0 " ,
summary . Key );
        }
        _event . Set ();
        Console . WriteLine ( " Async list objects
succeeded " );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Async list objects
failed . { 0 }" , ex . Message );
    }
}
}
}

```

List all objects in a bucket

Run the following code to list all objects in a bucket:

```

using Aliyun . OSS ;
// Initialize OssClient .
var client = new OssClient ( endpoint , accessKeyI d ,
accessKeyS ecret );
// List all objects in the bucket .
public void ListObject ( string bucketName )
{
    try
    {
        ObjectList ing result = null ;
        string nextMarker = string . Empty ;
        do
        {
            // The number of objects listed in each
page is specified by the maxKeys parameter . If the
object number is larger than the specified value ,
other files are listed in another page .

```



```

        var listObject sRequest = new ListObject sRequest
( bucketName )
    {
        Marker = nextMarker ,
        MaxKeys = 100
    };
    result = client . ListObject s ( listObject sRequest
);
    Console . WriteLine ( " File :");
    foreach ( var summary in result . ObjectSumm
aries )
    {
        Console . WriteLine ( " Name :{ 0 }", summary . Key
);
    }
    nextMarker = result . NextMarker ;
    } while ( result . IsTruncate d );
}
catch ( Exception ex )
{
    Console . WriteLine ( " List object failed . { 0 }", ex .
Message );
}
}

```

8.8.6 Delete objects

This topic describes how to delete objects.



Warning:

Delete objects with caution because deleted objects cannot be recovered.

For the complete code of deleting objects, see [GitHub](#).

Delete an object

Run the following code to delete a single object:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoi nt >";
var accessKeyI d = "< yourAccess KeyId >";
var accessKeyS ecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
var objectName = "< yourObject Name >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyI d ,
accessKeyS ecret );
try
{
    /// Delete an object .
    client . DeleteObje ct ( bucketName , objectName );
    Console . WriteLine ( " Delete object succeeded " );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Delete object failed . { 0 }", ex .
Message );
}

```

```
}
```

Delete multiple objects

You can delete a maximum of 1,000 objects simultaneously. Objects can be deleted in two modes: detail (verbose) and simple (quiet) modes.

- **verbose:** returns the list of objects that you have deleted successfully. The default mode is verbose.
- **quiet:** returns the list of objects that you failed to delete.

Run the following code to delete multiple objects simultaneously:

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    var keys = new List < string >();
    var listResult = client . ListObjects ( bucketName );
    foreach ( var summary in listResult . ObjectSummaries )
    {
        keys . Add ( summary . Key );
    }
    // If the quietMode is true , the quiet mode is
    // used . If the quietMode is false , the verbose mode
    // is used . The default mode is verbose .
    var quietMode = false ;
    // The third parameter of the DeleteObjectsRequest
    // specifies the return mode .
    var request = new DeleteObjectsRequest ( bucketName ,
keys , quietMode );
    // Delete multiple objects .
    var result = client . DeleteObjects ( request );
    if ((! quietMode ) && ( result . Keys != null ))
    {
        foreach ( var obj in result . Keys )
        {
            Console . WriteLine ( " Delete successfully : { 0 }
", obj . Key );
        }
    }
    Console . WriteLine ( " Delete objects succeeded " );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Delete objects failed . { 0 }", ex .
Message );
}
```

```
}
```

8.8.7 Copy objects

Copy a small-sized object

For the complete code of copying a small-sized object, see [GitHub](#).

Note the following limits before copying a small-sized object:

- You must have the read and write permission on the source object.
- You cannot copy an object from a region to another region. For example, you cannot copy an object from a bucket in Hangzhou region to a bucket in Qingdao region.
- The object size is limited to a maximum of 1 GB.

Run the following code to copy a small-sized object:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var sourceBucket = "< yourSourceBucketName >";
var sourceObject = "< yourSourceObjectName >";
var targetBucket = "< yourDestBucketName >";
var targetObject = "< yourDestObjectName >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    var metadata = new ObjectMetadata ();
    metadata.AddHeader ( "mk1 ", "mv1 " );
    metadata.AddHeader ( "mk2 ", "mv2 " );
    var req = new CopyObjectRequest ( sourceBucket ,
sourceObject , targetBucket , targetObject )
    {
        // If the value of NewObjectMetadata is null
        , the COPY mode is used ( the metadata of the
source object is copied ). If the value of NewObjectM
etadata is not null , the REPLACE mode is used ( the
metadata of the source object is replaced ).
        NewObjectMetadata = metadata
    };
    // Copy an object .
    client.CopyObject ( req );
    Console.WriteLine ( " Copy object succeeded " );
}
catch ( OssException ex )
{
    Console.WriteLine ( " Failed with error code : { 0 } ;
Error info : { 1 } . \ nRequestID : { 2 } \ tHostID : { 3 } " ,
ex.ErrorCode , ex.Message , ex.RequestId , ex
.HostId );
}
catch ( Exception ex )
{
}
```

```

        Console.WriteLine("Failed with error info : {0}", ex
        .Message);
    }

```

Copy a large-sized object

For the complete code of copying large-sized objects, see [GitHub](#).

· UploadPartCopy

To copy an object larger than 1 GB, you need to use UploadPartCopy.

1. Use `InitiateMultipartUploadRequest` to initiate an UploadPartCopy event.
2. Use the `UploadPartCopy` method to perform a multipart copy.
3. Use the `CompleteMultipartUpload` method to complete the object copy.

Run the following code for UploadPartCopy task:

```

using Aliyun.OSS;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var sourceBucket = "< yourSourceBucketName >";
var sourceObject = "< yourSourceObjectName >";
var targetBucket = "< yourDestBucketName >";
var targetObject = "< yourDestObjectName >";
var uploadId = "";
var partSize = 50 * 1024 * 1024;
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId,
accessKeySecret);
try
{
    // Initiate a copy task. You can use the
    InitiateMultipartUploadRequest method to specify the
    metadata information about the target object.
    var request = new InitiateMultipartUploadRequest(
targetBucket, targetObject);
    var result = client.InitiateMultipartUpload(request
);
    // Print the uploadId.
    uploadId = result.UploadId;
    Console.WriteLine("Init multipart upload succeeded
, UploadId: {0}", result.UploadId);
    // Calculate the number of parts.
    var metadata = client.GetObjectMetadata(sourceBucket
, sourceObject);
    var fileSize = metadata.ContentLength;
    var partCount = (int) fileSize / partSize;
    if (fileSize % partSize != 0)
    {
        partCount++;
    }
    // Start the UploadPartCopy task.
    var partETags = new List<PartETag>();
    for (var i = 0; i < partCount; i++)
    {
        var skipBytes = (long) partSize * i;

```

```

        var size = ( partSize < fileSize - skipBytes ) ?
partSize : ( fileSize - skipBytes );
        // Create UploadPart CopyReques t . You can
specify conditions with UploadPart CopyReques t .
        var uploadPart CopyReques t = new UploadPart
CopyReques t ( targetBuck et , targetObje ct , sourceBuck et
, sourceObje ct , uploadId )
        {
            PartSize = size ,
            PartNumber = i + 1 ,
            // Beginindex is used to locate the
location that the last UploadPart Copy starts .
            BeginIndex = skipBytes
        };
        // Call the uploadPart Copy method to copy
each part .
        var uploadPart CopyResult = client . UploadPart Copy
( uploadPart CopyReques t );
        Console . WriteLine ( " UploadPart Copy : { 0 }" , i );
        partETags . Add ( uploadPart CopyResult . PartETag );
    }
    // Complete the UploadPart Copy task .
    var completeMu ltipartUpl oadRequest =
new CompleteMu ltipartUpl oadRequest ( targetBuck et ,
targetObje ct , uploadId );
    // The partETags refers to the list of partETags
saved during the multipart upload . After OSS
receives the part list submitted by the user , it
verifies the validity of each data part one by
one . After the verificati on is passed , OSS combine
these parts into a complete object .
    foreach ( var partETag in partETags )
    {
        completeMu ltipartUpl oadRequest . PartETags . Add (
partETag );
    }
    var completeMu ltipartUpl oadResult = client .
CompleteMu ltipartUpl oad ( completeMu ltipartUpl oadRequest );
    Console . WriteLine ( " CompleteMu ltipartUpl oad succeeded
" );
}
catch ( OssExcepti on ex )
{
    Console . WriteLine ( " Failed with error code : { 0 } ;
Error info : { 1 } . \ nRequestID : { 2 } \ tHostID : { 3 }" ,
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }" ,
ex . Message );
}
}

```

• Resumable copy

If the copy is interrupted, you can use resumable copy to continue the copy. For the complete code of resumable copy, see [GitHub](#).

Run the following code for resumable copy:

```
using Aliyun . OSS ;
```

```

var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var sourceBucket = "< yourSourceBucketName >";
var sourceObject = "< yourSourceObjectName >";
var targetBucket = "< yourDestBucketName >";
var targetObject = "< yourDestObjectName >";
var checkpointDir = @"< yourCheckpointDir >";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId,
    accessKeySecret);
try
{
    var request = new CopyObjectRequest(sourceBucket,
        sourceObject, targetBucket, targetObject);
    // The checkpointDir directory stores the
    intermediate state information of a resumable
    upload. The information will be used when a
    failed upload task is resumed. // If the checkpoint
    Dir directory is null, resumable copy will not take
    effect, and the object that previously failed to
    be copied will be copied all over again.
    client.ResumableCopyObject(request, checkpointDir);
    Console.WriteLine("Resumable copy new object:{0}
        succeeded", request.DestinationKey);
}
catch (Exception ex)
{
    Console.WriteLine("Resumable copy new object failed
        , {0}", ex.Message);
}

```

8.8.8 Restore an archive object

For the complete code of restoring an archive object, see [GitHub](#).

You must restore an archive object before you read it. Do not call `restoreObject` for non-archive objects.

The state conversion process of an archive object is as follows:

- An archive object is in the frozen state.
- After you submit it for restoration, the server restores the object. The object is in the restoring state.
- You can read the object after it is restored. The restored state of the object lasts one day by default. You can prolong this period to a maximum of seven days. Once this period expires, the object returns to the frozen state.

Run the following code to restore an object:

```

using Aliyun.OSS;
using Aliyun.OSS.Model;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";

```

```

var bucketName = "< yourBucket Name >";
var objectName = "< yourObject Name >";
var objectContent = "More than just cloud.";
int maxWaitTimeInSeconds = 600;
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Create a bucket of the archive storage class..
    var bucket = client . CreateBucket ( bucketName ,
StorageClass . Archive );
    Console . WriteLine ( " Create Archive bucket succeeded , {
0 } " , bucket . Name );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Create Archive bucket failed , { 0
} " , ex . Message );
}
// Upload a file .
try
{
    byte [] binaryData = Encoding . ASCII . GetBytes ( objectCont
ent );
    MemoryStream requestContent = new MemoryStream (
binaryData );
    client . PutObject ( bucketName , objectName , requestCon tent
);
    Console . WriteLine ( " Put object succeeded , { 0 } " ,
objectName );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Put object failed , { 0 } " , ex .
Message );
}
var metadata = client . GetObjectMetadata ( bucketName ,
objectName );
string storageClass = metadata . HttpMetadata [ " x - oss -
storage - class " ] as string ;
if ( storageClass != " Archive " )
{
    Console . WriteLine ( " StorageClass is { 0 } " , storageCla
ss );
    return ;
}
// Restore the object .
RestoreObjectResult result = client . RestoreObject (
bucketName , objectName );
Console . WriteLine ( " RestoreObject result HttpStatusCode :
{ 0 } " , result . HttpStatusCode );
if ( result . HttpStatusCode != HttpStatusCode . Accepted )
{
    throw new OssException ( result . RequestId + " , " +
result . HttpStatusCode + " , " );
}
while ( maxWaitTimeInSeconds > 0 )
{
    var meta = client . GetObjectMetadata ( bucketName ,
objectName );
    string restoreStatus = meta . HttpMetadata [ " x - oss -
restore " ] as string ;

```

```

        if ( restoreSta tus != null && restoreSta tus .
StartsWith ( " ongoing - request =" false \", StringComparison .
InvariantC ultureIgno reCase ))
        {
            break ;
        }
        Thread . Sleep ( 1000 );
        maxWaitTim eInSeconds --;
    }
    if ( maxWaitTim eInSeconds == 0 )
    {
        Console . WriteLine ( " RestoreObj ect is timeout . " );
        throw new TimeoutExc eption ();
    }
    else
    {
        Console . WriteLine ( " RestoreObj ect is successful . " );
    }
}

```

For more information about the archive storage classes, see [Introduction to storage classes](#). For detailed information about related status code, see [RestoreObject](#).

8.8.9 Manage a symbolic link

A symbolic link is a special object that maps to an object similar to a shortcut used in Windows. You can configure user-defined Object Meta for symbolic links. To obtain a symbolic link, you must have the read permission on it.

Run the following code to create and obtain a symbolic link:

```

using Aliyun . OSS ;
var endpoint = "< yourEndpoi nt >";
var accessKeyI d = "< yourAccess KeyId >";
var accessKeyS ecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
var targetObj e ctName = "< yourTarget ObjectName >";
var symlinkObj ectName = "< yourSymlin kObjectNam e >";
var objectCont ent = " More than just cloud . " ;
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyI d ,
accessKeyS ecret );
try
{
    // Upload the target file .
    byte [] binaryData = Encoding . ASCII . GetBytes ( objectCont
ent );
    MemoryStre am requestCon tent = new MemoryStre am (
binaryData );
    client . PutObject ( bucketName , targetObj e ctName ,
requestCon tent );
    // Create a symbolic link .
    client . CreateSyml ink ( bucketName , symlinkObj ectName ,
targetObj e ctName );
    // Obtain a symbolic link .
    var ossSymlink = client . GetSymlink ( bucketName ,
symlinkObj ectName );
    Console . WriteLine ( " Target object is { 0 }", ossSymlink
. Target );
}
}

```



```
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }", ex
. Message );
}
```

For more information about creating symbolic links, see [PutSymlink](#).

For more information about obtaining symbolic links, see [GetSymlink](#).

8.9 Authorized access

Use STS for temporary access authorization

OSS supports Alibaba Cloud Security Token Service (STS) for temporary access authorization. STS is a web service that provides a temporary access token to a cloud computing user. Through the STS, you can assign a third-party application or a RAM user (you can manage the user ID) an access credential with a custom validity period and permissions. For more information about STS, see [STS introduction](#).

STS advantages:

- Your long-term key (AccessKey) is not exposed to a third-party application. You only need to generate an access token and send the access token to the third-party application. You can customize access permissions and the validity of this token.
- You do not need to keep track of permission revocation issues. The access token automatically becomes invalid when it expires.

For more information about the process of access to OSS with STS, see [RAM and STS scenario practices](#) in OSS Developer Guide.

Run the following code to create a signature request with STS:

```
using Aliyun . OSS ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccess KeyId >";
var accessKeySecret = "< yourAccess KeySecret >";
var securityToken = "< yourSecurityToken >";
// After a user obtains a temporary STS credential ,
the OSSClient is generated with the security token
and temporary access key ( AccessKeyId and AccessKeyS
ecret ).
// Create an OSSClient instance .
var ossStsClient = new OssClient ( endpoint , accessKeyId
, accessKeySecret , securityToken );
```

```
// Perform operations on OSS .
```

Sign a URL to authorize temporary access

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time for a URL to restrict the period of access from visitors.

For the complete code of signing a URL to authorize temporary access, see [GitHub](#).

- Use a signed URL to upload a file

Run the following code to upload a file with a signed URL:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var objectContent = " More than just cloud .";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Generate a signed URL for upload .
    var generatePresignedUriRequest = new GeneratePresignedUriRequest ( bucketName , objectName , SignHttpMethod.Put )
    {
        Expiration = DateTime . Now . AddHours ( 1 ),
    };
    var signedUrl = client . GeneratePresignedUri ( generatePresignedUriRequest );
    // Upload a file with the signed URL .
    var buffer = Encoding . UTF8 . GetBytes ( objectContent );
    using ( var ms = new MemoryStream ( buffer ) )
    {
        client . PutObject ( signedUrl , ms );
    }
    Console . WriteLine ( " Put object by signature succeeded . { 0 } ", signedUrl . ToString ());
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error code : { 0 }; Error info : { 1 }. \ nRequestID :{ 2 } \ tHostID :{ 3 }",
        ex . ErrorCode , ex . Message , ex . RequestId , ex . HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }",
        ex . Message );
}
```

```
}
```

- Use a signed URL to download an object

Run the following code to download an object with a signed URL:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
var downloadFilename = "< yourDownloadFilename >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    var metadata = client . GetObjectMetadata ( bucketName ,
objectName );
    var etag = metadata . ETag ;
    // Generate a signed URL for download .
    var req = new GeneratePresignedUriRequest (
bucketName , objectName , SignHttpMethod . Get );
    var uri = client . GeneratePresignedUri ( req );
    // Download an object with the signed URL .
    OssObject ossObject = client . GetObject ( uri );
    using ( var file = File . Open ( downloadFilename ,
 FileMode . OpenOrCreate ) )
    {
        using ( Stream stream = ossObject . Content )
        {
            int length = 4 * 1024 ;
            var buf = new byte [ length ];
            do
            {
                length = stream . Read ( buf , 0 , length );
                file . Write ( buf , 0 , length );
            } while ( length != 0 );
        }
    }
    Console . WriteLine ( " Get object by signatrue
succeeded . { 0 } " , uri . ToString ( ) );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error code : { 0 } ;
Error info : { 1 } . \ nRequestID : { 2 } \ tHostID : { 3 } " ,
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " ,
ex . Message );
}
```

```
}
```

8.10 Manage lifecycle rules

You can use OSS to configure lifecycle rules to reduce storage costs. The lifecycle management rules can be used for automatic deletion of expired objects and fragments, or storage class conversion to Infrequent Access for objects that will expire soon. Each rule includes:

- **Rule ID:** It identifies a rule. Ensure that each rule ID in a bucket is unique.
- **Policy:** You can configure a policy by using the following configuration methods. Only one method can be configured for a bucket.
 - **Configure by Prefix:** You can create multiple rules with this method. Ensure that each prefix is unique.
 - **Configure for Entire Bucket:** You can configure only one rule with this method.
- **Expired:** You can configure the following configuration methods:
 - **Set by Number of Days:** It specifies the number of days from the last modification time of objects.
 - **Set by Date:** It specifies the date prior to which objects are created. If objects are created prior to the date, these objects are deleted.
- **Status:** This lifecycle rule takes effect or not.

The parts uploaded by uploadPart method also support lifecycle rules. The last modification time of an object is the time the multipart upload event is initiated.

For more information about lifecycle management, see [Manage object lifecycle](#).

For the complete code of lifecycle management, see [GitHub](#).

Configure lifecycle rules

Run the following code to configure lifecycle rules:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
```

```

        var setBucketLifecycleRequest = new SetBucketLifecycleRequest(bucketName);
        // Create the first lifecycle rule that is valid
        within 3 days.
        LifecycleRule lcr1 = new LifecycleRule()
        {
            ID = "delete obsoleted files",
            Prefix = "obsoleted /",
            Status = RuleStatus.Enabled,
            ExpirationDays = 3
        };
        // Create the second lifecycle rule.
        LifecycleRule lcr2 = new LifecycleRule()
        {
            ID = "delete temporary files",
            Prefix = "temporary /",
            Status = RuleStatus.Enabled,
            ExpirationDays = 20
        };
        setBucketLifecycleRequest.AddLifecycleRule(lcr1);
        setBucketLifecycleRequest.AddLifecycleRule(lcr2);
        // Configure the lifecycle.
        client.SetBucketLifecycle(setBucketLifecycleRequest);
        Console.WriteLine("Set bucket:{0} Lifecycle succeeded", bucketName);
    }
    catch (OssException ex)
    {
        Console.WriteLine("Failed with error code:{0}; Error info:{1}. \nRequestId:{2}\tHostId:{3}", ex.ErrorCode, ex.Message, ex.RequestId, ex.HostId);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Failed with error info:{0}", ex.Message);
    }
}

```

View lifecycle rules

For the complete code of viewing lifecycle rules, see [GitHub](#).

Run the following code to view lifecycle rules:

```

using Aliyun.OSS;
using Aliyun.OSS.Common;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId, accessKeySecret);
try
{
    // View the lifecycle rules.
    var rules = client.GetBucketLifecycle(bucketName);
    Console.WriteLine("Get bucket:{0} Lifecycle succeeded", bucketName);
    foreach (var rule in rules)
    {
    }
}

```

```

        Console.WriteLine(" ID : { 0 }", rule.ID);
        Console.WriteLine(" Prefix : { 0 }", rule.Prefix);
        Console.WriteLine(" Status : { 0 }", rule.Status);
        if (rule.ExpirationDays.HasValue)
            Console.WriteLine(" Expiration Days : { 0 }", rule.ExpirationDays);
    }
}
catch (OssException ex)
{
    Console.WriteLine(" Failed with error code : { 0 };
    Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
        ex.ErrorCode, ex.Message, ex.RequestId, ex.HostId);
}
catch (Exception ex)
{
    Console.WriteLine(" Failed with error info : { 0 }", ex.Message);
}
}

```

Clear lifecycle rules

Run the following code to clear lifecycle rules:

```

using Aliyun.OSS;
using Aliyun.OSS.Common;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId, accessKeySecret);
try
{
    // Clear the lifecycle rules
    client.DeleteBucketLifecycle(bucketName);
    Console.WriteLine(" Delete bucket :{ 0 } Lifecycle succeeded ", bucketName);
}
catch (OssException ex)
{
    Console.WriteLine(" Failed with error code : { 0 };
    Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
        ex.ErrorCode, ex.Message, ex.RequestId, ex.HostId);
}
catch (Exception ex)
{
    Console.WriteLine(" Failed with error info : { 0 }", ex.Message);
}
}

```

```
}
```

8.11 Set logging

You can enable access logs to record bucket access to log files, which are stored in a specified bucket.

The log file format is as follows:

```
< TargetPrefix >< SourceBucket >- YYYY - mm - DD - HH - MM - SS -  
UniqueString
```

For more information about access log files, see [Set access logging](#). For the complete code of access logging, see [GitHub](#).

Enable access logging

Run the following code to enable bucket access logging:

```
using Aliyun . OSS ;  
using Aliyun . OSS . Common ;  
var endpoint = "< yourEndpoint >";  
var accessKeyId = "< yourAccessKeyId >";  
var accessKeySecret = "< yourAccessKeySecret >";  
var bucketName = "< yourBucketName >";  
var targetBucketName = "< yourTargetBucketName >";  
// Create an OSSClient instance.  
var client = new OssClient ( endpoint , accessKeyId ,  
accessKeySecret );  
try  
{  
    // The targetBucketName parameter is the name of  
    the bucket that logs are stored. The logging - option  
    indicates the prefix of log files. The bucketName  
    and targetBucketName can be the same bucket.  
    var request = new SetBucketLoggingRequest ( bucketName  
    , targetBucketName , " logging -");  
    // Enable access logging.  
    client . SetBucketLogging ( request );  
    Console . WriteLine ( " Set bucket : { 0 } Logging succeeded  
    ", bucketName );  
}  
catch ( OssException ex )  
{  
    Console . WriteLine ( " Failed with error info : { 0 };  
    Error info : { 1 }. \ nRequestID : { 2 } \ tHostID : { 3 }",  
        ex . ErrorCode , ex . Message , ex . RequestId , ex .  
        HostId );  
}  
catch ( Exception ex )  
{  
    Console . WriteLine ( " Failed with error info : { 0 }", ex  
        . Message );  
}
```

```
}
```

View access logging configurations

For the complete code of viewing access logging configurations, see [GitHub](#).

Run the following code to view the access logging configurations for a bucket:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // View bucket access logging configurations .
    var result = client . GetBucketLogging ( bucketName );
    Console . WriteLine ( " Get bucket : { 0 } Logging succeeded
, prefix : { 1 } " , bucketName , result . TargetPrefix );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error info : { 0 };
Error info : { 1 } . \ nRequestId : { 2 } \ tHostID : { 3 } " ,
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " , ex
. Message );
}
```

Disable access logging

For the complete code of disabling access logging, see [GitHub](#).

Run the following code to disable access logging for a bucket:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance.
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Disable access logging . Existing log files are
not affected .
    client . DeleteBucketLogging ( bucketName );
    Console . WriteLine ( " Delete bucket : { 0 } Logging
succeeded " , bucketName );
}
```



```

catch ( OssExcepti on ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } ;
    Error info : { 1 } . \ n RequestID : { 2 } \ t HostID : { 3 } " ,
        ex . ErrorCode , ex . Message , ex . RequestId , ex .
        HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " , ex
        . Message );
}

```

8.12 CORS

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) and [PutBucketcors](#) in OSS Developer Guide.

For the complete code of CORS, see [GitHub](#).

Configure CORS rules

For complete code of configuring CORS rules, see [GitHub](#).

Run the following code to configure CORS rules for a specified bucket:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoi nt >";
var accessKeyI d = "< yourAccess KeyId >";
var accessKeyS ecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
// Creates an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyI d ,
    accessKeyS ecret );
try
{
    var request = new SetBucketC orsRequest ( bucketName );
    var rule1 = new CORSRule ();
    // Specify the allowed source of the cross - origin
    access request .
    rule1 . AddAllowed Origin ( " http :// www . a . com " );
    // Specify the cross - region request methods ( GET ,
    PUT , DELETE , POST , and HEAD ) that are allowed .
    rule1 . AddAllowed Method ( " POST " );
    // AllowedHea ders and ExposeHea ders do not support
    wildcards .
    rule1 . AddAllowed Header ( "*" );
    // Specify the response header that allows user
    access from applicatio ns .
    rule1 . AddExposeH eader ( " x - oss - test " );
    // A maximum of 10 rules is allowed .
    request . AddCORSRu l e ( rule1 );
}

```

```

        var rule2 = new CORSRule();
        // AllowedOrigins and AllowedMethods allow only one
        wildcard asterisk (*). Wildcard asterisks (*) indicate
        that all sources of the cross-origin requests and
        operations are allowed.
        rule2.AddAllowedOrigin("http://www.b.com");
        rule2.AddAllowedMethod("GET");
        // Determine whether the header specified in the
        Access-Control-Headers in the pre-flight (OPTIONS)
        requests is allowed.
        rule2.AddExposeHeader("x-oss-test2");
        // Specify the cache time (seconds) for the
        response of browser pre-flight (OPTIONS) requests to
        a specific resource.
        rule2.MaxAgeSeconds = 100;
        request.AddCORSRule(rule2);
        // Configure CORS rules.
        client.SetBucketCors(request);
        Console.WriteLine("Set bucket:{0} Cors succeeded ",
        bucketName);
    }
    catch (OssException ex)
    {
        Console.WriteLine("Failed with error info:{0};
        Error info:{1}.\nRequestId:{2}\tHostID:{3}",
        ex.ErrorCode, ex.Message, ex.RequestId, ex.
        HostId);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Failed with error info:{0}", ex
        .Message);
    }
}

```

Obtain CORS rules

For the complete code of obtaining CORS rules, see [GitHub](#).

Run the following code to obtain CORS rules:

```

using Aliyun.OSS;
using Aliyun.OSS.Common;
var endpoint = "<yourEndpoint>";
var accessKeyId = "<yourAccessKeyId>";
var accessKeySecret = "<yourAccessKeySecret>";
var bucketName = "<yourBucketName>";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId,
accessKeySecret);
try
{
    // Obtain CORS rules.
    var result = client.GetBucketCors(bucketName);
    Console.WriteLine("Get bucket:{0} Cors succeeded ",
    bucketName);
    foreach (var rule in result)
    {
        foreach (var origin in rule.AllowedOrigins)
        {
            Console.WriteLine("Allowed origin:{0}", origin
        );
        }
    }
}

```

```

    }
}
catch ( OssExcepti on ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } ;
    Error info : { 1 } . \ nRequestID :{ 2 } \ tHostID :{ 3 } " ,
        ex . ErrorCode , ex . Message , ex . RequestId , ex .
        HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " , ex
    . Message );
}

```

Delete CORS rules

For the complete code of deleting CORS rules, see [GitHub](#).

Run the following code to delete all CORS rules for a specified bucket:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoi nt >";
var accessKeyI d = "< yourAccess KeyId >";
var accessKeyS ecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyI d ,
    accessKeyS ecret );
try
{
    // Delete CORS rules .
    client . DeleteBuck etCors ( bucketName );
    Console . WriteLine ( " Delete bucket :{ 0 } Cors succeeded
    " , bucketName );
}
catch ( OssExcepti on ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } ;
    Error info : { 1 } . \ nRequestID :{ 2 } \ tHostID :{ 3 } " ,
        ex . ErrorCode , ex . Message , ex . RequestId , ex .
        HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 } " , ex
    . Message );
}

```

8.13 Static website hosting

You can set your bucket configuration to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

For more information about static website hosting, see [Static website hosting](#).

Configure static website hosting

Run the following code to configure static website hosting:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccess KeyId >";
var accessKeySecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Configure static website hosting .
    var request = new SetBucketWebsiteRequest ( bucketName
, " index . html ", " error . html " );
    client . SetBucketWebsite ( request );
    Console . WriteLine ( " Set bucket :{ 0 } Website succeeded
", bucketName );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error info : { 0 };
Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }", ex
. Message );
}
```

View static website hosting configurations

Run the following code to view static website hosting configurations:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccess KeyId >";
var accessKeySecret = "< yourAccess KeySecret >";
var bucketName = "< yourBucket Name >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // View static website hosting configurations .
    var result = client . GetBucketWebsite ( bucketName );
    Console . WriteLine ( " Get bucket :{ 0 } Website succeeded
, index doc :{ 1 }, error doc :{ 2 }",
bucketName , result . IndexDocument ,
result . ErrorDocument );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error info : { 0 };
Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
```

```

        ex . ErrorCode , ex . Message , ex . RequestId , ex .
        HostId );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Failed with error info : { 0 }", ex
        . Message );
    }

```

Delete static website hosting configurations

Run the following code to delete static website hosting configurations:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // Delete static website hosting configurations .
    client . DeleteBucketWebsite ( bucketName );
    Console . WriteLine ( " Delete bucket :{ 0 } Website
succeeded ", bucketName );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error info : { 0 };
Error info : { 1 }. \ nRequestId :{ 2 }\ tHostID :{ 3 }",
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }", ex
. Message );
}

```

8.14 Anti-leech

This topic describes how to use anti-leech.

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leeching, see [Anti-leeching settings](#).

For the complete code of configuring anti-leech, see [GitHub](#).

Configure the referer whitelist

Run the following code to configure the referer whitelist:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    var refererList = new List < string >();
    // Add the referer field . The referer field allows
    question marks (?) and asterisks (*) for wildcard use .
    refererList . Add ( " http :// www . aliyun . com " );
    refererList . Add ( " http :// www . *. com " );
    refererList . Add ( " http :// www . ?. aliyuncs . com " );
    var srq = new SetBucketRefererRequest ( bucketName ,
refererList );
    // Configure the referer list for a bucket .
    client . SetBucketReferer ( srq );
    Console . WriteLine ( " Set bucket :{ 0 } Referer succeeded
", bucketName );
}
catch ( OssException ex )
{
    Console . WriteLine ( " Failed with error code : { 0 };
Error info : { 1 } . \ nRequestID :{ 2 }\ tHostID :{ 3 }",
ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
}
catch ( Exception ex )
{
    Console . WriteLine ( " Failed with error info : { 0 }", ex
. Message );
}
```

Obtain a referer whitelist

For the complete code of obtaining a referer whitelist, see [GitHub](#).

Run the following code to obtain a referer whitelist:

```
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
```

```

        // Obtain the referer list for a bucket .
        var rc = client . GetBucketReferer ( bucketName );
        Console . WriteLine ( " Get bucket :{ 0 } Referer succeeded
", bucketName );
        Console . WriteLine ( " allow ? " + ( rc . AllowEmpty Referer ? "
yes " : " no "));
        if ( rc . RefererList . Referers != null )
        {
            for ( var i = 0 ; i < rc . RefererList . Referers .
Length ; i ++ )
                Console . WriteLine ( rc . RefererList . Referers [ i
]);
        }
        else
        {
            Console . WriteLine ( " Empty Referer List " );
        }
    }
    catch ( OssException ex )
    {
        Console . WriteLine ( " Failed with error code : { 0 };
Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID :{ 3 }",
            ex . ErrorCode , ex . Message , ex . RequestId , ex .
HostId );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Failed with error info : { 0 }", ex
. Message );
    }
    finally
    {
        client . SetBucketReferer ( new SetBucketRefererRequest (
bucketName ));
    }
}

```

Clear a referer whitelist

Run the following code to clear a referer whitelist:

```

using Aliyun . OSS ;
using Aliyun . OSS . Common ;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
// Create an OSSClient instance .
var client = new OssClient ( endpoint , accessKeyId ,
accessKeySecret );
try
{
    // You cannot clear a referer whitelist directly .
    To clear a referer whitelist , you need to create
    the rule that allows an empty referer field and
    replace the original rule with the new rule .
    var srq = new SetBucketRefererRequest ( bucketName );
    client . SetBucketReferer ( srq );
    Console . WriteLine ( " Set bucket :{ 0 } Referer succeeded
", bucketName );
}
catch ( OssException ex )
{
}

```

```
        Console . WriteLine ( " Failed with error code : { 0 } ;  
Error info : { 1 } . \ nRequestID : { 2 } \ tHostID : { 3 } " ,  
ex . ErrorCode , ex . Message , ex . RequestId , ex .  
HostId ) ;  
}  
catch ( Exception ex )  
{  
    Console . WriteLine ( " Failed with error info : { 0 } " , ex  
    . Message ) ;  
}
```

8.15 Image processing

Image Processing (IMG) is a massive, secure, cost-effective and highly reliable image processing service. After source images are uploaded to OSS, you can process images on any Internet device at any anytime, from anywhere through simple RESTful APIs.

For more information about IMG, see [Image Processing](#).

Basic features

OSS offers the following IMG features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Image processing](#)
- [Image processing access rules](#): calls multiple IMG functions.

Usage

IMG uses standard HTTP GET. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is private read and write, only authorized users are allowed for access.

- Anonymous access

You can use the following format in level-3 domains to access a processed image:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Specifies the name of a bucket.
endpoint	Specifies endpoint used to access a region.
object	Specifies the name of an image object.
image	Specifies the reserved identifier for IMG .
action	Specifies the operations on an image, such as scaling, cropping, and rotating.

Parameter	Description
param	Specifies the parameter that indicates the operation on an image.

- Basic operations

For example, scale an image to a width of 100 px. Adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customized image styles

You can use the following format in level-3 domains to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- **style**: specifies a reserved identifier of a customized image style.
- **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

Cascade operations allow you to perform multiple operations on an image sequentially.

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

Example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- HTTPS access

IMG supports access through HTTPS. Example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

Authorized access allows you to customize an image style, HTTPS access, and cascade operations.

Run the following code to generate a signed URL for IMG:

```
using Aliyun.OSS;
using Aliyun.OSS.Common;
var endpoint = "< yourEndpoint >";
var accessKeyId = "< yourAccessKeyId >";
var accessKeySecret = "< yourAccessKeySecret >";
var bucketName = "< yourBucketName >";
var objectName = "< yourObjectName >";
// Create an OSSClient instance.
var client = new OssClient(endpoint, accessKeyId, accessKeySecret);
try
{
    var process = "image/resize,m_fixed,w_100,h_100";
    var req = new GeneratePresignedUriRequest(bucketName, objectName, SignHttpMethod.Get)
    {
        Expiration = DateTime.Now.AddHours(1),
        Process = process
    };
    // Generate a signed URI.
    var uri = client.GeneratePresignedUri(req);
    Console.WriteLine("Generate Presigned Uri:{0} with process:{1} succeeded", uri, process);
}
catch (OssException ex)
{
    Console.WriteLine("Failed with error code:{0}; Error info:{1}. \nRequestID:{2}\tHostID:{3}", ex.ErrorCode, ex.Message, ex.RequestId, ex.HostId);
}
catch (Exception ex)
{
    Console.WriteLine("Failed with error info:{0}", ex.Message);
}
```

```
}
```

• Access with SDK

You can use SDK to access and process an image object.

SDK allows you to customize image styles, HTTPS access, and cascade operations.

- Basic operations

Run the following code to perform basic operations on an image:

```
using System ;
using System . IO ;
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
using Aliyun . OSS . Util ;
namespace ImageProcess
{
    class Program
    {
        static void Main ( string [] args )
        {
            Program . ImageProcess () ;
            Console . ReadKey () ;
        }
        public static void ImageProcess ()
        {
            var endpoint = "< yourEndpoint >";
            var accessKeyId = "< yourAccessKeyId >";
            var accessKeySecret = "< yourAccessKeySecret >";
            var bucketName = "< yourBucketName >";
            var objectName = "< yourObjectName >";
            var localImageFilename = "< yourLocalImageFilename >";
            var downloadDir = "< yourDownloadDir >";
            // Create an OSSClient instance .
            var client = new OssClient ( endpoint ,
            accessKeyId , accessKeySecret );
            try
            {
                client . PutObject ( bucketName , objectName ,
                localImageFilename );
                // Scale the image .
                var process = "image / resize , m_fixed ,
                w_100 , h_100 ";
                var ossObject = client . GetObject ( new
                GetObjectRequest ( bucketName , objectName , process ));
                WriteToFile ( downloadDir + "/ sample -
                resize . jpg " , ossObject . Content );
                // Crop the image .
                process = "image / crop , w_100 , h_100 , x_100
                , y_100 , r_1 ";
                ossObject = client . GetObject ( new
                GetObjectRequest ( bucketName , objectName , process ));
                WriteToFile ( downloadDir + "/ sample - crop
                . jpg " , ossObject . Content );
                // Rotate the image .
                process = "image / rotate , 90 ";
                ossObject = client . GetObject ( new
                GetObjectRequest ( bucketName , objectName , process ));
```

```

        WriteToFile ( downloadDir + "/" + sample -
rotate . jpg ", ossObject . Content );
        // Sharpen the image .
        process = " image / sharpen , 100 ";
        ossObject = client . GetObject ( new
GetObjectRequest ( bucketName , objectName , process ));
        WriteToFile ( downloadDir + "/" + sample -
sharpen . jpg ", ossObject . Content );
        // Add watermarks .
        process = " image / watermark , text_SGVsb
G8g5Zu - 54mH5pyN5Y qhIQ ";
        ossObject = client . GetObject ( new
GetObjectRequest ( bucketName , objectName , process ));
        WriteToFile ( downloadDir + "/" + sample -
watermark . jpg ", ossObject . Content );
        // Convert the image format .
        process = " image / format , png ";
        ossObject = client . GetObject ( new
GetObjectRequest ( bucketName , objectName , process ));
        WriteToFile ( downloadDir + "/" + sample -
format . png ", ossObject . Content );
        // Obtain image information .
        process = " image / info ";
        ossObject = client . GetObject ( new
GetObjectRequest ( bucketName , objectName , process ));
        WriteToFile ( downloadDir + "/" + sample - info
. txt ", ossObject . Content );
        Console . WriteLine ( " Get Object :{ 0 } with
process :{ 1 } succeeded ", objectName , process );
    }
    catch ( OssException ex )
    {
        Console . WriteLine ( " Failed with error
code : { 0 }; Error info : { 1 }. \nRequestID :{ 2 }\ tHostID
:{ 3 }",
            ex . ErrorCode , ex . Message , ex .
RequestId , ex . HostId );
    }
    catch ( Exception ex )
    {
        Console . WriteLine ( " Failed with error
info : { 0 }", ex . Message );
    }
}

private static void WriteToFile ( string
filePath , Stream stream )
{
    using ( var requestStream = stream )
    {
        using ( var fs = File . Open ( filePath ,
FileMode . OpenOrCreate ))
        {
            IoUtils . WriteTo ( stream , fs );
        }
    }
}
}

```

```
}

```

- Customize an image style

Use the following code to customize an image style:

```
using System ;
using System . IO ;
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
using Aliyun . OSS . Util ;
namespace ImageProce ssCustom
{
    class Program
    {
        static void Main ( string [] args )
        {
            Program . ImageProce ssCustomSt yle () ;
            Console . ReadKey () ;
        }
        public static void ImageProce ssCustomSt yle ()
        {
            var endpoint = "< yourEndpoi nt >";
            var accessKeyI d = "< yourAccess KeyId >";
            var accessKeyS ecret = "< yourAccess KeySecret
>";
            var bucketName = "< yourBucket Name >";
            var objectName = "< yourObject Name >";
            var localImage Filename = "< yourLocalI
mageFilena me >";
            var downloadDi r = "< yourDownlo adDir >";
            // Create an OSSClient instance .
            var client = new OssClient ( endpoint ,
accessKeyI d , accessKeyS ecret );
            try
            {
                client . PutObject ( bucketName , objectName ,
localImage Filename );
                // Customize the image style : " style /<
yourCustom StyleName >".
                var process = " style / oss - pic - style - w
- 100 ";
                var ossObject = client . GetObject ( new
GetObjectR equest ( bucketName , objectName , process ));
                WriteToFil e ( downloadDi r + "/ sample -
style . jpg " , ossObject . Content );
                Console . WriteLine ( " Get Object :{ 0 } with
process :{ 1 } succeeded " , objectName , process );
            }
            catch ( OssExcepti on ex )
            {
                Console . WriteLine ( " Failed with error
code : { 0 } ; Error info : { 1 } . \ nRequestID :{ 2 } \ tHostID
:{ 3 }" ,
ex . ErrorCode , ex . Message , ex .
RequestId , ex . HostId );
            }
            catch ( Exception ex )
            {
                Console . WriteLine ( " Failed with error
info : { 0 }" , ex . Message );
            }
        }
    }
}
```

```

        private static void WriteToFile ( string
filePath , Stream stream )
        {
            using ( var requestStream = stream )
            {
                using ( var fs = File . Open ( filePath ,
FileMode . OpenOrCreate ))
                {
                    IoUtils . WriteTo ( stream , fs );
                }
            }
        }
    }
}

```

- Cascade operations

Use the following code to perform cascade operations on an image:

```

using System ;
using System . IO ;
using Aliyun . OSS ;
using Aliyun . OSS . Common ;
using Aliyun . OSS . Util ;
namespace ImageProcessCascade
{
    class Program
    {
        static void Main ( string [] args )
        {
            Program . ImageProcessCascade () ;
            Console . ReadKey () ;
        }
        public static void ImageProcessCascade ()
        {
            var endpoint = "< yourEndpoint >";
            var accessKeyId = "< yourAccessKeyId >";
            var accessKeySecret = "< yourAccessKeySecret >";
            var bucketName = "< yourBucketName >";
            var objectName = "< yourObjectName >";
            var localImageFilename = "< yourLocalImageFilename >";
            var downloadDir = "< yourDownloadDir >";
            // Create an OSSClient instance .
            var client = new OssClient ( endpoint ,
accessKeyId , accessKeySecret );
            try
            {
                client . PutObject ( bucketName , objectName ,
localImageFilename );
                // Perform cascade operations .
                var process = "image / resize , m_fixed ,
w_100 , h_100 / rotate , 90 ";
                var ossObject = client . GetObject ( new
GetObjectRequest ( bucketName , objectName , process ));
                WriteToFile ( downloadDir + "/ sample -
style . jpg " , ossObject . Content );
                Console . WriteLine ( " Get Object :{ 0 } with
process :{ 1 } succeeded " , objectName , process );
            }
            catch ( OssException ex )
            {
            }
        }
    }
}

```

```

        Console.WriteLine("Failed with error
code : { 0 }; Error info : { 1 }. \ nRequestID :{ 2 }\ tHostID
:{ 3 }",
        ex.ErrorCode, ex.Message, ex.
RequestId, ex.HostId);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Failed with error
info : { 0 }", ex.Message);
    }
}
private static void WriteToFile ( string
filePath, Stream stream )
{
    using ( var requestStream = stream )
    {
        using ( var fs = File.Open ( filePath,
        FileMode.OpenOrCreate ))
        {
            IoUtils.WriteTo ( stream, fs );
        }
    }
}
}
}
}

```

IMG tools

You can use the IMG viewer [ImageStyleViewer](#) to directly view the IMG result.

8.16 Exception handling

OSS .NET SDK has two types of exceptions: `ClientException` and `OSSEException`. Both share the properties of `RuntimeException`.

ClientException

The `ClientException` exception occurs when the client attempts to send requests and when the data is transmitted to OSS. For example, when a request is sent, `ClientException` may occur due to unavailable network connections. `ClientException` may occur during file uploads due to I/O exceptions.

OSSEException

`OSSEException` indicates a server exception. The exception occurs because a server error message is parsed. `OSSEException` includes the error code and information returned from OSS. It is used to locate and handle problems.

The following table describes common `OSSEException` information.

Parameter	Description
Code	Specifies the error code returned by OSS.
Message	Specifies the detailed error information returned by OSS.
RequestId	Specifies the UUID. It is unique and used to identify a request. If a problem persists, send the RequestId to OSS developers for help.
HostId	Identifies an OSS cluster. Ensure that the value of HostId is consistent with the Host (endpoint to access a bucket) used in the request.

OSS error codes

Error code	Description
AccessDenied	Access denied
BucketAlreadyExists	The bucket already exists.
BucketNotEmpty	The bucket is not empty.
EntityTooLarge	The entity size exceeds the maximum limit.
EntityTooSmall	The entity size is below the minimum limit.
FileGroupTooLarge	The total file group size exceeds the maximum limit.
FilePartNotExist	No such part exists.
FilePartStale	The part has expired.
InvalidArgument	The parameter format is invalid.
InvalidAccessKeyId	No such AccessKeyId exists.
InvalidBucketName	The bucket name is invalid.
InvalidDigest	The digest is invalid.
InvalidObjectName	The object name is invalid.
InvalidPart	The part is invalid.
InvalidPartOrder	The part order is invalid.

Error code	Description
InvalidTargetBucketForLogging	The buckets that store log files are invalid .
InternalError	Internal OSS error
MalformedXML	The XML format is invalid.
MethodNotAllowed	The method is not allowed.
MissingArgument	Parameters are not configured.
MissingContentLength	The content length is not configured.
NoSuchBucket	No such bucket exists.
NoSuchKey	No such object exists.
NoSuchUpload	No such uploadId exists.
NotImplemented	The methods are not implemented.
PreconditionFailed	Precondition error
RequestTimeTooSkewed	The local time set for OSSClient is deviated from the time set for the OSS server by over 15 minutes.
RequestTimeout	Request timeout
SignatureDoesNotMatch	Signature mismatch
TooManyBuckets	The number of buckets exceeds the limit.

9 Android

9.1 Preface

This topic is based on OSS Android SDK 2.9.1.

Compatibility

Later versions of OSS Android SDK are backward-compatible with earlier versions.

Source code

For the source code of OSS Android SDK, see [GitHub](#).

Code samples

OSS SDK for Android provides a variety of code samples for your reference.

The following table describes the content of code samples.

Sample file	Content
BaseTestCase.java	Initialization
OSSPutObjectTest.java	Upload objects
OSSGetObjectTest.java	Download objects
ManageObjectTest.java	Manage objects
OSSBucketTest.java	Bucket
ResumableUploadTest.java	Resumable upload
CRC64Test.java	Data security
ImagePersistTest.java	Image processing

9.2 Installation

This topic describes how to install OSS Android SDK.

Environment requirements

- Android system version: 2.3 or later
- You must have an Alibaba Cloud account and activate the OSS service.

Download OSS Android SDK

- Add OSS Android SDK.

Add the following dependency in Android Studio:

```
dependencies { compile 'com.aliyun.dpa:oss-android-sdk:+' }
```

- Download OSS Android SDK from [Github](#).

Install OSS Android SDK

- Method 1 (recommended): Add the following dependency in your Maven project.

```
dependencies {
    compile 'com.aliyun.dpa:oss-android-sdk:+'
}
```

- Method 2: Compile the source code into a jar package.

1. Clone the source code of the project and run the `gradle` command to compile the code into a jar package:

```
# Clone the project .
$ git clone https://github.com/aliyun/aliyun-oss-android-sdk.git

# Enter the directory of the project .
$ cd aliyun-oss-android-sdk/oss-android-sdk/

# Run the package script . Note that jdk 1.7 is required .
$ ../gradlew releaseJar

# Enter the directory where the generated jar package is stored .
$ cd build/libs && ls
```

2. Directly introduce the generated jar package.

Introduce the following three jar packages to the `libs` directory: `aliyun-oss-sdk-android-x.x.x.jar`, [okhttp-3.x.x.jar](#), and [okio-1.x.x.jar](#).

Related settings

- Permission settings

The following permissions are required to use the OSS Android SDK.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

```
< uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
< uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
```

**Note:**

Ensure the preceding permissions are set in your `AndroidManifest.xml` file to let SDK work normally.

- Proguard settings

Add the following code to the Proguard settings:

```
- keep class com.alibaba.sdk.android.oss.** { *; }
- dontwarn okio.**
- dontwarn org.apache.commons.codec.binary.**
```

9.3 Quick start

This topic describes how to use OSS Android SDK to quickly perform common OSS operations, such as creating a bucket, uploading an object, and downloading an object.

You can use the sample project in the following two methods:

- View the `sample` directory which includes sample projects of uploading local file, downloading objects, resumable upload, and setting callback. For more information, see [Github](#).
- Run the `git clone` command to clone the [project](#).

Before running the project, you must configure the Config parameter. The sample code of configuring this parameter is as follows:

```
public class Config {
    // To run the sample correctly, the following
    // variables must have valid values.
    // The endpoint value below is just the example.
    // Please use proper value according to your region
    // Specify the endpoint that you want to access.
    public static final String OSS_ENDPOINT = "http://oss-cn-shanghai.aliyuncs.com";
    // Specify the URL used to test the callback
    // function.
    public static final String OSS_CALLBACK_URL = "http://oss-demo.aliyuncs.com:23450";
    // Specify the address of the authentication server.
    // You can also start the local STS authentication
    // server by running the local authentication
```

```

script in the sts_local_server directory of the
project .
public static final String STS_SERVER_URL = " http
: // **** / sts / getsts "; // Specify the URL of the STS
authentication server .

public static final String BUCKET_NAME = " Input the
bucket name .";
public static final String OSS_ACCESS_KEY_ID = "<
yourAccess KeyId >";
public static final String OSS_ACCESS_KEY_SECRET =
"< yourAccess KeySecret >";

public static final int DOWNLOAD_SUCCESS = 1 ;
public static final int DOWNLOAD_FAIL = 2 ;
public static final int UPLOAD_SUCCESS = 3 ;
public static final int UPLOAD_FAIL = 4 ;
public static final int UPLOAD_PROGRESS = 5 ;
public static final int LIST_SUCCESS = 6 ;
public static final int HEAD_SUCCESS = 7 ;
public static final int RESUMABLE_SUCCESS = 8 ;
public static final int SIGN_SUCCESS = 9 ;
public static final int BUCKET_SUCCESS = 10 ;
public static final int GET_STS_SUCCESS = 11 ;
public static final int MULTIPART_SUCCESS = 12 ;
public static final int STS_TOKEN_SUCCESS = 13 ;
public static final int FAIL = 9999 ;
public static final int REQUESTCODE_AUTH = 10111 ;
public static final int REQUESTCODE_LOCALPHOTO =
10112 ;
}

```

**Note:**

- For more information about how to configure the URL of the STS authentication server, see [Set up direct data transfer for mobile apps](#).
- For more information about the script in the sts_local_server directory, see [GitHub](#).

Create a bucket

A bucket is a container for objects stored in OSS. Every object is contained in a bucket

.

Run the following code to create a bucket:

```

CreateBucketRequest createBucketRequest = new CreateBucket
Request("< bucketName >");
// Set the ACL for the bucket to public-read. The
default ACL for a created bucket is private.
createBucketRequest.setBucketACL(CannedAccessControlList
.PublicRead);
// Specify the region where the bucket belongs to.
createBucketRequest.setLocationConstraint("oss-cn-
hangzhou");

```

```

OSSAsyncTask createTask = oss.asyncCreateBucket(
    createBucketRequest, new OSSCompletedCallback<CreateBucketRequest,
    CreateBucketResult>() {
        @Override
        public void onSuccess(CreateBucketRequest request,
            CreateBucketResult result) {
            Log.d("locationConstraint", request.getLocationConstraint());
        }
        @Override
        public void onFailure(CreateBucketRequest request,
            ClientException clientException, ServiceException
            serviceException) {
            // Handle the exceptions returned for the request
            if (clientException != null) {
                // A local exception (such as network
                exception) occurs.
                clientException.printStackTrace();
            }
            if (serviceException != null) {
                // A service exception occurs.
                Log.e("ErrorCode", serviceException.getErrorCode());
                Log.e("RequestId", serviceException.getRequestId());
                Log.e("HostId", serviceException.getHostId());
                Log.e("RawMessage", serviceException.getRawMessage());
            }
        }
    });

```

For the naming conventions for a bucket, see [Basic concepts](#). For more information about how to create a bucket, see [Manage buckets](#).

For more information about the endpoints, see [Regions and endpoints](#).

Upload an object

Run the following code to upload a specified local file to OSS:

```

// Construct an upload request.
PutObjectRequest put = new PutObjectRequest("< bucketName >", "< objectName >", "< uploadFile Path >");

// You can configure the callback function when calling
// the asynchronous interface to upload a file.
put.setProgressCallback(new OSSProgressCallback<PutObjectRequest>() {
    @Override
    public void onProgress(PutObjectRequest request, long
        currentSize, long totalSize) {
        Log.d("PutObject", "currentSize: " + currentSize
            + " totalSize: " + totalSize);
    }
});

OSSAsyncTask task = oss.asyncPutObject(put, new
    OSSCompletedCallback<PutObjectRequest, PutObjectResult>() {

```

```

    @ Override
    public void onSuccess ( PutObjectR equest request ,
PutObjectR esult result ) {
        Log . d ( " PutObject ", " UploadSucc ess " );
        Log . d ( " ETag ", result . getETag ());
        Log . d ( " RequestId ", result . getRequest Id ());
    }

    @ Override
    public void onFailure ( PutObjectR equest request
, ClientExce ption clientExce pion , ServiceExc eption
serviceExc eption ) {
        // Handle the exceptions returned for the request
        .
        if ( clientExce pion != null ) {
            // A local exception ( such as network
exception ) occurs .
            clientExce pion . printStack Trace ();
        }
        if ( serviceExc eption != null ) {
            // A service exception occurs .
            Log . e ( " ErrorCode ", serviceExc eption . getErrorCo
de ());
            Log . e ( " RequestId ", serviceExc eption . getRequest
Id ());
            Log . e ( " HostId ", serviceExc eption . getHostId ());
            Log . e ( " RawMessage ", serviceExc eption .
getRawMess age ());
        }
    }
});
// task . cancel (); // You can cancel the upload task .
// task . waitUntilF inished (); // You can also wait until
the task is complete .

```

Download a specified object

Run the following code to download a specified object to the local device.

```

// Construct a download request .
GetObjectR equest get = new GetObjectR equest ("< bucketName
>", "< objectName >");

OSSAsyncTa sk task = oss . asyncGetOb ject ( get , new
OSSComple tedCallback < GetObjectR equest , GetObjectR esult >()
{
    @ Override
    public void onSuccess ( GetObjectR equest request ,
GetObjectR esult result ) {
        // The request is successful .
        Log . d ( " asyncGetOb ject ", " DownloadSu ccess " );
        Log . d ( " Content - Length ", "" + result . getContent
Length ());

        InputStrea m inputStrea m = result . getObjectC ontent
();
        byte [] buffer = new byte [ 2048 ];
        int len ;

        try {
            while (( len = inputStrea m . read ( buffer )) != - 1
) {

```



```

        // You can add your own code here to
        process the downloaded data .
    }
    } catch ( IOException e ) {
        e . printStack Trace ();
    }
}

@Override
// If the GetObject request is successful , GetObjectR
// esult which includes an instance of the input stream
// is returned . Add your own code to process the
// input stream .
public void onFailure ( GetObjectR equest request
, ClientExce ption clientExce ption , ServiceExc eption
serviceExc eption ) {
    // Handle exceptions returned to the request .
    if ( clientExce ption != null ) {
        // A local exception ( such as network
        exception ) occurs .
        clientExce ption . printStack Trace ();
    }
    if ( serviceExc eption != null ) {
        // A service exception occurs .
        Log . e ( " ErrorCode ", serviceExc eption . getErrorCo
de ());
        Log . e ( " RequestId ", serviceExc eption . getRequest
Id ());
        Log . e ( " HostId ", serviceExc eption . getHostId ());
        Log . e ( " RawMessage ", serviceExc eption .
getRawMess age ());
    }
}
});

// task . cancel (); // You can cancel the download task .
// task . waitUntilF inished (); // You can also wait
// untile the task is complete .

```

**Note:**

You must process the returned input stream by running your own code.

9.4 Initialization

OSSClient is the client of OSS for Android systems. It provides the caller with a series of methods to operate and manage the buckets and objects. Before you use the OSS Android SDK to initiate a request to OSS, you must first initialize an OSSClient instance and complete necessary settings.

**Note:**

Make sure that the life cycle of the OSSClient is consistent with that of the application. Create a global OSSClient instance when an application starts, and destroy it when the application ends.

Determine an endpoint

An endpoint is the address of Alibaba Cloud OSS in a region. It supports the following two formats:

Endpoint type	Description
OSS region address	Address of the region where an OSS bucket is located. For more information about the endpoints in various regions, see Regions and endpoints .
User-defined domain name	Domain name defined by the user, with the CNAME directing to the OSS domain

- OSS region address

You can use one of the following two methods to search for an endpoint mapped to the address of the region where an OSS bucket is located:

- You can query the mapping relationship between the endpoint and the region. For more information, see [Regions and endpoints](#).
- You can log on to [Alibaba Cloud OSS Console](#), open the Bucket Overview page, and find the suffix of the bucket domain. For example, the suffix `oss - cn - hangzhou . aliyuncs . com` of the bucket domain `bucket - 1 . oss - cn - hangzhou . aliyuncs . com` is the endpoint of the bucket on the Internet.

- CNAME

You can bind your domain to a bucket through the CNAME and access the objects in the bucket through the domain.

For example, you want to bind the domain `new-image.xxxxx.com` to a bucket named “image” in the Shenzhen region: you must contact your domain hosting provider for `xxxxx.com` to configure a new domain name resolution record used to resolve `http://new-image.xxxxx.com` to `http://image.oss-cn-shenzhen.aliyuncs.com`. The record type is CNAME.

Configure the endpoint and credential

A mobile client is an untrusted environment. An extremely high risk occurs if you store the `AccessKeyId` and `AccessKeySecret` in a mobile client for signing requests. We recommend you use `STS Authentication Mode` or `Self -`

signed Mode . For more information, see [Authorized access](#) and [Set up direct data transfer for mobile apps](#).



Note:

If you are using STS authentication mode, we recommend you use `OSSAuthCredentialsProvider` to access the application server. Expired tokens are updated automatically.

For more information about sample projects, see [GitHub](#).

An example of `EndPoint` and `CredentialProvider` settings is shown as follows:

```
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

String stsServer = "STS application server address , such as http://abc.com "
// We recommend you use OSSAuthCredentialsProvider because expired tokens can be updated quickly .
OSSCredentialProvider credentialProvider = new OSSAuthCredentialsProvider ( stsServer );

// If this configuration class is not set , default configuration applies .
ClientConfiguration conf = new ClientConfiguration ();
conf.setConnectTimeout ( 15 * 1000 ); // Connection time - out . The default value is 15 seconds .
conf.setSocketTimeout ( 15 * 1000 ); // Socket time - out . The default value is 15 seconds .
conf.setMaxConcurrentRequest ( 5 ); // The maximum number of concurrent requests . The default value is 5 .
conf.setMaxErrorRetry ( 2 ); // The maximum number of retries after a failure . The default value is 2 .

OSS oss = new OSSClient ( getApplicationContext (), endpoint , credentialProvider );
```



Note:

For more authentication modes, see [Authorized access](#).

Configure the endpoint according to the CNAME

If you have bound the CNAME to the bucket, you can direct the endpoint to the CNAME directly. For example,

```
String endpoint = "http://new-image.xxxx.com";
...
```

```
OSS oss = new OSSClient ( getApplicationContext (),
    endpoint , credential Provider );
```

Enable logging

Mobile terminals have complicated environments during usage. The OSS SDK may fail to function in a region or during a period of time. To help developers locate issues, the OSS SDK records some logs to a local directory after the log feature is enabled. To enable the log feature, you must initialize the OSSClient before using it, by calling the following method.

```
// The log style .
// You can call the OSSLog . enableLog () method to
// enable displaying logs in the console .
// This setting also supports saving one copy of the
// log file to the SD card of the mobile phone at
// \ OSSLog \ logs . csv . This setting is not enabled by
// default .
// The log records the request data , returned data and
// exceptions of OSS operations .
// Such as requestId and response header . The following
// is a log record case .
// android_version : 5 . 1 . The Android version
// mobile_model : XT1085 . The Android phone model
// network_state : connected . The network status
// network_type : WIFI . The network connection type
// Specific operation information :
// [ 2017 - 09 - 05 16 : 54 : 52 ] - Encounter local exception
// : // java . lang . IllegalArgumentException : The bucket
// name is invalid .
// A bucket name must :
// 1 ) be comprised of lower - case characters , numbers
// or dash (-);
// 2 ) start with lower case or numbers ;
// 3 ) be between 3 - 63 characters long .
// -----> end of log
OSSLog . enableLog (); // Call this method to enable logs .
```



Note:

- Log files are stored on the SD card at \ OSSLog \ logs . csv .
- You can upload the file to the server to further trace the issue.
- You can also upload the log files after connecting to Alibaba Cloud Log Service. Go to [Log Service](#).

Set network parameters

You can set detailed ClientConfiguration during the initialization process as follows:

```
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";
```

```
// We recommend you initialize OSSClient using STS
// on the mobile client. For more information, see: [
// Resource Access Management].
OSSCredentialProvider credentialProvider = new
OSSStsTokenCredentialProvider("< StsToken . AccessKeyId >", "<
StsToken . SecretKeyId >", "< StsToken . SecurityToken >");

ClientConfiguration conf = new ClientConfiguration();
conf.setConnectTimeout(15 * 1000); // Connection time
// out. The default value is 15 seconds.
conf.setSocketTimeout(15 * 1000); // Socket time - out.
// The default value is 15 seconds.
conf.setMaxConcurrentTaskNum(5); // The maximum number
// of concurrent requests. The default value is 5.
conf.setMaxErrorRetry(2); // The maximum number of
// retries after a failure. The default value is 2.

OSS oss = new OSSClient(getApplicationContext(),
endpoint, credentialProvider, conf);
```

Synchronous and asynchronous interfaces in the SDK

Network requests are not allowed to be processed in the UI thread in mobile development, so most SDK interfaces support synchronous and asynchronous calls to handle requests. After it has been called, the synchronous interface blocks other requests while waiting for the returned results. The asynchronous interface has to import a callback function that processes the result of requests.

Synchronous interfaces cannot be called on the UI thread. If a `ClientException` or `ServiceException` occurs, it is thrown immediately. `ClientException` refers to local exceptions, including network exceptions, invalid parameters, and so on. `ServiceException` refers to service exceptions returned by OSS, such as authentication failures and server errors, among others.

If an exception occurs while an asynchronous request is being processed, it is handled by a callback function.

When an asynchronous interface is called, the function returns a task. You can cancel a task, wait for it to finish, or get the results directly. For example:

```
OSSAsyncTask task = oss.asyncGetObject(...);
task.cancel(); // Cancel the task
task.waitForFinished(); // Wait until the task is
// finished
GetObjectResult result = task.getResult(); // Block other
// requests and wait for the returned results
```



Note:

The interface supports both the synchronous and asynchronous call methods. For simplicity, this document only provides examples of how to call important interfaces in synchronous and asynchronous modes. For other interfaces, only asynchronous call examples are provided.

Set whether to enable the DNS settings

```
ClientConfiguration conf = new ClientConfiguration();
conf.setHttpDnsEnable(true); // The default value is "
true", meaning enable. To disable the feature, set
the value to "false".
```

Set the custom user-agent

```
ClientConfiguration conf = new ClientConfiguration();
// The set ua is added to the end of the default
ua of the SDK by default. Use the last ua value.
conf.setUserAgentMark("customUser Agent");
```

9.5 Manage buckets

A bucket serves as a container that stores objects. Objects belong to a bucket.

OSS uses buckets as the namespaces and management objects for advanced features such as billing, permission control, and logging. The bucket name is globally unique throughout OSS and cannot be changed. Every object stored on OSS must be included in a bucket.

Create a bucket

The following code creates a bucket:

```
CreateBucketRequest createBucketRequest = new CreateBucket
Request("< bucketName >");
createBucketRequest.setBucketACL(CannedAccessControlList
.PublicRead); // Specify the ACL permission of the
bucket.
createBucketRequest.setLocationConstraint("oss-cn-
hangzhou"); // Specify the data center that stores the
bucket.
OSSAsyncTask createTask = oss.asyncCreateBucket(
createBucketRequest, new OSSCompletedCallback<CreateBucket
Request, CreateBucketResult>() {
    @Override
    public void onSuccess(CreateBucketRequest request,
CreateBucketResult result) {
        Log.d("locationConstraint", request.getLocation
Constraint());
    }
    @Override
    public void onFailure(CreateBucketRequest request
, ClientException clientException, ServiceException
serviceException) {
```

```

        // Request exception
        if ( clientException != null ) {
            // Local exception , such as a network
            clientException . printStack Trace ();
        }
        if ( serviceException != null ) {
            // Service exception
            Log . e ( " ErrorCode ", serviceException . getErrorCo
de ());
            Log . e ( " RequestId ", serviceException . getRequest
Id ());
            Log . e ( " HostId ", serviceException . getHostId ());
            Log . e ( " RawMessage ", serviceException .
getRawMess age ());
        }
    }
});

```

The preceding code specifies the ACL and data center for the created bucket.

- Each user can have a maximum 30 buckets.
- The name of each bucket is globally unique. If the name of a newly created bucket is the same as an in-service bucket that belongs to another user, the creation fails.
- When you create a bucket, you can configure the bucket ACL. If no ACL is configured, by default the bucket permission is private.
- The result of successful bucket creation be returned to the data center where the bucket is located.

Get the bucket ACL permission

The following code gets the ACL of a bucket:

```

GetBucketA CLRequest getBucketA CLRequest = new GetBucketA
CLRequest ("< bucketName >");
OSSAsyncTa sk getBucketA cLTask = oss . asyncGetBu cketACL (
getBucketA CLRequest , new OSSComple tedCallback < GetBucketA
CLRequest , GetBucketA CLResult >() {
    @ Override
    public void onSuccess ( GetBucketA CLRequest request ,
GetBucketA CLResult result ) {
        Log . d ( " BucketAcl ", result . getBucketA CL ());
        Log . d ( " Owner ", result . getBucketO wner ());
        Log . d ( " ID ", result . getBucketO wnerID ());
    }
    @ Override
    public void onFailure ( GetBucketA CLRequest request
, ClientExce ption clientExce ption , ServiceExc eption
serviceExc eption ) {
        // Request exception
        if ( clientException != null ) {
            // Local exception , such as a network
            clientException . printStack Trace ();
        }
        if ( serviceException != null ) {
            // Service exception

```

```

        de ());
        Log . e (" ErrorCode ", serviceExc eption . getErrorCo
        Id ());
        Log . e (" RequestId ", serviceExc eption . getRequest
        Log . e (" HostId ", serviceExc eption . getHostId ());
        Log . e (" RawMessage ", serviceExc eption .
        getRawMess age ());
    }
}
});

```

When you use the preceding code to obtain the bucket ACL permission, consider the following points:

- Currently, three access permissions are available for a bucket: public-read-write, public-read, and private.
- Only the bucket owner can use the GetBucketACL interface.
- The returned result contains the bucket ACL permission, ID, and name (consistent with the ID) of the bucket owner.

Delete a bucket

The following code deletes a bucket:

```

DeleteBuck etRequest deleteBuck etRequest = new DeleteBuck
etRequest ("< bucketName >");
OSSAsyncTa sk deleteBuck etTask = oss . asyncDelet eBucket (
deleteBuck etRequest , new OSSComple t edCallback < DeleteBuck
etRequest , DeleteBuck etResult >() {
    @ Override
    public void onSuccess ( DeleteBuck etRequest request ,
DeleteBuck etResult result ) {
        Log . d (" DeleteBuck et ", " Success !") ;
    }
    @ Override
    public void onFailure ( DeleteBuck etRequest request
, ClientExce ption clientExce ption , ServiceExc eption
serviceExc eption ) {
        // Request exception
        if ( clientExce ption != null ) {
            // Local exception , such as a network
            exception
            clientExce ption . printStack Trace ();
        }
        if ( serviceExc eption != null ) {
            // Service exception
            Log . e (" ErrorCode ", serviceExc eption . getErrorCo
de ());
            Log . e (" RequestId ", serviceExc eption . getRequest
Id ());
            Log . e (" HostId ", serviceExc eption . getHostId ());
            Log . e (" RawMessage ", serviceExc eption .
getRawMess age ());
        }
    }
}

```



```
});
```

**Notice:**

- To prevent accidental deletion, OSS does not allow users to delete a non-empty bucket.
- Only the bucket owner has the permission to delete the bucket.

9.6 Upload objects

9.6.1 Overview

This topic describes the methods provided by OSS Android SDK to upload objects.

In OSS, an object is the basic unit for data operations. OSS Android SDK provides the following object upload methods:

- **Simple upload**: Supports uploading local files and binary `byte[]` arrays that are smaller than or equal to 5 GB.
- **Append upload**: Supports uploading files that are smaller than or equal to 5 GB.
- **Resumable upload**: Supports concurrent resumable upload tasks and custom part size. We recommend you use this method when uploading files that are larger than 5 GB and smaller than or equal to 48.8 TB.

**Note:**

For more information about the application scenarios for the preceding upload methods, see Upload objects in the OSS Developer Guide.

When uploading a file, you can view the upload progress through the [progress bar](#). After a file is uploaded, you can perform [upload callback](#).

9.6.2 Simple upload

This topic describes how to upload a file in the simple upload method.

Upload a local file

- Upload a local file by calling the synchronous interface:

```
// Construct an upload request .
PutObjectRequest request = new PutObjectRequest("<
bucketName >", "< objectName >", "< uploadFile Path >");

// The setting of the object metadata is optional .
// ObjectMeta data metadata = new ObjectMeta data ();
```

```
// metadata . setContent Type (" applicatio n / octet - stream
"); // Specify the content - type .
// metadata . setContent MD5 ( BinaryUtil . calculateB ase64Md5 (
uploadFile Path )); // Verify the MD5 value .
// put . setMetadat a ( metadata );

try {
    PutObjectR esult putResult = oss . putObject ( put );

    Log . d (" PutObject ", " UploadSucc ess ");
    Log . d (" ETag ", putResult . getETag ());
    Log . d (" RequestId ", putResult . getRequest Id ());
} catch ( ClientExce ption e ) {
    // A local exception ( such as network exception )
    occurs .
    e . printStack Trace ();
} catch ( ServiceExc eption e ) {
    // A service error occurs .
    Log . e (" RequestId ", e . getRequest Id ());
    Log . e (" ErrorCode ", e . getErrorCo de ());
    Log . e (" HostId ", e . getHostId ());
    Log . e (" RawMessage ", e . getRawMess age ());
}
```

**Note:**

In OSS Android SDK, you can call the synchronous interface only in child threads but not UI treads. Otherwise, an exception occurs. If you want to upload the file in a UI tread, call the asynchronous interface.

- Upload a local file by calling the asynchronous interface:

```
// Construct an upload request .
PutObjectR equest put = new PutObjectR equest ("<
bucketName >", "< objectName >", "< uploadFile Path >");

// You can set progress callback when uploading a
file by calling the asynchrono us interface .
put . setProgres sCallback ( new OSSProgres sCallback <
PutObjectR equest >() {
    @ Override
    public void onProgress ( PutObjectR equest request ,
long currentSiz e , long totalSize ) {
        Log . d (" PutObject ", " currentSiz e : " + currentSiz
e + " totalSize : " + totalSize );
    }
});

OSSAsyncTa sk task = oss . asyncPutOb ject ( put , new
OSSComple tedCallback < PutObjectR equest , PutObjectR esult
>() {
    @ Override
    public void onSuccess ( PutObjectR equest request ,
PutObjectR esult result ) {
        Log . d (" PutObject ", " UploadSucc ess ");
        Log . d (" ETag ", result . getETag ());
        Log . d (" RequestId ", result . getRequest Id ());
    }

    @ Override
```

```

        public void onFailure ( PutObjectRequest request
        , ClientException clientException , ServiceException
        serviceException ) {
            // A request exception occurs .
            if ( clientException != null ) {
                // A local exception ( such as network
                exception ) occurs .
                clientException.printStackTrace();
            }
            if ( serviceException != null ) {
                // A service exception occurs .
                Log . e ( " ErrorCode ", serviceException .
                getErrorCode () );
                Log . e ( " RequestId ", serviceException .
                getRequestId () );
                Log . e ( " HostId ", serviceException . getHostId
                ());
                Log . e ( " RawMessage ", serviceException .
                getRawMessage () );
            }
        }
    });
    // task . cancel (); // You can cancel the task .
    // task . waitUntilFinished (); // You can also wait
    until the task is complete .

```

Upload a binary byte[] array

Run the following code to upload a binary byte[] array:

```

byte [] uploadData = new byte [ 100 * 1024 ];
new Random (). nextBytes ( uploadData );

// Construct an upload request .
PutObjectRequest put = new PutObjectRequest ( "< bucketName
>", "< objectName >", uploadData );

try {
    PutObjectResult putResult = oss . putObject ( put );

    Log . d ( " PutObject ", " UploadSuccess " );
    Log . d ( " ETag ", putResult . getETag () );
    Log . d ( " RequestId ", putResult . getRequestId () );
} catch ( ClientException e ) {
    // A local exception ( such as network exception )
    occurs .
    e . printStackTrace ();
} catch ( ServiceException e ) {
    // A service exception occurs .
    Log . e ( " RequestId ", e . getRequestId () );
    Log . e ( " ErrorCode ", e . getErrorCode () );
    Log . e ( " HostId ", e . getHostId () );
    Log . e ( " RawMessage ", e . getRawMessage () );
}

```

```
}
```

9.6.3 Append upload

This topic describes how to upload a file in the append upload method, in which the uploaded file is appended to an existing object.

An object created by the AppendObject operation is an appendable object, and an object created by the PutObject operation is a normal object.

Run the following code to upload a file in the append upload method:

```
AppendObjectRequest append = new AppendObjectRequest("<
bucketName >", "< objectName >", "< uploadFile Path >");

ObjectMetadata metadata = new ObjectMetadata();
metadata.setContentType("application/octet-stream");
append.setMetadata(metadata);

// Specify the position where the uploaded data is
// appended.
append.setPosition(0);

// Set progress callback.
append.setProgressCallback(new OSSProgressCallback<
AppendObjectRequest>() {
    @Override
    public void onProgress(AppendObjectRequest request,
        long currentSize, long totalSize) {
        Log.d("AppendObject", "currentSize: " +
currentSize + " totalSize: " + totalSize);
    }
});
// Perform append upload by calling the asynchronous
// interface.
OSSAsyncTask task = oss.asyncAppendObject(append,
new OSSCompletedCallback<AppendObjectRequest, AppendObject
Result>() {
    @Override
    public void onSuccess(AppendObjectRequest request,
AppendObjectResult result) {
        Log.d("AppendObject", "AppendSuccess");
        Log.d("NextPosition", "" + result.getNextPosition
());
    }

    @Override
    public void onFailure(AppendObjectRequest request
, ClientException clientException, ServiceException
serviceException) {
        // Handle the exceptions
    }
});
```

Set the position parameter properly as follows when uploading a file in the appended upload method.

- Set the value of `position` to 0 when you create a new appendable object by performing the `AppendObject` operation.
- When you append the uploaded data to an appendable object, set the value of `position` to the current size of the object.

You can get the size of an object in the following two methods:

- Get the object size from the response to the upload request.
- Get the object size by initiating a `HeadObject` request.

9.6.4 Resumable upload

This topic describes how to perform resumable uploads.

In a wireless network, it takes a long time to upload a large file. The upload may fail due to poor network connection or network switching. In this case, the entire file needs to be uploaded again. OSS Android SDK provides the resumable upload feature to address this problem.



Notice:

- Currently, resumable upload is applicable only to local files.
- We recommend you do not use resumable upload when uploading small-sized and medium-sized files from a mobile device. Resumable upload adopts the multipart upload method, which initiates multiple network requests for uploading a single file, reducing the upload efficiency.

Before uploading a file to OSS using the resumable upload method, you can specify the folder used to store resumable upload records.

- The resumable upload method is applied only in the current upload task. In scenarios where the folder used to store resumable upload records is not specified, if a part of a large file fails to be uploaded due to network problems, it takes a large amount of time and traffic to upload the entire file again.
- If you specify the folder used to store resumable upload records, a failed resumable upload task can be restarted from the position recorded.

When a resumable upload task fails and is not resumed for a long time, the uploaded parts become useless and occupy the storage space of OSS. In this case, you can set lifecycle rules for the bucket to clear useless parts in a timely manner. For more information, see [Lifecycle Management](#).

**Note:**

- Resumable upload is performed based on the following APIs: `InitMultipartUpload`, `UploadPart`, `ListParts`, `CompleteMultipartUpload`, and `AbortMultipartUpload`. If you use the STS authentication mode to perform resumable upload tasks, make sure that you can access the preceding APIs.
- Resumable upload supports the upload callback function, which is used in the same way as it is used in normal upload tasks.
- The MD5 value of each part is verified in a resumable upload task by default. Therefore, you do not have to specify the `Content - Md5` header in the request.
- The following code uses a calling method in which resumable upload records are not permanently stored on the local device:

```
// Create a resumable upload request .
ResumableU ploadReque st request = new ResumableU
ploadReque st ("< bucketName >", "< objectKey >", "< uploadFile
Path >");
// Configure the progress callback function for the
upload task .
request . setProgres sCallback ( new OSSProgres sCallback <
ResumableU ploadReque st >() {
@ Override
public void onProgress ( ResumableU ploadReque st request
, long currentSiz e , long totalSize ) {
Log . d ( " resumableU pload ", " currentSiz e : " +
currentSiz e + " totalSize : " + totalSize );
}
});
// Call the asynchrono usly resumable upload interface .
OSSAsyncTa sk resumableT ask = oss . asyncResum ableUpload
( request , new OSSComple t edCallback < ResumableU ploadReque
st , ResumableU ploadResul t >() {
@ Override
public void onSuccess ( ResumableU ploadReque st request
, ResumableU ploadResul t result ) {
Log . d ( " resumableU pload ", " success !" ) ;
}

@ Override
public void onFailure ( ResumableU ploadReque st request
, ClientExce ption clientExce pion , ServiceExc eption
serviceExc eption ) {
// Handle exceptions .
}
});
```

```
// resumableTask.waitForFinished(); // Wait until the task is complete.
```

- The following code uses a calling method in which resumable upload records are permanently stored on the local device:

```
String recordDirectory = Environment.getExternalStorageDirectory().getAbsolutePath() + "/oss_record/";

File recordDir = new File(recordDirectory);

// The directory must exist. Otherwise, it will be automatically created.
if (!recordDir.exists()) {
    recordDir.mkdirs();
}

// Create a resumable upload request with a parameter indicating the absolute path of the folder for storing resumable upload records.
ResumableUploadRequest request = new ResumableUploadRequest("<bucketName>", "<objectKey>", "<uploadFilePath>", recordDirectory);
// Set the progress callback function for the upload task..
request.setProgressCallback(new OSSProgressCallback<ResumableUploadRequest>() {
    @Override
    public void onProgress(ResumableUploadRequest request, long currentSize, long totalSize) {
        Log.d("resumableUpload", "currentSize: " + currentSize + " totalSize: " + totalSize);
    }
});

OSSAsyncTask resumableTask = oss.asyncResumableUpload(request, new OSSCompletedCallback<ResumableUploadRequest, ResumableUploadResult>() {
    @Override
    public void onSuccess(ResumableUploadRequest request, ResumableUploadResult result) {
        Log.d("resumableUpload", "success!");
    }

    @Override
    public void onFailure(ResumableUploadRequest request, ClientException clientException, ServiceException serviceException) {
        // Handle exceptions.
    }
});

// resumableTask.waitForFinished();
```

- The following code implements a resumable upload task:

```
// Specify whether to delete the settings of resumable upload records when the OSSAsyncTask.cancel() method is called
String recordDirectory = Environment.getExternalStorageDirectory().getAbsolutePath() + "/oss_record/";
```

```

File recordDir = new File ( recordDire ctory );

// The directory must exist . Otherwise , it will be
// automatica lly created .
if (! recordDir . exists () ) {
    recordDir . mkdirs ();
}

// Create a resumable upload request with a parameter
// indicating the absolute path of the folder for
// storing resumable upload records
ResumableU ploadReque st request = new ResumableU
ploadReque st ("< bucketName >", "< objectKey >", "< uploadFile
Path >", recordDire ctory );
// If it is set to false , the resumable upload
// records are not deleted when the upload task is
// canceled . If you do not set it , the default
// value is true , indicating that the resumable upload
// records are deleted after the task is canceled
// and the files are uploaded again during the next
// upload .
request . setDeleteU ploadOnCan celling ( false );
// Set the progress callback function for the upload
// task .
request . setProgres sCallback ( new OSSProgres sCallback <
ResumableU ploadReque st >() {
    @ Override
    public void onProgress ( ResumableU ploadReque st request
    , long currentSiz e , long totalSize ) {
        Log . d (" resumableU pload ", " currentSiz e : " +
        currentSiz e + " totalSize : " + totalSize );
    }
});

OSSAsyncTa sk resumableT ask = oss . asyncResum ableUpload
( request , new OSSComple t edCallback < ResumableU ploadReque
st , ResumableU ploadResul t >() {
    @ Override
    public void onSuccess ( ResumableU ploadReque st request
    , ResumableU ploadResul t result ) {
        Log . d (" resumableU pload ", " success !") ;
    }

    @ Override
    public void onFailure ( ResumableU ploadReque st request
    , ClientExce ption clientExce ption , ServiceExc eption
    serviceExc eption ) {
        // Handle exceptions .
    }
});

```



```
// resumableTask . waitUntilFinished ();
```

9.6.5 Progress bar

This topic describes how to use the progress bar to indicate the upload progress.

The following code shows how to call the `asyncPutObject` method to use the progress bar.

```
// Construct an upload request .
PutObjectRequest put = new PutObjectRequest ("< bucketName >", "< objectName >", "< uploadFile Path >");

// Configure the callback function to display the progress bar .
put . setProgressCallback ( new OSSProgressCallback < PutObjectRequest >() {
    @Override
    public void onProgress ( PutObjectRequest request , long
        currentSize , long totalSize ) {
        Log . d (" PutObject ", " currentSize : " + currentSize
        + " totalSize : " + totalSize );
    }
});

// Upload the file asynchronously .
OSSAsyncTask task = oss . asyncPutObject ( put , new
OSSCompletedCallback < PutObjectRequest , PutObjectResult >()
{
    @Override
    public void onSuccess ( PutObjectRequest request ,
PutObjectResult result ) {
        Log . d (" PutObject ", " UploadSuccess ");
        Log . d (" ETag ", result . getETag ());
        Log . d (" RequestId ", result . getRequest Id ());
    }

    @Override
    public void onFailure ( PutObjectRequest request
, ClientException clientException , ServiceException
serviceException ) {
        // A request exception occurs .
        if ( clientException != null ) {
            // A local exception ( such as network
exception ) occurs .
            clientException . printStackTrace ();
        }
        if ( serviceException != null ) {
            // A service exception occurs .
            Log . e (" ErrorCode ", serviceException . getErrorCo
de ());
            Log . e (" RequestId ", serviceException . getRequest
Id ());
            Log . e (" HostId ", serviceException . getHostId ());
            Log . e (" RawMessage ", serviceException .
getRawMessage ());
        }
    }
});
// task . cancel (); // You can cancel the upload task .
```

```
// task . waitUntilF inished (); // Wait until the upload
task is complete .
```

9.6.6 Upload callback

This topic describes how to use the upload callback function.

After a file is uploaded to OSS, OSS can return the callback information to the application server. To get the callback information, you can add callback-related parameters in the upload request. Note that it takes more time for the client to wait for the response than in simple upload tasks if the callback function is enabled because OSS takes time to respond the callback request.

The following code enables the upload callback function:

```
PutObjectR equest put = new PutObjectR equest ( testBucket ,
testObject , uploadFile Path );

put . setCallbac kParam ( new HashMap < String , String >() {
{
    put ( " callbackUr l " , " 110 . 75 . 82 . 106 / callback " );
    put ( " callbackHo st " , " oss - cn - hangzhou . aliyuncs .
com " );
    put ( " callbackBo dyType " , " applicatio n / json " );
    put ( " callbackBo dy " , "{ \" mimeType \" : ${ mimeType } , \" size \" :
${ size } }" );
}
} });

OSSAsyncTa sk task = oss . asyncPutOb ject ( put , new
OSSComple tedCallback < PutObjectR equest , PutObjectR esult >()
{
@ Override
    public void onSuccess ( PutObjectR equest request ,
PutObjectR esult result ) {
        Log . d ( " PutObject " , " UploadSucc ess " );

        // This parameter has value only when the servercall
back parameter is set .
        String serverCall backReturn Json = result . getServerC
allbackRet urnBody ();

        Log . d ( " servercall back " , serverCall backReturn Json );
    }

@ Override
    public void onFailure ( PutObjectR equest request ,
ClientExce ption clientExce ption , ServiceExc eption
serviceExc eption ) {
        // Handle exceptions .
    }
} });
```

If you want to configure custom parameters in the callback information, see the following code:

```
put . setCallbac kParam ( new HashMap < String , String >() {
```

```

        {
            put (" callbackUrl ", " http :// 182 . 92 . 192 . 125 /
leibin / notify . php ");
            put (" callbackHost ", " oss - cn - hangzhou . aliyuncs .
com ");
            put (" callbackBodyType ", " applicatio n / json ");
            put (" callbackBody ", "{ \" object \":${ object }, \" size
\":${ size }, \" my_var1 \":${ x : var1 }, \" my_var2 \":${ x : var2
}}");
        }
    });

    put . setCallbackVars ( new    HashMap < String ,    String >() {
        {
            put (" x : var1 ", " value1 ");
            put (" x : var2 ", " value2 ");
        }
    });

```

For more information about the callback function, see [Callback](#).

9.7 Download objects

9.7.1 Overview

OSS Android SDK provides various methods for you to download objects, including:

- [#unique_594](#)
- [#unique_595](#)
- [#unique_596](#)

9.7.2 Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use the streaming download method. Streaming download enables you to download a part of an object each time until you download the entire object.

You can obtain the input stream of an object if you download the object in the streaming download method. You must have the write permission on the object to perform streaming download.

Run the following code to perform a streaming download task synchronously:

```

// Construct an object download request .
GetObjectRequest get = new GetObjectRequest ("< bucketName
>", "< objectKey >");

// Configure the download progress callback function .
get . setProgressListener ( new    OSSProgressCallback <
GetObjectRequest >() {
    @ Override

```

```

        public void onProgress (GetObjectRequest request
, long currentSize, long totalSize) {
            OSSLog.logDebug ("getobj_progress : " +
currentSize + " total_size : " + totalSize, false);
        }
    });

    try {
        // Execute the download request synchronously and
        return the result.
        GetObjectResult getResult = oss . getObject (get);

        Log.d ("Content - Length", "" + getResult . getContentLength
());

        // Obtain the input stream of the object.
        InputStream inputStream = getResult . getObjectContent ();

        byte[] buffer = new byte [ 2048 ];
        int len;

        while ((len = inputStream . read (buffer)) != - 1) {
            // Process the downloaded data. For example, display
            images or write data to files.
        }

        // You can view the metadata of the download object.
        ObjectMetadata metadata = getResult . getMetadata ();
        Log.d ("ContentType", metadata . getContentType());

    } catch (ClientException e) {
        // Catch local exceptions, such as network exceptions.
        e.printStackTrace();
    } catch (ServiceException e) {
        // Catch service exceptions.
        Log.e ("RequestId", e.getRequestId());
        Log.e ("ErrorCode", e.getErrorCode());
        Log.e ("HostId", e.getHostId());
        Log.e ("RawMessage", e.getRawMessage());
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Run the following code to perform a streaming download tasks asynchronously:

```

GetObjectRequest get = new GetObjectRequest ("< bucketName
>", "< objectKey >");
// Configure the download progress callback function.
get.setProgressListener (new OSSProgressCallback <
GetObjectRequest >() {
    @Override
    public void onProgress (GetObjectRequest request
, long currentSize, long totalSize) {
        OSSLog.logDebug ("getobj_progress : " +
currentSize + " total_size : " + totalSize, false);
    }
});
OSSAsyncTask task = oss . asyncGetObject (get, new
OSSCompletedCallback < GetObjectRequest, GetObjectResult >()
{
    @Override

```

```

public void onSuccess ( GetObjectR equest request ,
GetObjectR esult result ) {
    // The request is successful .
    InputStrea m inputStream = result . getObjectC ontent ();

    byte [] buffer = new byte [ 2048 ];
    int len ;

    try {
        while (( len = inputStream . read ( buffer )) != - 1 ) {
            // Process the downloaded data .
        }
    } catch ( IOExceptio n e ) {
        e . printStack Trace ();
    }
}

@Override
public void onFailure ( GetObjectR equest request ,
ClientExce ption clientExce pion , ServiceExc eption
serviceExc eption ) {
    // Handle the exceptions returned for the request .
    if ( clientExce pion != null ) {
        // A local exception ( such as network exception )
        occurs .
        clientExce pion . printStack Trace ();
    }
    if ( serviceExc eption != null ) {
        // A service exception occurs .
        Log . e ( " ErrorCode " , serviceExc eption . getErrorCo de ());
        Log . e ( " RequestId " , serviceExc eption . getRequest Id ());
        Log . e ( " HostId " , serviceExc eption . getHostId ());
        Log . e ( " RawMessage " , serviceExc eption . getRawMess age
());
    }
}
});

```

9.7.3 Range download

If you only need a part of the data in an object, you can use the range download method to download specified content.

Run the following code to perform a range download task:

```

GetObjectR equest get = new GetObjectR equest ("< bucketName
>", "< objectKey >");

// Set the download range .
get . setRange ( new Range ( 0 , 99 )); // Download data
between the starting byte value of 0 and the
maximum byte value 99 inclusive ( that is , a total
range of 100 bytes ).
// get . setRange ( new Range ( 100 , Range . INFINITE )); //
Download data between the starting byte value of 100
and the end of the object .
OSSAsyncTa sk task = oss . asyncGetOb ject ( get , new
OSSComple tedCallback < GetObjectR equest , GetObjectR esult >()
{
    @ Override

```

```

public void onSuccess (GetObjectRequest request ,
GetObjectResult result) {
    // The request is successful .
    InputStream inputStream = result . getObjectContent ();

    byte [] buffer = new byte [ 2048 ];
    int len ;

    try {
        while (( len = inputStream . read ( buffer )) != - 1 ) {
            // Process the downloaded data .
        }
    } catch ( IOException e ) {
        e . printStackTrace ();
    }
}

@Override
public void onFailure (GetObjectRequest request ,
ClientException clientException , ServiceException
serviceException) {
    // Handle the exceptions returned for the request .
    if ( clientException != null ) {
        // A local exception ( such as network exception )
        occurs .
        clientException . printStackTrace ();
    }
    if ( serviceException != null ) {
        // A service exception occurs .
        Log . e ( " ErrorCode " , serviceException . getErrorCode () );
        Log . e ( " RequestId " , serviceException . getRequestId () );
        Log . e ( " HostId " , serviceException . getHostId () );
        Log . e ( " RawMessage " , serviceException . getRawMessage () );
    }
}
});

```

9.7.4 Download an object to a local file

This topic describes how to download an object to a local file.

Run the following code to download a specified object to a local file:

```

public void testDownloadObjectToFile () throws Exception
{
    // Upload a file .
    PutObjectRequest put = new PutObjectRequest (
mBucketName , " file1m " ,
OSSTestConfig . FILE_DIR + " file1m ");
    OSSTestConfig . TestPutCallback putCallback = new
OSSTestConfig . TestPutCallback ();
    String srcFileBase64Md5 = BinaryUtil . toBase64String (
BinaryUtil . calculateMd5 ( OSSTestConfig . FILE_DIR + "
file1m " ));
    OSSAsyncTask putTask = oss . asyncPutObject ( put ,
putCallback );
    putTask . waitUntilFinished ();
    assertEquals ( 200 , putCallback . result . getStatusC
ode ());

    // Download an object to a local file .
}

```

```

        GetObjectRequest get = new GetObjectRequest (
            mBucketName, "file1m");
        OSSTestConfig.TestGetCallback getCallback = new
        OSSTestConfig.TestGetCallback();
        OSSAsyncTask getTask = oss.asyncGetObject ( get ,
            getCallback );
        getTask.waitForFinished();
        assertEquals ( 200 , getCallback.result.getStatusCode () );
        long length = getCallback.result.getContentLength ();
        byte [] buffer = new byte [( int ) length ];
        int readCount = 0 ;
        while ( readCount < length ) {
            readCount += getCallback.result.getObjectContent (). read ( buffer , readCount , ( int ) length - readCount );
        }
        try {
            FileOutputStream fout = new FileOutputStream (
                OSSTestConfig.FILE_DIR + "download_file1m");
            fout.write ( buffer );
            fout.close ();
        } catch ( Exception e ) {
            OSSLog.logInfo ( e.toString () );
        }
        String downloadFileBase64Md5 = BinaryUtil.toBase64String ( BinaryUtil.calculateMd5 ( OSSTestConfig.FILE_DIR + "download_file1m" ));
        assertEquals ( srcFileBase64Md5 , downloadFileBase64Md5 );
    }
}

```

9.7.5 Conditional download

This topic describes how to download objects by setting conditions.

You can specify one or more conditions when downloading objects to determine that the object can be downloaded if the conditions are met, and an error code is returned and the object cannot be downloaded if the conditions are not met. The following table describes the conditions that you can specify.

Parameter	Description
If - Modified - Since	Indicates that if the specified time is earlier than the time when the object is modified, the object can be downloaded . Otherwise, a 304 Not Modified error is returned.

Parameter	Description
If - Unmodified - Since	Indicates that if the specified time is later than or equal to the time when the object is modified, the object can be downloaded. Otherwise, a 412 Precondition Failed error is returned.
If - Match	Indicates that if the specified ETag matches that of an object, the object can be downloaded. Otherwise, a 412 Precondition Failed error is returned.
If - None - Match	Indicates that if the specified ETag does not match that of an object, the object can be downloaded. Otherwise, a 304 Not Modified error is returned.

If - Modified - Since and If - Unmodified - Since can be configured as download conditions at the same time. If - Match and If - None - Match can be configured as download conditions at the same time.

You can use `ossClient.getObjectMeta` to obtain the ETag of an object.

Run the following code to download an object with conditions:

```
public void testGetObjectWithHeaders () throws Exception
{
    GetObjectRequest request = new GetObjectRequest (
        bucketName , objectName );
    request.setRequestHeaders (...);

    OSSAsyncTask task = oss . asyncGetObject ( request ,
        getCallback );
    task . waitUntilFinished ();

    GetObjectRequest rq = getCallback . request ;
    GetObjectResult result = getCallback . result ;
}
```

9.7.6 Resumable download

In a resumable download, if the download process is interrupted, it can be restarted from where it stopped, which saves time and traffic.

For example, when you download a video using an application on your mobile phone , the download process is interrupted by default if the network mode of the mobile phone switches from Wi-Fi networks to the cellular data network. When the network

mode switches back to Wi-Fi networks, you can manually restart the download process from where it stopped, that is, perform a resumable download.

Detail analysis

The If-Range HTTP request header field specifies the condition that needs to be met to make the Range header field functional.

- The Range header field takes effect only when the condition included in the If-Range header field is met. In this case, the OSS server returns the 206 Partial Content code and the content specified by the Range header field.
- If the condition included in the If-Range header field is not met, the OSS server returns a 200 OK status code and the complete resources that are requested.
- Both the Last-Modified and ETag values can be used for verification. However, the two values cannot be used at the same time.

The If-Range header field is used in resumable downloads to verify whether the resources downloaded in the last paused download are changed.

When the `If - Range` header field is used, the client only needs to initiate one request. However, when `If - Unmodified - Since` or `If - Match` is used, if the specified condition is not met, a 412 pre-condition check failure status code is returned, and the client must initiate another request to obtain the resources.

Implementation

For more information about how to perform resumable downloads on OSS objects in Android systems, see the following implementation code in iOS. If you want to perform resumable downloads, you can implement the method by yourself or download open source third-party frameworks.

You can run the following code to perform resumable downloads on OSS objects in iOS.

```
// 1 . Use SDK to get the download URL of the
object .
String signedURLs string = ossClient . presignCon strainedOb
jectURL ( bucket , object , expires );

// 2 . Add a download task .

mDownloadM anager = DownloadMa nager . getInstanc e ();
mDownloadM anager . add ( signedURLs string , new
DownloadLi stner () {
    @ Override
    public void onFinish () {
```

```

        Toast . makeText ( MainActivity . this , " Download
complete !", Toast . LENGTH_SHORT ). show ();
    }

    @ Override
    public void onProgress ( float progress ) {
        pb_progress . setProgress ( ( int ) ( progress
* 100 ));
        tv_progress . setText ( String . format ( "%. 2f
", progress * 100 ) + "%");
    }

    @ Override
    public void onPause () {
        Toast . makeText ( MainActivity . this , " Download
paused !", Toast . LENGTH_SHORT ). show ();
    }

    @ Override
    public void onCancel () {
        tv_progress . setText ( " 0 %");
        pb_progress . setProgress ( 0 );
        btn_download . setText ( " Download ");
        Toast . makeText ( MainActivity . this , " Download
canceled !", Toast . LENGTH_SHORT ). show ();
    }
});

// 3 . Start the download .
mDownloadManager . download ( signedURLs );

// 4 . Pause the download .
mDownloadManager . cancel ( signedURLs );

// 5 . Continue to download .
mDownloadManager . download ( signedURLs );

```

**Note:**

Currently, OSS servers do not support the `If - Range` field. Therefore, you cannot use `NSURLSessionDownloadTask` to perform resumable downloads in Android systems. When `NSURLSessionDownloadTask` is used to perform resumable downloads, the `If - Range` header field is sent to the OSS server to verify whether the downloaded object is changed.

9.8 Manage objects

9.8.1 Overview

You can manage objects in a bucket by calling APIs to perform the following operations:

- [Determine whether an object exists](#)

- [Obtain Object Meta](#)
- [List objects](#)
- [Copy objects](#)
- [Delete objects](#)

9.8.2 Determine whether a specified object exists

OSS provides a synchronous interface for you to check whether a specified object exists in a bucket.

Run the following code to determine whether a specified object exists:

```
try {
    if ( oss . doesObject Exist ("< bucketName >", "< objectKey >")) {
        Log . d (" doesObject Exist ", " object exist .");
    } else {
        Log . d (" doesObject Exist ", " object does not exist .");
    }
} catch ( ClientExce ption e ) {
    // Catch local exceptions , such as network exceptions
    e . printStack Trace ();
} catch ( ServiceExc eption e ) {
    // Catch service exceptions .
    Log . e (" ErrorCode ", e . getErrorCo de ());
    Log . e (" RequestId ", e . getRequest Id ());
    Log . e (" HostId ", e . getHostId ());
    Log . e (" RawMessage ", e . getRawMess age ());
}
```

9.8.3 Copy an object

You can copy an object from a bucket (source bucket) to another bucket (target bucket) in the same region.

Run the following code to copy an object.

```
// Construct a copy request .
CopyObject Request copyObject Request = new CopyObject
Request ("< srcBucketN ame >", "< srcObjectK ey >",
"< destBucket Name >", "< destObject Key >");

// ( Optional ) Configure the metadata of the object to
be copied .
// ObjectMeta data objectMeta data = new ObjectMeta data ();
// objectMeta data . setContent Type (" applicatio n / octet -
stream ");
// copyObject Request . setNewObje ctMetadata ( objectMeta data );

// Copy the object by calling the asynchrono us
interface .
OSSAsyncTa sk copyTask = oss . asyncCopyO bject ( copyObject
Request , new OSSComple tedCallback < CopyObject Request ,
CopyObject Result >() {
```

```

    @ Override
    public void onSuccess ( CopyObject Request request ,
CopyObject Result result ) {
        Log . d ( " copyObject ", " copy    success !");
    }

    @ Override
    public void onFailure ( CopyObject Request request
, ClientExce ption clientExce ption , ServiceExc eption
serviceExc eption ) {
        // Handle the exceptions returned for the request
        .
        if ( clientExce ption != null ) {
            // A local exception ( such as network
exception ) occurs .
            clientExce ption . printStack Trace ();
        }
        if ( serviceExc eption != null ) {
            // A service exception occurs .
            Log . e ( " ErrorCode ", serviceExc eption . getErrorCo
de ());
            Log . e ( " RequestId ", serviceExc eption . getRequest
Id ());
            Log . e ( " HostId ", serviceExc eption . getHostId ());
            Log . e ( " RawMessage ", serviceExc eption .
getRawMess age ());
        }
    }
});

```

**Note:**

- The source object and the target object must be in the same region.
- If the path of the source object is the same as that of the target bucket, you can modify the metadata of the existing target bucket.
- You can only copy objects that is smaller or equal to 1 GB. For objects larger than 1 GB, you must copy them by performing other operations, such as MultipartUpload.

9.8.4 List objects

This topic describes how to list all objects or objects with a specified prefix in a bucket.

List all objects in a bucket

Run the following code to list all objects in a bucket:

```

ListObject sRequest listObject s = new ListObject sRequest
("< bucketName >");

// Configure the callback informatio n returned when
the request is successful and fails . Send the list
request by calling the asynchrono us interface .

```

```

OSSAsyncTask task = oss.asyncListObjects(listObjects,
new OSSCompletedCallback<ListObjectsRequest, ListObjectsResult>() {
    @Override
    public void onSuccess(ListObjectsRequest request,
ListObjectsResult result) {
        Log.d("AsyncListObjects", "Success!");
        for (int i = 0; i < result.getObjectSummaries().
size(); i++) {
            Log.d("AsyncListObjects", "object: " + result.
getObjectSummaries().get(i).getKey() + " "
+ result.getObjectSummaries().get(i).getETag() + " "
+ result.getObjectSummaries().get(i).getLastModified
());
        }
    }
});

@Override
public void onFailure(ListObjectsRequest request,
ClientException clientException, ServiceException
serviceException) {
    // Handle the exceptions returned for the request.
    if (clientException != null) {
        // A local exception (such as network exception)
        occurs.
        clientException.printStackTrace();
    }
    if (serviceException != null) {
        // A service exception occurs.
        Log.e("ErrorCode", serviceException.getErrorCode());
        Log.e("RequestId", serviceException.getRequestId());
        Log.e("HostId", serviceException.getHostId());
        Log.e("RawMessage", serviceException.getRawMessage
());
    }
}
});
// Wait until the task is complete.
task.waitForFinished();

```

List objects with a specified prefix in a bucket

Run the following code to list all object prefixed with "file" in a bucket:

```

ListObjectsRequest listObjects = new ListObjectsRequest
("< bucketName >");

// Set the prefix.
listObjects.setPrefix("file");

// Configure the callback information returned when
the request is successful and fails. Send the list
request by calling the asynchronous interface.
OSSAsyncTask task = oss.asyncListObjects(listObjects,
new OSSCompletedCallback<ListObjectsRequest, ListObjectsResult>() {
    @Override
    public void onSuccess(ListObjectsRequest request,
ListObjectsResult result) {
        Log.d("AsyncListObjects", "Success!");
        for (int i = 0; i < result.getObjectSummaries().
size(); i++) {

```

```

        Log . d ( " AyncListOb jects ", " object : " + result .
getObjectS ummaries (). get ( i ). getKey () + " "
        + result . getObjectS ummaries (). get ( i ). getETag () + " "
        + result . getObjectS ummaries (). get ( i ). getLastMod ified
        ());
    }
}

@ Override
public void onFailure ( ListObject sRequest request ,
ClientExce ption clientExce ption , ServiceExc eption
serviceExc eption ) {
    // Handle the exceptions returned for the request .
    if ( clientExce ption != null ) {
        // A local exception ( such as network exception )
        occurs .
        clientExce ption . printStack Trace ();
    }
    if ( serviceExc eption != null ) {
        // A service exception occurs .
        Log . e ( " ErrorCode ", serviceExc eption . getErrorCo de ());
        Log . e ( " RequestId ", serviceExc eption . getRequest Id ());
        Log . e ( " HostId ", serviceExc eption . getHostId ());
        Log . e ( " RawMessage ", serviceExc eption . getRawMess age
        ());
    }
}
});
task . waitUntilF inished ();

```

The following table describes the parameters in the preceding code.

Parameter	Description
delimiter	Specifies the character used to group object names. All objects whose name includes the specified prefix and the delimiter character for the first time are classified into a group of elements: CommonPrefixes.
marker	Indicates that objects with names after the marker in alphabetical order are returned.
maxkeys	Specifies the maximum number of the returned objects for the current request. The value of this parameter cannot be larger than 1,000. Default value: 100

Parameter	Description
prefix	Indicates that the keys of the returned objects must be prefixed with the value of this parameter.  Note: If you include the prefix in the query request, it is also included in the returned key.

9.8.5 Delete an object

This topic describes how to delete an object.



Warning:

Delete an object with caution because a deleted object cannot be recovered.

Run the following code to delete an object:

```
// Create a delete request
DeleteObjectRequest delete = new DeleteObjectRequest("<
bucketName >", "< objectKey >");
// Delete an object by calling the asynchronous
interface .
OSSAsyncTask deleteTask = oss.asyncDeleteObject(delete,
new OSSCompletedCallback<DeleteObjectRequest, DeleteObject
ctResult>() {
    @Override
    public void onSuccess (DeleteObjectRequest request,
DeleteObjectResult result) {
        Log.d("asyncCopyAndDelete", "success!");
    }

    @Override
    public void onFailure (DeleteObjectRequest request
, ClientException clientException, ServiceException
serviceException) {
        // Handle the exceptions returned for the request
        .
        if (clientException != null) {
            // A local exception (such as network
exception) occurs.
            clientException.printStackTrace();
        }
        if (serviceException != null) {
            // A service exception occurs.
            Log.e("ErrorCode", serviceException.getErrorCode());
            Log.e("RequestId", serviceException.getRequest
Id());
            Log.e("HostId", serviceException.getHostId());
            Log.e("RawMessage", serviceException.getRawMess
age());
        }
    }
}
```

```
}  
});
```



Notice:

To call `DeleteObject` to delete an object, you must have the write permission on the bucket that stores the object.

9.8.6 Set the Content-Type of an object

In Web services, the Content-Type of an object specifies the type of the object and determines the method and encoding used to read the object.

- In some cases, you must specify the Content-Type of an object to be uploaded. Otherwise, the object cannot be read in the appropriate method. If you do not specify the Content-Type when using OSS Android SDK to upload an object, the SDK automatically specifies it according to the suffix of the object.

```
// Construct an upload request .  
PutObjectRequest put = new PutObjectRequest ("<  
bucketName >", "< objectKey >", "< uploadFile Path >");  
  
ObjectMeta metadata = new ObjectMeta ();  
// Set the Content - Type of the object to be  
uploaded .  
metadata . setContent Type (" applicatio n / octet - stream ");  
// Customize the user metadata .  
metadata . addUserMetadata (" x - oss - meta - name1 ", " value1  
");  
put . setMetadata ( metadata );  
  
OSSAsyncTask task = oss . asyncPutObject ( put , new  
OSSCompletedCallback < PutObjectRequest , PutObjectResult  
>() {  
    ...  
});
```

9.8.7 Obtain the metadata of an object

You can call the `HeadObject` method to obtain only the metadata but not the content of an object.

Run the following code to obtain the metadata of an object:

```
// Create a request to obtain the metadata of an  
object by calling the synchronous interface .  
HeadObjectRequest head = new HeadObjectRequest ("<  
bucketName >", "< objectKey >");  
  
// Obtain the metadata of an object .  
OSSAsyncTask task = oss . asyncHeadObject ( head , new  
OSSCompletedCallback < HeadObjectRequest , HeadObjectResult  
>() {  
    @Override
```



```

    public void onSuccess ( HeadObject Request request ,
        HeadObject Result result ) {

        // Obtain the size of the object .
        Log . d ( " headObject ", " object Size : " + result .
            getMetadat a (). getContent Length ());
        // Obtain the type of the object .
        Log . d ( " headObject ", " object Content Type : " +
            result . getMetadat a (). getContent Type ());
    }

    @ Override
    public void onFailure ( HeadObject Request request
        , ClientExce ption clientExce pion , ServiceExc eption
        serviceExc eption ) {
        // Handle the exceptions returned for the request
        .
        if ( clientExce pion != null ) {
            // A local exception ( such as network
            exception ) occurs .
            clientExce pion . printStack Trace ();
        }
        if ( serviceExc eption != null ) {
            // A service exception occurs .
            Log . e ( " ErrorCode ", serviceExc eption . getErrorCo
            de ());
            Log . e ( " RequestId ", serviceExc eption . getRequest
            Id ());
            Log . e ( " HostId ", serviceExc eption . getHostId ());
            Log . e ( " RawMessage ", serviceExc eption .
            getRawMess age ());
        }
    }
});

// task . waitUntilF inished (); // Wait until the task is
    complete .

```

9.9 Data security

OSS Android SDK provides methods to verify data integrity, which ensure data security when you uploading, downloading, and copying objects.

Due to the complex network environment for mobile devices, errors may occur when data is transmitted between the client and the OSS server. Therefore, OSS Android SDK provides two methods to verify data integrity: CRC end-to-end verification and MD5 verification.

- CRC verification

If you enable CRC verification when reading the download stream, data integrity is automatically verified after the stream is completely read.

Run the following code to enable CRC verification:

```
GetObjectRequest request = new GetObjectRequest (
    OSSTestConfig.ANDROID_TEST_BUCKET, testFile );
// Enable CRC verification.
request.setCRC64 ( OSSRequest.CRC64Config.YES );
//....
try {
    GetObjectResult result = oss.getObject ( request );
    InputStream in = result.getObjectContent ();
    ByteArrayOutputStream output = new ByteArrayOutputStream
    outputStream ();
    byte [] buffer = new byte [ BUFFER_SIZE ];
    int len;
    while ( ( len = in.read ( buffer ) ) > - 1 ) {
        output.write ( buffer, 0, len );
    }
    output.flush ();
    in.close ();
} catch ( ClientException e ) {
    //...
} catch ( InconsistentException e ) {
    //....
}
```

- MD5 verification

To check whether the object uploaded in the multipart upload method is the same as the local file, you can upload the parts with their Content-MD5 values. The OSS server checks the MD5 values of the uploaded parts and determines the upload task is successful only when the MD5 values are the same as the specified Content-MD5 values, ensuring that the uploaded object is the same as the local file.

Run the following code to enable MD5 verification:

```
UploadPartRequest uploadPart = new UploadPartRequest ("<
bucketName >", "< objectKey >", uploadId, currentIndex );
// Configure the part content.
uploadPart.setPartContent ( partData );
// Configure the MD5 value.
```

```
uploadPart . setMd5Digest ( BinaryUtil . calculateBase64Md5 (
data ));
```

9.10 Sign a URL

SDK supports the signing of the URL with a validity period or a public resource URL so that the URL can be transferred to a third party for authorized access.

Sign a private resource URL with a validity period

For a bucket or an object not assigned the public-read permission, the following interface can be called to obtain a signed URL to access the bucket or object:

```
String url = oss . presignConstructObjectURL ("< bucketName
>", "< objectKey >", 30 * 60 );
```

Sign a public resource URL

For a bucket or an object with the public-read permission, the following interface can be called to obtain a public URL to access the bucket or object:

```
String url = oss . presignPublicObjectURL ("< bucketName >",
"< objectKey >");
```

9.11 Authorized access

OSS Android SDK provides two authentication modes to ensure data security of mobile devices: STS authentication mode and self-signed mode.

STS authentication mode

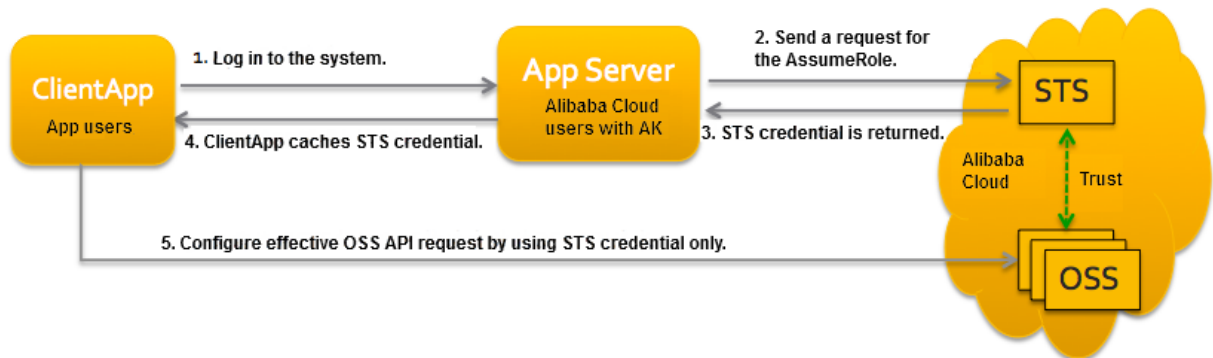
You can temporarily grant OSS access to a user by using the Alibaba Cloud Security Token Service (STS). Alibaba Cloud STS is a web service providing temporary access tokens for cloud computing users. With the STS, you can assign a third-party application or federated user (you can manage the user ID) an access credential with a custom validity period and permissions. For more information about STS, see [Introduction](#).

The STS authentication mode has the following advantages:

- You do not need to disclose your long-term key (AccessKey) to a third-party application. You only need to generate an access token and send the access token to the third-party application. You can customize the access permissions and the validity of this token.

- You do not need to revoke permissions. The access token becomes invalid automatically when it expires.

For example, the interaction process in an application is shown in the following figure:



The solution is as follows:

1. Log on to the application. You can manage your application ID. You can customize the ID management system or use an external web account or OpenID. For each valid app user, the AppServer can precisely define the minimum access permission .
2. The AppServer requests a security token (SecurityToken) from the STS. Before the STS is called, the AppServer needs to determine the minimum access permission (in the policy syntax) of the application user and the expiration time of the authorization. Then, you can obtain the security token by calling the STS AssumeRole API. For more information about the roles and usage, see [Role Management](#) in the RAM User Guide.
3. The STS returns a valid access credential (FederationToken in the application) to the AppServer. The access credential contains a SecurityToken, a temporary pair of access keys (an AccessKeyID and an AccessKeySecret), and the expiration time.
4. The AppServer returns the FederationToken to the ClientApp. The ClientApp can cache this credential. When the credential becomes invalid, the ClientApp needs to request a new valid access credential from the AppServer. For example, if the access credential is valid for one hour, the ClientApp can request the AppServer to update the access credential every 30 minutes.
5. The ClientApp uses the locally cached FederationToken to request Alibaba Cloud service APIs. The cloud service detects the STS access credential, verifies the credential with the STS service, and responds to the user requests accordingly.

For more information about the process of accessing OSS resources through STS, see [Access OSS with a temporary access token provided by STS](#) in the OSS Developer Guide.

For more information about STS security tokens, see [Role Management](#) in the RAM User Guide. You can call the [AssumeRole](#) API of STS to obtain the valid access credential. You can also directly use STS SDKs to call this method. For more information, see [#unique_621](#).

**Note:**

To use the STS authentication mode, you must first activate the Alibaba Cloud RAM service.

For more information about STS usage, see [STS user guide](#).

For more information about OSS authentication policy configuration, see [Overview](#).

- Set the StsToken directly

You can get a pair of the StsToken in the application by a certain method (such as network requests from your business server) and then use it to initialize the SDK. If this method is adopted, you must follow the expiration time of the STSToken. When the STSToken is about to expire, you must take the initiative to update the new STSToken for the SDK.

The sample code for initialization is as follows:

```
String endpoint = "http://oss-cn-hangzhou.aliyuncs.com";

OSSCredentialProvider credentialProvider = new
OSSStsTokenCredentialProvider("< StsToken.AccessKeyId >",
"< StsToken.SecretKeyId >", "< StsToken.SecurityToken >");
```

```
OSS oss = new OSSClient ( getApplicationContext (),
    endpoint , credential Provider );
```

When you notice that the token is about to expire, you can rebuild a new `OSSClient` or update the `CredentialProvider` by the following method:

```
oss . updateCredentialProvider ( new OSSStsTokenCredentialProvider ("< StsToken . AccessKeyId >", "< StsToken . SecretKeyId >", "< StsToken . SecurityToken >"));
```

- **Implement the callback to get the StsToken**

If you want the SDK to automatically update the Token, you must implement the callback in the SDK application, which gets a `FederationToken` (STSToken) and returns the token. The SDK uses the token for signing. When the token needs to be updated, the SDK calls the callback to get the token.

```
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com ";

OSSCredentialProvider credential Provider = new
OSSFederationCredentialProvider () {

    @ Override
    public OSSFederationToken getFederationToken () {
        // Here you must implement a method to get a
        FederationToken and construct it into the OSSFederationToken
        object to return it .
        // If the token is not obtained , " nil " is
        returned .

        OSSFederationToken * token ;
        // The following are some codes that can be
        used to obtain the token , for example from your
        server .
        ...
        return token ;
    }
};

OSS oss = new OSSClient ( getApplicationContext (),
    endpoint , credential Provider );
```



Note:

Additionally, if you have obtained all the fields required by the token by other means, you can use the callback to directly return the token. In this case, you

must manually update the token, and then reset the OSSCredentialProvider of the OSSClient instance.

Suppose the URL of the server you set up is `http://localhost:8080/distribute-token.json`. If you access the URL, the returned data is shown as follows:

```
{
  " StatusCode ": 200 ,
  " AccessKeyI d ":" STS . iA645eTOXE qP3cg3VeHf ",
  " AccessKeyS ecret ":" rV3VQrpFQ4 BsyHSAvi5N VLPiVffDJ
v4LojUBZCf ",
  " Expiration ":" 2015 - 11 - 03T09 : 52 : 59Z ",
  " SecurityTo ken ":" CAES7QIIAR KAAZPlqaN9 ILiQZPS + JDkS /
GSZN45RLx4 YS / p30gaUC + oJl3XSlbJ7 StKpQ ...."}

```

You can use the following codes to implement the `OSSFederat ionCredent ialProvide r` instance:

```
OSSCredent ialProvide r credetialP rovider = new
OSSFederat ionCredent ialProvide r () {
@ Override
public OSSFederat ionToken getFederat ionToken () {
try {
URL stsUrl = new URL (" http :// localhost : 8080 /
distribute - token . json ");
HttpURLCon nection conn = ( HttpURLConnection ) stsUrl .
openConnec tion ();
InputStrea m input = conn . getInputSt ream ();
String jsonText = IOUtils . readStream AsString ( input ,
OSSConstan ts . DEFAULT_CH ARSET_NAME );
JSONObject jsonObj = new JSONObject ( jsonText );
String ak = jsonObj . getString ( " AccessKeyI d " );
String sk = jsonObj . getString ( " AccessKeyS ecret " );
String token = jsonObj . getString ( " SecurityTo ken " );
String expiration = jsonObj . getString ( " Expiration " );
return new OSSFederat ionToken ( ak , sk , token ,
expiration );
} catch ( Exception e ) {
e . printStack Trace ();
}
return null ;
}
};

```

- Self-signed mode

The code is as follows:



Notice:

For the STS authentication mode and self-signed mode, make sure the results are returned when you call the implemented callback function. Therefore, if you must obtain the token and signature from the business server with a network request, we recommend that you call the synchronous APIs of the network library. The

callback is run in the child thread of the request from the SDKs, which does not block the main thread.

Self-signed mode

You can store the AccessKeyID and AccessKeySecret to your business server and then obtain them by implementing the callback in the SDK. Post the signature string to be signed to the server, sign the string using the OSS-specified signature algorithm on the business server, and transfer the signed string to the callback function, which returns the string to the server.

For more information about the signature algorithm, see [Add a signature to the header](#).

The content is a string merged according to various request parameters. The algorithm is:

```
signature = " OSS " + AccessKeyID + ":" + base64 ( hmac - sha1
( AccessKeySecret , content ) )
```

The code is as follows:

```
String endpoint = " http :// oss - cn - hangzhou . aliyuncs . com
";

credential Provider = new OSSCustomSignerCredentialProvider () {
    @ Override
    public String signContent ( String content ) {
        // Here you must sign a string using the OSS -
        specified signature algorithm , merge the AccessKeyID
        with the signed string , and return the string .
        // Generally , post the string to your business
        server and then return the signature .
        // If the signing fails , " nil " is returned after
        the error is described .

        // The following is a demo using a local
        algorithm for signing .
        return " OSS " + AccessKeyID + ":" + base64 ( hmac - sha1
        ( AccessKeySecret , content ) );
    }
};

OSS oss = new OSSClient ( getApplicationContext () ,
    endpoint , credential Provider );
```



Notice:

For the STS authentication mode and self-signed mode, make sure the results are returned when you call the implemented callback function. Therefore, if you must

obtain the token and signature from the business server with a network request, we recommend that you call the synchronous APIs of the network library. The callback is run in the child thread of the request from the SDKs, which does not block the main thread.

9.12 Image processing

Image Processing (IMG) is a massive, secure, cost-effective and highly reliable image processing service.

After source images are uploaded to OSS, you can process the images on any Internet device at any anytime, from anywhere through simple `RESTful` APIs. You can establish your own image processing services based on the image processing service provided by OSS.

Features

For more information about the features of OSS image processing service, see [OSS image processing](#).

[OSS image processing service](#) provides the following features:

- [Resize images](#)
- [Crop images](#)
- [Rotate images](#)
- [Apply effects](#)
- [Convert format](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.

Usage

- **Anonymous access**

```
String url = oss.presignPublicURL( testBucket ,
    testObject );
OSSLog.logDebug( "signPublic URL ", "get url : " + url );
```

Add the `x-oss-process:operation` parameter into the generated URL, in which `operation` indicates the operation performed on the image.

- **Authorized access**

When processing images by using OSS Android SDK, call the `request`.

`setxOssProcess()` method to configure image processing parameters, as shown in the following code:

```
// Construct the image download request .
GetObjectRequest get = new GetObjectRequest ("<
bucketName >", " example . jpg ");

// Process the image .
request . setxOssProcess (" image / resize , m_fixed , w_100 ,
h_100 ");

OSSAsyncTask task = oss . asyncGetObject ( get , new
OSSCompletedCallback < GetObjectRequest , GetObjectResult
>() {
    @Override
    public void onSuccess ( GetObjectRequest request ,
GetObjectResult result ) {
        // The request is successful .
        InputStream inputStream = result . getObjectContent ();

        byte [] buffer = new byte [ 2048 ];
        int len ;

        try {
            while (( len = inputStream . read ( buffer )) != -
1 ) {
                // Process the downloaded data .
            }
        } catch ( IOException e ) {
            e . printStackTrace ();
        }
    }

    @Override
    public void onFailure ( GetObjectRequest request
, ClientException clientException , ServiceException
serviceException ) {
        // Handle the exceptions . You can add your
        own code .
    }
});
```



Note:

To perform other operations on the image, replace the parameters of `request`.

`setxOssProcess()`.

- Access through SDK

```
GetObjectRequest request = new GetObjectRequest("bucket"
    - name, "image - name");
request.setxOssProcess("image / resize, m_lfit, w_100,
    h_100"); // Configure image processing parameters.
OSSAsyncTask task = ossClient.asyncGetObject(request,
    getCallback());
```

Save image processing result

Run the following code to save the image processing results:

```
OSSImagePersistRequest * request = [OSSImagePersistRequest
    new];
request.fromBucket = _privateBucketName;
request.fromObject = OSS_IMAGE_KEY;
request.toBucket = _privateBucketName;
request.toObject = @"image_persist_key";
// Scale the image.
request.action = @"image / resize, w_100";

[[[ossClient imageActionPersist:request] continueWithBlock
:^ id _Nullable (OSSTask * _Nonnull task) {
    return nil;
}] waitUntilFinished];
```

9.13 Exception handling

OSS Android SDK may be subject to two types of exceptions: `ClientException` and `ServiceException`, which are as follows:

ClientException

`ClientException` indicates an exception inside SDK, such as an incorrect parameter, an unreachable network, or an active cancellation.

ServiceException

`OSSEException` indicates a server error, which is from parsing a server error message.

`OSSEException` generally has the following components:

- **Code:** The error code OSS returns to users.
- **Message:** The detailed error message provided by OSS.
- **RequestId:** The UUID that uniquely identifies the request. When you cannot solve the problem, you can seek help from OSS development engineers by providing this RequestId.
- **HostId:** Marking the accessed OSS cluster.

- **rawMessage** The original body text of an HTTP response.

The following are common OSS exceptions:

Error code	Description
AccessDenied	Access denied
BucketAlreadyExists	The bucket already exists.
BucketNotEmpty	The bucket is not empty.
EntityTooLarge	The entity is too large.
EntityTooSmall	The entity is too small.
FileGroupTooLarge	The object group is too large.
FilePartNotExist	An object part does not exist.
FilePartStale	An object part has expired.
InvalidArgument	Parameter format error.
InvalidAccessKeyId	The AccessKeyID does not exist.
InvalidBucketName	The bucket name is invalid.
InvalidDigest	The digest is invalid.
InvalidObjectName	The object name is invalid.
InvalidPart	A part is invalid.
InvalidPartOrder	The part order is invalid.
InvalidTargetBucketForLogging	The logging operation has an invalid target bucket.
InternalError	Internal OSS error.
MalformedXML	Invalid XML format.
MethodNotAllowed	The method is not supported.
MissingArgument	A parameter is missing.
MissingContentLength	The content length is missing.
NoSuchBucket	The bucket does not exist.
NoSuchKey	The object does not exist.
NoSuchUpload	The Multipart Upload ID does not exist.
NotImplemented	The method cannot be processed.
PreconditionFailed	Pre-processing error.

Error code	Description
RequestTimeTooSkewed	The request initiation time exceeds the server time by 15 minutes.
RequestTimeout	Request timed out.
SignatureDoesNotMatch	Signature error.
TooManyBuckets	The user' s bucket quantity exceeds the limit.

10 iOS

10.1 Preface

This topic is based on OSS iOS SDK 8.0.

Overview

This topic assumes that you have activated OSS and created an AccessKeyID and AccessKeySecret. In OSS iOS documentation, ID indicates the AccessKeyID and KEY indicates the AccessKeySecret.

If you have not activated or do not know about OSS, log on to the [OSS Product Homepage](#) for more help.

Environment requirements

- iOS V8.0 and later
- You must have registered an Alibaba Cloud account and activated OSS.

SDK source code and API documents

For the source code of the SDK, see [GitHub](#). For more information, see [OSS iOS SDK](#).

Download the SDK

- [Download from GitHub](#)
- Pod dependency: pod 'AliyunOSSiOS'
- demo URL: <https://github.com/aliyun/aliyun-oss-ios-sdk>
-

10.2 Installation

This topic describes how to install the OSS iOS SDK.

Environment requirements

- iOS: 8.0 or later
- MacOS: 10.10 or later

Directly import the framework

For more information about how to obtain the OSS iOS SDK framework, see [GitHub](#).

In Xcode, drag the framework and drop it to your target area. In the dialog box that appears, select Copy items if needed.

Pod dependency

If your project uses Cocoapods to manage dependencies, add the following dependency to the Podfile. In this case, you do not need to import the OSS iOS SDK framework.

```
pod 'AliyunOSSiOS'
```

For more information about the Cocoapods management tool, see [CocoaPods installation and use tutorial](#).



Note:

You can choose either of the preceding methods.

Import the header file to your project

```
# import <AliyunOSSiOS/OSSService.h>
```



Notice:

After you import the framework, add `-ObjC` to Other Linker Flags of Build Settings in your project. If the `-force_load` option is configured for your project, add `-force_load <framework path>/AliyunOSSiOS`.

Compatible with IPv6-only networks

Domain name resolution in wireless networks is subject to hijackings. To address this issue, the OSS mobile SDK imports HTTPDNS for domain name resolution and directly uses IP addresses to request the OSS server. In an IPv6-only network, compatibility issues may occur. The App Store has officially issued the review requirements for applications, requiring applications to be IPv6-only network compatible. To meet this requirement, the SDK starts to be compatible from V2.5.0. In the later version, aside from `-ObjC` settings, two system libraries must be imported:

```
libresolv . tbd
CoreTelephony . framework
SystemConfiguration . framework
```

About the ATS policy of Apple

During WWDC 2016, Apple announced that all the applications in Apple App Store must implement App Transport Security (ATS) from January 1, 2017. In other words, all newly submitted applications are not allowed to use `NSAllowsArbitraryLoads` to bypass the ATS limitation. Additionally, the application submitters must ensure that all network requests from the application are encrypted through HTTPS. Otherwise, the application may fail the review.

The OSS iOS SDK V 2 . 6 . 0 and later provide the support for this ATS policy. The SDK sends only HTTPS requests. The SDK supports the prefix of `https ://` in the `Endpoint` field value. You need only to set the correct HTTPS `Endpoint` field. In this way, all network requests can meet the requirements.



Notice:

- When you set the `Endpoint` field, you must use the URL that is prefixed with `https ://`.
- Ensure that the application initiates only HTTPS requests when implementing callbacks such as signing and obtaining STS tokens.

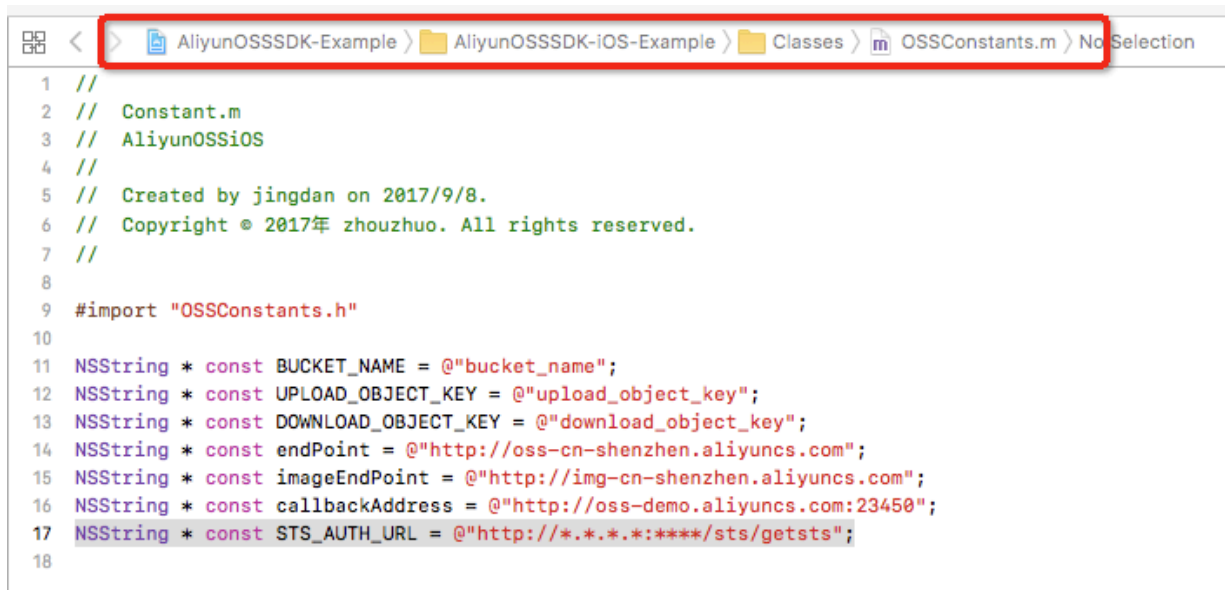
10.3 Quick start

This topic describes how to get started with the iOS SDK.

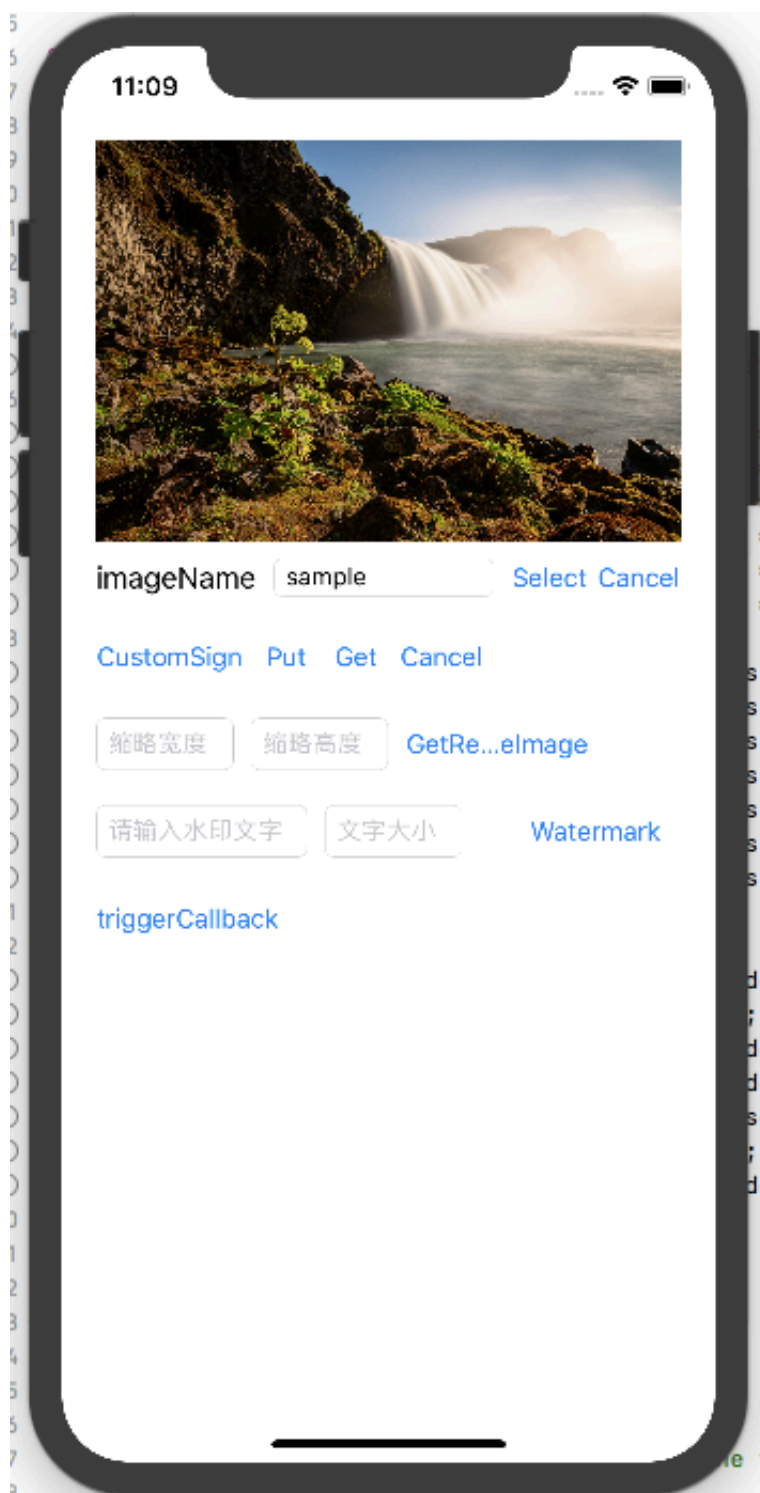
For more information, see the following examples and cases of this project:

- [iOS demo](#)
- [Mac demo](#)
- [Swift demo](#)
- [Test cases](#)

You can also directly run the git clone <https://github.com/aliyun/aliyun-oss-ios-sdk> command and configure the following parameters:



Run the project demo as follows:



Examples

The following code shows you how to upload and download files on the iOS and Mac platforms.

- iOS platform

1. Add the reference.

```
# import < AliyunOSSi OS / OSSService . h >
```

2. Initialize the OSSClient instance.

To initialize an OSSClient instance, you must configure the Endpoint field, authentication mode, and Client parameters. Three authentication modes are available: plaintext setting, self-signed, and STS authentication. For more information about how to use the STS authentication mode and learn about the basic knowledge of RAM, see [Authorized access](#). The following code assumes that you have activated RAM and have basic knowledge of it, such as how to obtain the AccessKey ID, AccessKey Secret, and role ID of a RAM user.

Set AccessKeyId, SecretKeyId, and RoleArn parameters, as shown in [Script files](#). You can use the Python SDK to start a local HTTP service. Use the client code to access the local service and obtain the StsToken.AccessKeyId, StsToken.SecretKeyId, and StsToken.SecurityToken field values.

For more information, see the "[STS instructions](#)" section in the sample.

```
NSString * endpoint = " https :// oss - cn - hangzhou .
aliyunos . com ";

// We recommend that you use STS to initialize
the OSSClient instance if you use the mobile
device . For more informatio n , see the " STS
instructio ns " section in the sample , as shown in
https :// github . com / aliyun / aliyun - oss - ios - sdk /
tree / master / DemoByOC .
id < OSSCredent ialProvide r > credential = [[ OSSStsToke
nCredentia lProvider alloc ] initWithAc ccessKeyId :@"
AccessKeyI d " secretKeyI d :@" AccessKeyS ecret "
securityTo ken :@" SecurityTo ken "];

client = [[ OSSClient alloc ] initWithEn dpoint : endpoint
credential Provider : credential ];
```

The use of the OSSClient instance to initiate upload and download requests is thread-safe. You can run multiple tasks concurrently.

3. Upload a file.

Assume that you have created a bucket in the console. An `OSSTask` is returned for each SDK-relevant operation. You can configure a continuation for the task

to be completed asynchronously or call `waitUntilFinished` to wait for the task to be complete.

```
OSSPutObjectRequest * put = [ OSSPutObjectRequest new ];
put . bucketName = @"< bucketName >";
put . objectKey = @"< objectKey >";
put . uploadingData = < NSData * >; // Directly upload
NSData .
put . uploadProgress = ^( int64_t bytesSent , int64_t
totalBytesSent , int64_t totalBytesExpectedToSend ) {
    NSLog (@"%lld , %lld , %lld ", bytesSent , totalBytes
ent , totalBytesExpectedToSend );
};
OSSTask * putTask = [ client putObject : put ];
[ putTask continueWithBlock :^ id ( OSSTask * task ) {
    if ( ! task . error ) {
        NSLog (@" upload object success !") ;
    } else {
        NSLog (@" upload object failed , error : %@", task
. error );
    }
    return nil ;
}];
// Wait until the task is complete .
// [ putTask waitUntilFinished ];
```

4. Download the specified object.

Use the following code to download the specified object as `NSData` :

```
OSSGetObjectRequest * request = [ OSSGetObjectRequest
new ];
request . bucketName = @"< bucketName >";
request . objectKey = @"< objectKey >";
request . downloadProgress = ^( int64_t bytesWritten ,
int64_t totalBytesWritten , int64_t totalBytes
ExpectedToWrite ) {
    NSLog (@"%lld , %lld , %lld ", bytesWritten ,
totalBytesWritten , totalBytesExpectedToWrite );
};
OSSTask * getTask = [ client getObject : request ];
[ getTask continueWithBlock :^ id ( OSSTask * task ) {
    if ( ! task . error ) {
        NSLog (@" download object success !") ;
        OSSGetObjectResult * getResult = task . result ;
        NSLog (@" download result : %@", getResult .
downloadedData );
    } else {
        NSLog (@" download object failed , error : %@",
task . error );
    }
    return nil ;
}];
// Wait until the task is complete .
```

```
// [ task waitUntilF inished ];
```

- MAC platform

Aside from the import method, all operations on the MAC platform are the same as those on the iOS platform.

```
import < AliyunOSS0 SX / AliyunOSSi OS . h >
```

10.4 Initialization

The OSSClient is the iOS client of OSS. It provides the caller with a series of methods to operate and manage buckets and objects. Before you use the OSS iOS SDK to initiate a request to OSS, you need to first initialize an OSSClient instance and complete necessary settings.



Note:

Make sure that the lifecycle of the OSSClient is consistent with that of the application . When an application is started, create an OSSClient instance. When the application is ended, release the instance.

Determine an endpoint

An endpoint is the URL of Alibaba Cloud OSS in a region. It supports the following two formats:

Endpoint type	Description
OSS endpoint	The URL of the region where the bucket resides. For more information about the endpoints of each region, see Regions and endpoints .
Custom domain	The domain defined by the user. CNAME allows you to customize the OSS domain.

- OSS endpoints

When you use the endpoint of a bucket, you can query the endpoint in either of the following ways:

- Query the relationship between the endpoint and region, as shown in [Regions and endpoints](#).
- You can log on to the [OSS console](#). In the left-side navigation pane, locate the Buckets section, enter bucket-1 in the search box and click the search icon. Click the bucket name to go to the Overview tab. In the Domain Names section, the suffix of `bucket - 1 . oss - cn - hangzhou . aliyuncs . com` is `oss - cn - hangzhou . aliyuncs . com`, which is the endpoint for Internet access.

- Custom domains

Use CNAME to bind a custom domain to a bucket, and then use the domain to access objects in the bucket.

For example, to bind the new-image.xxxxx.com domain to the bucket named image in China (Shenzhen), follow these steps: Contact your hosting provider for xxxxx.com to configure a new domain name resolution record. This record is used to resolve `http :// new - image . xxxxx . com` to `http :// image . oss - cn - shenzhen . aliyuncs . com`. The record type is CNAME.

Configure an endpoint and credential

A mobile device is an untrusted environment. It is highly risky to store `AccessKeyId` and `AccessKeySecret` in a mobile device to sign requests. We recommend that you use STS Authentication Mode or Self - signed Mode. For more information, see [Authorized access](#) and [Direct transmission through a mobile device](#).



Note:

If the STS authentication mode is used, we recommend that you use `OSSAuthCredentialsProvider` to directly access STS. The token is automatically updated after it expires.

The following example shows you how to configure the Endpoint and Credential Provider fields:

```
NSString * endpoint = "http :// oss - cn - hangzhou . aliyuncs . com ";
```

```
// Use the AccessKey ID and AccessKey Secret that are
// issued by Alibaba Cloud to construct a credential
// provider .
// We recommend that you use the plaintext AccessKey
// pair only for tests . For more information about
// authentication modes , see the " Authorized access "
// section .
id < OSSCredentialProvider > credential = [[ OSSPlainTextAKSKPair
CredentialProvider alloc ] initWithPlainTextAccessKey :@"< your
accessKeyId >" secretKey :@"< your
accessKeySecret >"];

client = [[ OSSClient alloc ] initWithEndpoint : endpoint
credentialProvider : credential ];
```

Set the Endpoint field to CNAME

If you bind CNAME to a bucket, you can set the Endpoint field to CNAME.

```
NSString * endpoint = " http :// new - image . xxxxx . com ";

id < OSSCredentialProvider > credential = [[ OSSPlainTextAKSKPair
CredentialProvider alloc ] initWithPlainTextAccessKey :@"< your
accessKeyId >" secretKey :@"< your
accessKeySecret >"];

client = [[ OSSClient alloc ] initWithEndpoint : endpoint
credentialProvider : credential ];
```



Notice:

After Apple implemented the ATS standard, all Endpoint URLs must be HTTPS URLs, while domains set by CNAME do not support certificate settings. Therefore, you cannot use CNAME to set the Endpoint field.

Enable access logging

Mobile devices are used in various environments. The OSS SDK cannot provide normal services in some regions or in a certain period of time. To further locate problems developers encounter, the OSS SDK records the log information locally after access logging is enabled. Before you use the OSSClient, initialize the OSSClient instance and call the following method to enable access logging:

```
// The log entry style .
// 2017 / 10 / 25 11 : 05 : 43 : 863 [ Debug ]: the 17th : <
NSThread : 0x7f809910 8580 >{ number = 3 , name = ( null )}
// 2017 / 10 / 25 11 : 05 : 43 : 863 [ Debug ]: the 15th : <
NSThread : 0x7f809760 52c0 >
// 2017 / 10 / 25 11 : 05 : 43 : 863 [ Debug ]: -----
TestDebug -----
```

```
[ OSSLog enableLog ]; // Run this method to enable access logging .
```

**Note:**

- Logs are stored in the Caches/OSSLogs folder of the sandbox.
- You can upload logs to the server to further track problems.

You can also access Alibaba Cloud [Log Service](#) to upload logs.

Configure ClientConfiguration

Use the following code to configure ClientConfiguration in detail during the initialization process:

```
NSString * endpoint = " https :// oss - cn - hangzhou . aliyuncs . com ";

// We recommend that you use the plaintext AccessKey pair only for tests . For more information about authentication modes , see the " Authorized access " section .
id < OSSCredentialProvider > credential = [[ OSSPlainTextAKSKPair Credential Provider alloc ] initWithPlainTextAccessKey :@"< your accessKeyId >" secretKey :@"< your accessKeySecret >"];

OSSClientConfiguration * conf = [ OSSClientConfiguration new ];
conf . maxRetryCount = 3 ; // The number of retry attempts after the network request encounters an exception and fails .
conf . timeoutIntervalForRequest = 30 ; // The timeout period of a network request .
conf . timeoutIntervalForResource = 24 * 60 * 60 ; // The maximum timeout period for resource transmission .

client = [[ OSSClient alloc ] initWithEndpoint : endpoint credentialProvider : credential clientConfiguration : conf ];
```

OSSTask

An OSSTask is immediately returned for all operations that call APIs. Example:

```
OSSTask * task = [ client getObject : get ];
```

To implement an asynchronous callback, you can configure a continuation for the task. Example:

```
[ task continueWithBlock : ^( OSSTask * task ) {
    // do something
    ...
    return nil ;
}];
```



```
}};
```

You can also wait until the task is complete (synchronous wait). Example:

```
[ task waitUntilF inished ];
```

10.5 Manage a bucket

A bucket serves as a container that stores objects. Objects belong to the bucket they are created in or uploaded to.

Create a bucket

Use the following code to create a bucket:

```
OSSCreateB ucketReque st * create = [ OSSCreateB ucketReque st
    new ];
create . bucketName = @"< bucketName >";
create . xOssACL = @" public - read ";
create . location = @" oss - cn - hangzhou ";

OSSTask * createTask = [ client createBuck et : create ];

[ createTask continueWi thBlock :^ id ( OSSTask * task ) {
    if (! task . error ) {
        NSLog (@" create bucket success !") ;
    } else {
        NSLog (@" create bucket failed , error : %@", task . error );
    }
    return nil ;
}];
```



Note:

- Each user can create a maximum of 30 buckets.
- The name of each bucket must be globally unique. In other words, a user cannot create a bucket with the same name as existing buckets. In this case, the bucket fails to be created.
- When you create a bucket, you can configure the bucket ACL. If no ACL is configured, the bucket ACL is Private.
- The region where the bucket belongs to is returned if the bucket is created.

List buckets

Use the following code to list all buckets owned by the requester:

```
OSSGetServ iceRequest * getService = [ OSSGetServ iceRequest
    new ];
OSSTask * getService Task = [ client getService : getService ];
[ getService Task continueWi thBlock :^ id ( OSSTask * task ) {
```

```

        if (! task . error ) {
            OSSGetServ iceResult * result = task . result ;
            NSLog (@" buckets : %@", result . buckets );
            NSLog (@" owner : %@, %@", result . ownerId , result .
ownerDispN ame );
            [ result . buckets enumerateO bjectsUsin gBlock:^( id
_Nonnull obj , NSUInteger idx , BOOL * _Nonnull stop ) {
                NSDictionary * bucketInfo = obj ;
                NSLog (@" BucketName : %@", [ bucketInfo objectForKey ey
:@" Name "]);
                NSLog (@" CreationDa te : %@", [ bucketInfo
objectForKey ey:@" CreationDa te "]);
                NSLog (@" Location : %@", [ bucketInfo objectForKey ey
:@" Location "]);
            }];
        }
        return nil ;
    }
};

```

**Note:**

Anonymous users do not have the permission to list buckets.

List objects in a bucket

Use the following code to list objects in a bucket:

```

OSSGetBuck etRequest * getBucket = [ OSSGetBuck etRequest new
];
getBucket . bucketName = @"< bucketName >";
// getBucket . marker = @"";
// getBucket . prefix = @"";
// getBucket . delimiter = @"";
OSSTask * getBucketT ask = [ client getBucket : getBucket ];
[ getBucketT ask continueWi thBlock:^( id ( OSSTask * task ) {
    if (! task . error ) {
        OSSGetBuck etResult * result = task . result ;
        NSLog (@" get bucket success !") ;
        for ( NSDictionary * objectInfo in result . contents
) {
            NSLog (@" list object : %@", objectInfo );
        }
    } else {
        NSLog (@" get bucket failed , error : %@", task . error
);
    }
}
return nil ;
}];

```

**Note:**

- To list the objects in the bucket, you must have the operation permission on the bucket.
- You can set the prefix, marker, delimiter, and max-keys parameters to list desired results only.

Delete a bucket

Use the following code to delete a bucket:

```
OSSDeleteBucketRequest * delete = [OSSDeleteBucketRequest
    new];
delete.bucketName = @"< bucketName >";
OSSTask * deleteTask = [client deleteBucket : delete];
[deleteTask continueWithBlock:^(id<OSSTask> * task) {
    if (!task.error) {
        NSLog(@"delete bucket success!");
    } else {
        NSLog(@"delete bucket failed, error: %@", task.error);
    }
    return nil;
}];
```

- Only the bucket owner has the permission to delete the bucket.
- To prevent unexpected data loss, OSS does not allow bucket owners to delete non-empty buckets.

10.6 Upload objects

10.6.1 Overview

This document describes the file upload methods provided by the OSS iOS SDK.

Objects are the basic data units for user operations in OSS. The OSS iOS SDK provides the following object upload methods:

- **Simple upload:** supports the upload of a file up to 5 GB in size. It includes local file upload and upload from the memory.
- **Multipart upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files.
- **Append upload:** supports the upload of a file up to 5 GB in size.
- **Resumable upload:** supports the upload of a file up to 48.8 TB in size. It applies to the upload of large files. It supports concurrent upload and you can define the size of each part.



Note:

For more information about the scenarios of each upload method, see the "Upload files" section in OSS Developer Guide.

After you complete the file upload, you can perform upload callback, as shown in [Upload callback](#).

10.6.2 Simple upload

This topic describes how to use simple upload.

Upload a file from memory or upload a local file

You can upload OSSData directly or use NSURL to upload a file:

```
OSSPutObjectRequest * put = [ OSSPutObjectRequest new ];
// Configure required fields .
put . bucketName = @"< bucketName >";
put . objectKey = @"< objectKey >";
put . uploadingFileURL = [ NSURL fileURLWithPath :@"< filepath >"];
// put . uploadingData = < NSData * >; // Directly upload NSData .
// Configure optional fields .
put . uploadProgress = ^( int64_t bytesSent , int64_t totalBytesSent , int64_t totalBytesExpectedToSend ) {
    // The length of the segment being uploaded , the total length of the segments uploaded , and the total length of the segments to be uploaded .
    NSLog (@"%lld , %lld , %lld " , bytesSent , totalBytesSent , totalBytesExpectedToSend );
};
// For more information about the descriptions of optional fields , see https://docs.aliyun.com/pub/oss/api-reference/object&PutObject .
// put . contentType = @"";
// put . contentMd5 = @"";
// put . contentEncoding = @"";
// put . contentDisposition = @"";
// put . objectMeta = [ NSMutableDictionary dictionaryWithObjectsAndKeys :@"value1 " , @"x-oss-meta-name1 " , nil ]; // You can configure Object Meta or other HTTP headers for the upload .
OSSTask * putTask = [ client putObject : put ];
[ putTask continueWithBlock :^ id ( OSSTask * task ) {
    if ( ! task . error ) {
        NSLog (@"upload object success !") ;
    } else {
        NSLog (@"upload object failed , error : %@", task . error );
    }
    return nil ;
}];
// [ putTask waitUntilFinished ];
// [ putTask cancel ];
```

Upload a file to a directory

OSS does not use folders. All elements are stored as objects. To simulate a folder in the OSS console, you can create an object whose name ends with a forward slash (/). This object can be uploaded and downloaded. The OSS console displays an object whose name ends with a forward slash (/) as a folder.

When you upload a file, if you write ObjectKey as " folder / subfolder / file ", you are simulating the upload of the file to the file object in the folder / subfolder / directory. Note that the default path is the root directory, and does not need to start with a forward slash (/).

Configure the Content-Type field and enable MD5 verification during an upload

You can explicitly specify the Content-Type field during the upload. If it is not specified, the SDK automatically determines its value based on the file name or the uploaded ObjectKey field value. Additionally, if the Content-MD5 field is set when an object is uploaded, OSS uses the Content-MD5 field value to check whether the received object content is consistent with the sent object content. The SDK provides methods to calculate the Base64 and MD5 values.

```
OSSPutObjectRequest * put = [OSSPutObjectRequest new];
// Configure required fields.
put.bucketName = @"< bucketName >";
put.objectKey = @"< objectKey >";
put.uploadingFileURL = [NSURL URLWithString:@"< filepath >"];
// put.uploadingData = < NSData *>; // Directly upload NSData.
// Optional. Set the Content-Type field.
put.contentType = @"application/octet-stream";
// Optional. Set the MD5 verification.
put.contentMd5 = [OSSUtil base64Md5FromFilePath:@"< filePath >"]; // Set the MD5 value of the file path.
// put.contentMd5 = [OSSUtil base64Md5FromData:< NSData *>]; // Set the MD5 value for the binary data.
// Optional. Set the upload progress.
put.uploadProgress = ^(int64_t bytesSent, int64_t totalBytesSent, int64_t totalBytesExpectedToSend) {
    // The length of the segment being uploaded, the total length of the segments uploaded, and the total length of the segments to be uploaded.
    NSLog(@"%lld, %lld, %lld", bytesSent, totalBytesSent, totalBytesExpectedToSend);
};
OSSTask * putTask = [client putObject:put];
[putTask continueWithBlock:^(id<OSSTask> * task) {
    if (!task.error) {
        NSLog(@"upload object success!");
    } else {
        NSLog(@"upload object failed, error: %@", task.error);
    }
    return nil;
}];
// [putTask waitUntilFinished];
```

```
// [ put    cancel ];
```

10.6.3 Multipart upload

This topic describes how to use multipart upload.

Initialize a multipart upload task

Use the following code to initialize a multipart upload task:

```
__block NSString * uploadId = nil ;
__block NSMutableArray * partInfos = [ NSMutableArray new ];
NSString * uploadToBucket = @"< bucketName >";
NSString * uploadObjectKey = @"< objectKey >";
OSSInitMultipartUploadRequest * init = [ OSSInitMultipartUploadRequest new ];
init.bucketName = uploadToBucket ;
init.objectKey = uploadObjectKey ;
// init.contentType = @"application/octet-stream";
OSSTask * initTask = [ client multipartUploadInit : init ];
[ initTask waitUntilFinished ];
if (! initTask.error) {
    OSSInitMultipartUploadResult * result = initTask.result ;
    uploadId = result.uploadId ;
} else {
    NSLog(@"multipart upload failed , error : %@", initTask.error);
    return ;
}
```



Note:

- `OSSInitMultipartUploadRequest` is used to specify the name of the object you want to upload and the name of the bucket to which the object belongs.
- The response to `multipartUploadInit` contains the upload ID. The upload ID is the unique identifier of the multipart upload task.

Upload parts

Use the following code to upload parts:

```
NSString * filePath = [ docDir stringByAppendingPathComponent:@"***"];
// The object you want to upload .
int chunkCount = *;
// The number of parts .
uint64_t offset = fileSize / chunkCount ;
// The size of each part .
for ( int i = 1 ; i <= chunkCount ; i ++ ) {
    OSSUploadPartRequest * uploadPart = [ OSSUploadPartRequest new ];
    uploadPart.bucketName = uploadToBucket ;
    uploadPart.objectKey = uploadObjectKey ;
    uploadPart.uploadId = uploadId ;
```

```

    uploadPart . partNumber = i ; // part number start from
    1

    NSFileHandle * readHandle = [ NSFileHandle fileHandle
    ForReadingAtPath : filePath ];
    [ readHandle seekToFileOffset : offset * ( i - 1 ) ];

    NSData * data = [ readHandle readDataOfLength : offset
    ];
    uploadPart . uploadPartData = data ;

    OSSTask * uploadPartTask = [ client uploadPart : uploadPart
    ];

    [ uploadPartTask waitUntilFinished ];

    if ( ! uploadPartTask . error ) {
        OSSUploadPartResult * result = uploadPartTask . result ;
        uint64_t fileSize = [[[ NSFileManager defaultManager
    ] attributesOfItemAtPath : uploadPart . uploadPartFileURL .
    absoluteString error : nil ] fileSize ];
        [ partInfos addObject : [ OSSPartInfo partInfoWithPartNum : i
    eTag : result . eTag size : fileSize ] ];
    } else {
        NSLog ( @" upload part error : %@", uploadPartTask . error );
        return ;
    }
}
}

```

The preceding code uses `uploadPart` to upload each part.



Notice:

- An upload ID and a part number must be specified for each multipart upload request.
- If you use `uploadPart`, each part must be larger than 100 KB aside from the last part. However, the `UploadPart` API does not check the uploaded part size immediately. This API checks the part size only when multipart upload is complete.
- The part number ranges from 1 to 10,000. If a part number is beyond this range, OSS returns the `InvalidArgument` error code.
- When each part is uploaded, the stream is directed to the starting position of the uploaded part.
- After each part is uploaded, OSS returns a response that contains the ETag value of the part. If the ETag value and the MD5 value of the part are the same, you must combine the ETag value with the part number into `PartETag` and save the `PartETag` for subsequent upload of parts.

Complete the multipart upload

In the following code, the `partInfos` field value indicates the list of `PartETags` saved during the multipart upload. OSS verifies the validity of each part after OSS receives the `partInfos` field value. When all data parts are verified to be valid, OSS combines them into a complete object.

```
OSSComplet eMultipart UploadRequest * complete = [ OSSComplet
eMultipart UploadRequest new ];
complete . bucketName = uploadToBucket ;
complete . objectKey = uploadObjectKey ;
complete . uploadId = uploadId ;
complete . partInfos = partInfos ;

OSSTask * completeTask = [ client completeMultipartUpload
: complete ];

[[ completeTask continueWithBlock:^(id ( OSSTask * task ) {
    if (! task . error ) {
        OSSComplet eMultipart UploadResult * result = task . result
        ;
        // ...
    } else {
        // ...
    }
    return nil ;
}]] waitUntilFinished ];
```

You can set the `Server Callback` parameter to complete the multipart upload request . A callback request is sent to the specified server address after the multipart upload task is complete. You can view the `servercallback` result in `result.serverReturnJsonObject` of the response.

```
OSSComplet eMultipart UploadRequest * complete = [ OSSComplet
eMultipart UploadRequest new ];
complete . bucketName = @"< bucketName >";
complete . objectKey = @"< objectKey >";
complete . uploadId = uploadId ;
complete . partInfos = partInfos ;
complete . callbackParameters = @{
    @" callbackURL ": @"< server address >",
    @" callbackBody ": @"< test >"
};
complete . callbackVariables = @{
    @" var1 ": @" value1 ",
    @" var2 ": @" value2 "
};
OSSTask * completeTask = [ client completeMultipartUpload
: complete ];
[[ completeTask continueWithBlock:^(id ( OSSTask * task ) {
    if (! task . error ) {
        OSSComplet eMultipart UploadResult * result = task .
result ;
        NSLog(@" server call back return : %@", result .
serverReturnJsonObject );
    } else {
        // ...
    }
}]];
```



```

    }
    return nil ;
}] waitUntilF inished ];

```

**Note:**

To check whether the object uploaded to OSS is consistent with the local file, attach the Content-MD5 field value to the upload request. The OSS server helps you with the MD5 verification. The upload can succeed only when the MD5 value of the object received by the OSS server is the same as the Content-MD5 field value. This method can ensure the consistency of uploaded object.

Use the following code to configure MD5 verification:

```

OSSUploadP artRequest * uploadPart = [ OSSUploadP artRequest
new ];
uploadPart . bucketName = TEST_BUCKE T ;
uploadPart . uploadId = uploadId ;
...
uploadPart . contentMd5 = [ OSSUtil fileMD5Str ing : filepath ];

```

List parts

Use the following code to call `listParts` to obtain all uploaded parts of an upload event:

```

OSSListPar tsRequest * listParts = [ OSSListPar tsRequest new
];
listParts . bucketName = @"< bucketName >";
listParts . objectKey = @"< objectkey >";
listParts . uploadId = @"< uploadid >";

OSSTask * listPartTa sk = [ client listParts : listParts ];

[ listPartTa sk continueWi thBlock :^ id ( OSSTask * task ) {
    if ( ! task . error ) {
        NSLog (@" list part result success !") ;
        OSSListPar tsResult * listPartRe sult = task . result ;
        for ( NSDictionary * partInfo in listPartRe sult . parts
        ) {
            NSLog (@" each part : %@", partInfo );
        }
    } else {
        NSLog (@" list part result error : %@", task . error );
    }
    return nil ;
}];

```

**Notice:**

If a bucket contains more than 1,000 parts that are uploaded by using multipart upload, OSS returns the first 1,000 parts by default. In the response, the `IsTruncated` field is set to false and `NextPartNumberMarker` is returned to indicate the next starting position to continue reading data.

Cancel a multipart upload event

Use the following code to cancel a multipart upload request that is mapped to the specified upload ID:

```
OSSAbortMultipartUploadRequest * abort = [OSSAbortMultipartUploadRequest new];
abort.bucketName = @"< bucketName >";
abort.objectKey = @"< objectKey >";
abort.uploadId = uploadId;

OSSTask * abortTask = [client abortMultipartUpload:abort];

[abortTask waitUntilFinished];

if (!abortTask.error) {
    OSSAbortMultipartUploadResult * result = abortTask.result;
    uploadId = result.uploadId;
} else {
    NSLog(@"multipart upload failed, error: %@", abortTask.error);
    return;
}
```

10.6.4 Append upload

This topic describes how to use append upload.

You can use `AppendObject` to append objects. The object that is created by using `AppendObject` is an Appendable object, while the object that is uploaded by using `PutObject` is a Normal object.

Use the following code for append upload:

```
OSSAppendObjectRequest * append = [OSSAppendObjectRequest new];
// Configure required fields.
append.bucketName = @"< bucketName >";
append.objectKey = @"< objectKey >";
append.appendPosition = 0; // Specify the starting position to append the object.
NSString * docDir = [self getDocumentDirectory];
append.uploadingFileURL = [NSURL fileURLWithPath:@"< filepath >"];
// Configure optional fields.
append.uploadProgress = ^(int64_t bytesSent, int64_t totalBytesSent, int64_t totalBytesExpectedToSend) {
    NSLog(@"%lld, %lld, %lld", bytesSent, totalBytesSent, totalBytesExpectedToSend);
};
// For more information about the descriptions of optional fields, see https://docs.aliyun.com/#/pub/oss/api-reference/object/AppendObject.
// append.contentType = @"";
// append.contentMd5 = @"";
// append.contentEncoding = @"";
// append.contentDisposition = @"";
```

```

OSSTask * appendTask = [ client appendObject : append ];
[ appendTask continueWithBlock:^(id<OSSTask> * task) {
    NSLog(@"objectKey: %@", append.objectKey);
    if (!task.error) {
        NSLog(@"append object success!");
        OSSAppendObjectResult * result = task.result;
        NSString * etag = result.eTag;
        long nextPosition = result.xOssNextAppendPosition;
    } else {
        NSLog(@"append object failed, error: %@", task.error);
    }
    return nil;
}];

```

10.6.5 Resumable upload

This topic describes how to use resumable upload.

In a wireless network environment, it takes a long time to upload a large file. The upload may fail due to poor network connectivity or network switches. In this case, the entire file needs to be uploaded again. To address this problem, the SDK provides resumable upload.



Notice:

- Currently, this feature allows you to upload only local files.
- Resumable upload is inapplicable to small- and medium-sized files from a mobile device. Resumable upload adopts the multipart upload method. This method is inefficient because it initiates multiple network requests for uploading a single file.

Before you upload a file, you can specify the folder for the file that stores resumable upload records. Resumable upload takes effect only for the current upload task. If you do not specify the folder for the file that stores resumable upload records, a part of a large file may fail to be uploaded due to network errors. To upload the entire file, large amounts of time and traffic are required to make retry attempts. If you specify the folder for the file that stores resumable upload records and the upload fails, the reupload starts from the recorded position. For more information, see the subsequent example.

When a resumable upload task fails and is not resumed for a long time, the uploaded parts may become unnecessary in OSS. In this case, you can set the lifecycle rule for the bucket to clear parts periodically. For more information, see [Lifecycle management](#).

To manage parts, if the current task is cancelled during resumable upload, the parts already uploaded to the server are cleared synchronously by default. If you want to save the resumable upload record, you must specify a folder of a file that stores the resumable upload record and modify the `deleteUploadIdOnCancelling` parameter. If the resumable upload record is saved on the local device for a long time but the parts uploaded to the server are deleted based on the lifecycle rule configured for the bucket, the records on the server and the mobile device may be inconsistent.



Note:

- Resumable upload depends on `InitMultipartUpload / UploadPart / ListParts / CompleteMultipartUpload / AbortMultipartUpload`. If you use the STS authentication mode, add the permissions required to use the APIs.
- Resumable upload supports post-upload callback notifications that can be used in the same way as the preceding typical upload callback notifications.
- MD5 verification is automatically enabled for each multipart upload task during resumable upload. You do not need to set the `Content-Md5` field in the request header.
- The invocation method (not specified by default) to store resumable upload records permanently in the local device:

```
OSSResumableUploadRequest * resumableUpload = [
    OSSResumableUploadRequest new];
resumableUpload.bucketName = OSS_BUCKET_PRIVATE;
//...
NSString * cachesDir = [NSSearchPathForDirectoriesInDomains
    (NSCachesDirectory, NSUserDomainMask, YES)
    firstObject];
resumableUpload.recordDirectoryPath = cachesDir;
```

- Implementation of resumable upload

```
// Obtain an upload ID to upload the file. If the
// task fails and can be resumed, you can use the
// same upload ID to upload the same file to the
// same bucket.
OSSResumableUploadRequest * resumableUpload = [
    OSSResumableUploadRequest new];
resumableUpload.bucketName = <bucketName>;
resumableUpload.objectKey = <objectKey>;
resumableUpload.partSize = 1024 * 1024;
resumableUpload.uploadProgress = ^(int64_t bytesSent,
    int64_t totalBytesSent, int64_t totalBytesExpectedToSend) {
    NSLog(@"%lld, %lld, %lld", bytesSent, totalBytesSent,
        totalBytesExpectedToSend);
```

```

};
NSString * cachesDir = [ NSSearchPa thForDirec toriesInDo
    mains ( NSCachesDi rectory , NSUserDoma inMask , YES )
    firstObjec t ];
// Configure the checkpoint file .
resumableU pload . recordDire ctoryPath = cachesDir ;
// Set the resumableU pload . deleteUplo adIdOnCanc elling
    field to No . When a task is cancelled , the
    checkpoint file is not deleted . If you retain
    default value Yes , the checkpoint file is deleted
    when a task is cancelled , and the entire file
    needs to be uploaded again when the task restarts .
resumableU pload . deleteUplo adIdOnCanc elling = NO ;

resumableU pload . uploadingF ileURL = [ NSURL fileURLWit
    hPath :< your file path >];
OSSTask * resumeTask = [ client resumableU pload :
    resumableU pload ];
[ resumeTask continueWi thBlock :^ id ( OSSTask * task ) {
    if ( task . error ) {
        NSLog (@" error : %@", task . error );
        if ( [ task . error . domain isEqualToS tring :
            OSSClientE rrorDomain ] && task . error . code == OSSClientE
            rrorCodeCa nnotResume Upload ) {
            // The task cannot be resumed . You must
            obtain a new upload ID to upload the file .
        }
    } else {
        NSLog (@" Upload file success ");
    }
    return nil ;
}];

// [ resumeTask waitUntilF inished ];
// [ resumableU pload cancel ];

```

10.6.6 Upload callback

This topic describes how to use upload callback.

When an upload is complete, the OSS instance can send a callback to the applicatio
n server. To implement the callback, you need only to add the relevant callback
parameters to the request that is sent to OSS. Compared with simple upload, the
upload callback notification mechanism requires the client to wait for a longer time
to process a callback request and response.

For more information, see [Callback](#).

Use the following code for upload callback:

```

OSSPutObje ctRequest * request = [ OSSPutObje ctRequest new ];
request . bucketName = @"< bucketName >";
request . objectKey = @"< objectKey >";
request . uploadingF ileURL = [ NSURL fileURLWit hPath :@<
    filepath >"];
// Set callback parameters .
request . callbackPa ram = @{

```

```

        @" callbackUrl ": @"< your server
callback address >",
        @" callbackBody ": @"< your callback
body >"
    };
// Set custom variables .
request . callbackVar = @{
    @"< var1 >": @"< value1 >",
    @"< var2 >": @"< value2 >"
};
request . uploadProgress = ^( int64_t bytesSent , int64_t
totalBytesEnt , int64_t totalBytes ExpectedTo Send ) {
    NSLog (@"%lld , %lld , %lld ", bytesSent , totalBytesEnt
, totalBytes ExpectedTo Send );
};
OSSTask * task = [ client putObject : request ];
[ task continueWithBlock :^ id ( OSSTask * task ) {
    if ( task . error ) {
        OSSLogError (@"%@", task . error );
    } else {
        OSSPutObjectResult * result = task . result ;
        NSLog (@" Result - requestId : %@, headerFields : %@,
servercallback : %@",
            result . requestId ,
            result . httpResponseHeaderFields ,
            result . serverReturnJsonString );
    }
    return nil ;
}];

```

10.7 Download objects

10.7.1 Overview

This document describes how to use the OSS iOS SDK to download objects.

The OSS iOS SDK provides the following download methods:

- [Simple download](#)
- [Streaming download](#)
- [Range download](#)
- [Resumable download](#)

10.7.2 Simple download

This topic describes how to use simple download.

You can download an object to a local file or as NSData:

```

OSSGetObjectRequest * request = [ OSSGetObjectRequest new ];
// Configure required fields .
request . bucketName = @"< bucketName >";
request . objectKey = @"< objectKey >";
// Configure optional fields .

```

```

request . downloadProgress = ^( int64_t bytesWritten ,
int64_t totalBytes Written , int64_t totalBytes ExpectedTo
Write ) {
    // The length of the segment being downloaded , the
    total length of the segments downloaded , and the
    total length of the segments to be downloaded .
    NSLog (@"%lld , %lld , %lld " , bytesWritten , totalBytes
Written , totalBytes ExpectedTo Write );
};
// request . range = [ [ OSSRange alloc ] initWithStart : 0
withEnd : 99 ]; // Download the foremost 99 Bytes .
// request . downloadTo FileURL = [ NSURL fileURLWithPath :@"<
filepath >"]; // Specify the destination path if you
need to directly download the object to a file .

OSSTask * getTask = [ client getObject : request ];

[ getTask continueWithBlock :^ id ( OSSTask * task ) {
    if (! task . error ) {
        NSLog (@" download object success !") ;
        OSSGetObjectResult * getResult = task . result ;
        NSLog (@" download result : %@", getResult . downloadedD
ata );
    } else {
        NSLog (@" download object failed , error : %@", task .
error );
    }
    return nil ;
}];

// [ getTask waitUntilFinished ];
// [ request cancel ];

```

10.7.3 Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use streaming download. Streaming download enables you to download part of the object content each time until you download the entire object.

The OSS iOS SDK provides no streaming download APIs. Instead, it provides the multipart callback function similar to `didReceiveData` in the `NSURLSession` library. Note: The download result does not contain the actual data if the multipart callback is configured.

Use the following code for streaming download:

```

OSSGetObjectRequest * request = [ OSSGetObjectRequest new ];
// Configure required fields .
request . bucketName = @"< bucketName >";
request . objectKey = @"< objectKey >";
// Configure the multipart callback function .
request . onReceivedData = ^( NSData * data ) {
    NSLog (@" Receive data , length : %ld " , [ data length ]);
};

```

```

    OSSTask * getTask = [ client getObject : request ];
    [ getTask continueWithBlock:^(id ( OSSTask * task ) {
        if (! task . error ) {
            NSLog (@" download object success !") ;
        } else {
            NSLog (@" download object failed , error : %@", task .
error );
        }
        return nil ;
    }]);
// [ getTask waitUntilFinished ];
// [ request cancel ];

```

10.7.4 Range download

If you only need part of the data in an object, you can use range download to download specified content.

Range download is suitable for downloading large objects. If the Range parameter is specified in the request header, the response contains the length of the entire object and the range of data.

Run the following code for range download:

```

OSSGetObjectRequest * request = [ OSSGetObjectRequest new ];
request . bucketName = @"< bucketName >";
request . objectKey = @"< objectKey >";
request . range = [[ OSSRange alloc ] initWithStart : 1
withEnd : 99 ]; // bytes = 1 - 99
// request . range = [[ OSSRange alloc ] initWithStart :- 1
withEnd : 99 ]; // bytes = - 99
// request . range = [[ OSSRange alloc ] initWithStart : 10
withEnd :- 1 ]; // bytes = 10 -
request . downloadProgress = ^( int64_t bytesWritten ,
int64_t totalBytesWritten , int64_t totalBytesExpectedTo
Write ) {
    NSLog (@"%lld , %lld , %lld ", bytesWritten , totalBytes
Written , totalBytesExpectedToWrite );
};
OSSTask * getTask = [ client getObject : request ];
[ getTask continueWithBlock:^(id ( OSSTask * task ) {
    if (! task . error ) {
        NSLog (@" download object success !") ;
        OSSGetObjectResult * getResult = task . result ;
        NSLog (@" download result : %@", getResult . downloadedD
ata );
    } else {
        NSLog (@" download object failed , error : %@", task .
error );
    }
    return nil ;
}]);
// [ getTask waitUntilFinished ];

```



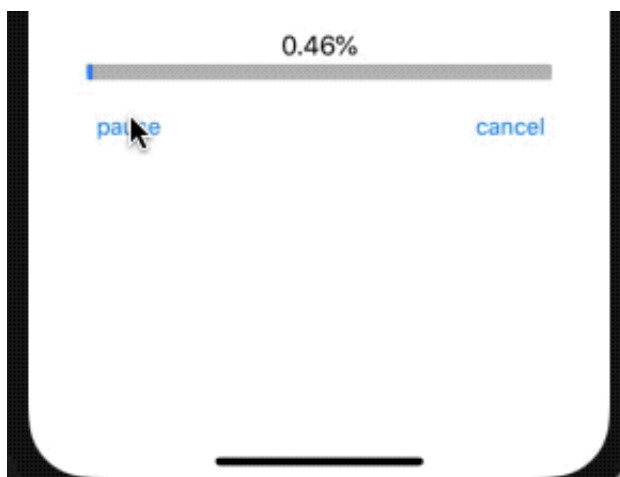
```
// [ request cancel ];
```

10.7.5 Resumable download

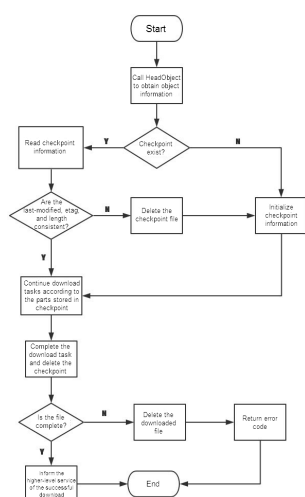
When a client downloads resources over a network, the download is interrupted for some reason. In this case, you can use resumable download to continue downloading the uncompleted parts. This method saves time and traffic.

Scenario: When you use a video application to download a video on a mobile phone, and the network switches from Wi-Fi to Cellular during the download, the application automatically interrupts the download. When you switch back to Wi-Fi, you can manually restart the download task to resume the download.

The following figure shows the effect of resumable download.



The following figure shows the process of resumable download.



Detail analysis

- The Range field is added to the HTTP 1.1 header to specify the scope within which to obtain data. The formats of the Range field value are as follows:
 - `Range : bytes = 100 -` : starts from the 101st Byte until the download is complete.
 - `Range : bytes = 100 - 200` : starts the download from the 101st Byte and stops the download at the 201st Byte. The Range field value is counted from 0 by default. This method is typically used for multipart transmission of large objects, such as videos.
 - `Range : bytes =- 100` : The start position is not specified. The server downloads the last 100 Bytes, instead of the 100 Bytes starting from 0.
 - `Range : bytes = 0 - 100 , 200 - 300` : downloads content within multiple ranges at the same time. This is untypical.
- During resumable download, the server requires to use the `If - Match` value in the header to verify whether the object on the server changes. The `If - Match` field value corresponds to the `ETag` value.
- When the client initiates a request, the `Range` and `If - Match` field values are included in the request. The OSS server verifies the ETag value that corresponds to the If-Match field value. If the received ETag value does not match the ETag value saved in OSS, OSS returns the 412 precondition status code.
- The OSS server currently supports the following fields for GetObject: `Range` , `If - Match` , `If - None - Match` , `If - Modified - Since` , `If - Unmodified - Since` . Therefore, you can use resumable download to download resources from OSS on your mobile device.

Specific implementation

Example:



Note:

The following code is for reference only. We recommend that you do not use it in production projects. To download the code, you can download it by yourself or use third-party open-source download software.

```
# import "DownloadService.h"
# import "OSSTestMacros.h"
```

```

@ implementation DownloadRequest
@ end

@ implementation Checkpoint
- (instancetype) copyWithZone:(NSZone *) zone {
    Checkpoint * other = [[[ self class ] allocWithZone: zone
    ] init];

    other . etag = self . etag ;
    other . totalExpectedLength = self . totalExpectedLength ;

    return other ;
}
@ end

@interface DownloadService () < NSURLSessionTaskDelegate ,
    NSURLSessionDataDelegate >

@property (nonatomic , strong ) NSURLSession * session ;
    // The network session .
@property (nonatomic , strong ) NSURLSessionDataTask *
    dataTask ; // The data request task .
@property (nonatomic , copy ) DownloadFailureBlock failure ;
    // The request error .
@property (nonatomic , copy ) DownloadSuccessBlock success ;
    // The request succeeds .
@property (nonatomic , copy ) DownloadProgressBlock
    progress ; // The download progress .
@property (nonatomic , copy ) Checkpoint * checkpoint ;
    // The checkpoint file .
@property (nonatomic , copy ) NSString * requestURLString
    ; // The object resource URL used in a download
    request .
@property (nonatomic , copy ) NSString * headURLString ;
    // The object resource URL used in a HEAD request .
@property (nonatomic , copy ) NSString * targetPath ; // The
    path to the object .
@property (nonatomic , assign ) unsigned long long
    totalReceivedContentLength ; // The size of the
    downloaded content .
@property (nonatomic , strong ) dispatch_semaphore_t
    semaphore ;

@ end

@ implementation DownloadService

- (instancetype) init
{
    self = [ super init ];
    if ( self ) {
        NSURLSessionConfiguration * conf = [ NSURLSession
        configuration defaultSessionConfiguration ];
        conf . timeoutIntervalForRequest = 15 ;

        NSOperationQueue * processQueue = [ NSOperationQueue
        new ];
        _session = [ NSURLSession sessionWithConfiguration :
        conf delegate : self delegateQueue : processQueue ];
        _semaphore = dispatch_semaphore_create ( 0 );
    }
}

```

```

        _checkpoint t = [[ Checkpoint alloc ] init ];
    }
    return self ;
}

+ (instancetype) downloadServiceWithRequest:(DownloadRequest *) request {
    DownloadService * service = [[ DownloadService alloc ]
init ];
    if ( service ) {
        service . failure = request . failure ;
        service . success = request . success ;
        service . requestURLString = request . sourceURLString ;
    ;
        service . headURLString = request . headURLString ;
        service . targetPath = request . downloadFilePath ;
        service . progress = request . downloadProgress ;
        if ( request . checkpoint ) {
            service . checkpoint = request . checkpoint ;
        }
    }
    return service ;
}

/**
 * Use HeadObject to obtain Object Meta . Compare the
 * ETag value saved in OSS with that saved in the
 * local checkpoint file . The result is returned .
 */
- ( BOOL ) getFileInfo {
    __block BOOL resumable = NO ;
    NSURL * url = [ NSURL URLWithString : self . headURLString ];
    NSMutableURLRequest * request = [[ NSMutableURLRequest
alloc ] initWithURL : url ];
    [ request setHTTPMethod : @" HEAD " ];

    NSURLSessionDataTask * task = [[ NSURLSession sharedSession ]
dataTaskWithRequest : request completion Handler : ^(
NSData * _Nullable data , NSURLResponse * _Nullable
response , NSError * _Nullable error ) {
        if ( error ) {
            NSLog ( @" Failed to obtain Object Meta . error
: %@" , error );
        } else {
            NSHTTPURLResponse * httpResponse = ( NSHTTPURLR
esponse *) response ;
            NSString * etag = [ httpResponse . allHeaderFields
objectForKey : @" Etag " ];
            if ( [ self . checkpoint . etag isEqualToString :
etag ] ) {
                resumable = YES ;
            } else {
                resumable = NO ;
            }
        }
    }
    dispatch_semaphore_signal ( self . semaphore );
}];
[ task resume ];

    dispatch_semaphore_wait ( self . semaphore , DISPATCH_T
IME_FOREVER );
    return resumable ;
}

```

```

/**
 * Obtain the size of the local file .
 */
- ( unsigned long long ) fileSizeAt Path :( NSString *) filePath
{
    unsigned long long fileSize = 0 ;
    NSFileManager * dfm = [ NSFileManager defaultManager ];
    if ( [ dfm fileExistsAtPath : filePath ] ) {
        NSError * error = nil ;
        NSDictionary * attributes = [ dfm attributes
OfItemAtPath : filePath error : & error ];
        if ( ! error && attributes ) {
            fileSize = attributes . fileSize ;
        } else if ( error ) {
            NSLog ( @" error : %@", error );
        }
    }

    return fileSize ;
}

- ( void ) resume {
    NSURL * url = [ NSURL URLWithString : self . requestURL
String ];
    NSMutableURLRequest * request = [ [ NSMutableURLRequest
alloc ] initWithURL : url ];
    [ request setHTTPMethod : @" GET " ];

    BOOL resumable = [ self getFileInfo ]; // If the
resumable field value is NO , resumable upload is
unfeasible . Otherwise , continue the resumable upload
logic .
    if ( resumable ) {
        self . totalReceivedContent Length = [ self fileSizeAt
Path : self . targetPath ];
        NSString * requestRange = [ NSString stringWithFormat
:@" bytes =%llu -", self . totalReceivedContent Length ];
        [ request setValue : requestRange forHTTPHeaderField
:@" Range " ];
    } else {
        self . totalReceivedContent Length = 0 ;
    }

    if ( self . totalReceivedContent Length == 0 ) {
        [ [ NSFileManager defaultManager ] createFileAtPath :
self . targetPath contents : nil attributes : nil ];
    }

    self . dataTask = [ self . session dataTaskWithRequest :
request ];
    [ self . dataTask resume ];
}

- ( void ) pause {
    [ self . dataTask cancel ];
    self . dataTask = nil ;
}

- ( void ) cancel {
    [ self . dataTask cancel ];
    self . dataTask = nil ;
    [ self removeFileAtPath : self . targetPath ];
}

```

```

- ( void ) removeFile AtPath :( NSString *) filePath {
    NSError * error = nil ;
    [[ NSFileManager defaultManager ] removeItemAtPath : self
    . targetPath error :& error ];
    if ( error ) {
        NSLog (@" remove file with error : %@", error );
    }
}

# pragma mark - NSURLSessionDataDelegate

- ( void ) URLSession :( NSURLSession *) session task :(
    NSURLSessionTask *) task
    didFinishWithEror :( nullable NSError *) error {
    NSHTTPURLResponse * httpResponse = ( NSHTTPURLResponse
    *) task . response ;
    if ([ httpResponse isKindOfClass:[ NSHTTPURLResponse
    class ]]) {
        if ( httpResponse . statusCode == 200 ) {
            self . checkpoint . etag = [[ httpResponse
            allHeaderFields ] objectForKey:@" Etag "];
            self . checkpoint . totalExpectedLength = httpResponse
            . expectedContentLength ;
        } else if ( httpResponse . statusCode == 206 ) {
            self . checkpoint . etag = [[ httpResponse
            allHeaderFields ] objectForKey:@" Etag "];
            self . checkpoint . totalExpectedLength = self .
            totalReceivedContentLength + httpResponse . expectedCo
            ntentLength ;
        }
    }

    if ( error ) {
        if ( self . failure ) {
            NSMutableDictionary * userInfo = [ NSMutableDictionary
            dictionaryWithDictionary : error . userInfo ];
            [ userInfo setObject : self . checkpoint forKey
            :@" checkpoint "];

            NSError * tError = [ NSError errorWithDomain :
            error . domain code : error . code userInfo : userInfo ];
            self . failure ( tError );
        }
        else if ( self . success ) {
            self . success (@" status ": @" success ");
        }
    }
}

- ( void ) URLSession :( NSURLSession *) session dataTask :
( NSURLSessionDataTask *) dataTask didReceiveResponse :(
    NSURLResponse *) response completionHandler :( void (^)(
    NSURLSessionResponseDisposition)) completionHandler
{
    NSHTTPURLResponse * httpResponse = ( NSHTTPURLResponse
    *) dataTask . response ;
    if ([ httpResponse isKindOfClass:[ NSHTTPURLResponse
    class ]]) {
        if ( httpResponse . statusCode == 200 ) {
            self . checkpoint . totalExpectedLength = httpResponse
            . expectedContentLength ;
        } else if ( httpResponse . statusCode == 206 ) {

```

```

        self . checkpoint . totalExpectedLength = self .
totalReceivedContentLength + httpResponse . expectedContentLength ;
    }
}

completionHandler ( NSURLSessionResponseAllow );
}

- ( void ) URLSession : ( NSURLSession * ) session dataTask : (
NSURLSessionDataTask * ) dataTask didReceiveData : ( NSData * )
data {

    NSFileHandle * fileHandle = [ NSFileHandle fileHandle
ForWritingAtPath : self . targetPath ];
    [ fileHandle seekToEndOfFile ];
    [ fileHandle writeData : data ];
    [ fileHandle closeFile ];

    self . totalReceivedContentLength += data . length ;
    if ( self . progress ) {
        self . progress ( data . length , self . totalReceivedContentLength , self . checkpoint . totalExpectedLength );
    }
}

@end

```

The preceding sample code shows the process logic when data is received over the network. The fields in the code are described as follows:

- **DownloadService:** the core of the download logic.
- **URLSession:dataTask:didReceiveData:** The received network data is appended to the object. The download progress record is updated.
- **URLSession:task:didCompleteWithError:** determines whether the download task is complete, and then returns the result to the upper layer business.
- **URLSession:dataTask:didReceiveResponse:completionHandler:** processes object-relevant information, such as the ETag value for the precheck of resumable download and the content-length field value to calculate the download progress.

DownloadRequest is defined as follows:

```

# import < Foundation / Foundation . h >

typedef void (^ DownloadProgressBlock )( int64_t bytesReceived , int64_t totalBytesReceived , int64_t totalBytesExpectedToReceive );
typedef void (^ DownloadFailureBlock )( NSError * error );
typedef void (^ DownloadSuccessBlock )( NSDictionary * result );

@interface Checkpoint : NSObject < NSCopying >

@property ( nonatomic , copy ) NSString * etag ; // The ETag value of the resource .

```

```

@property (nonatomic, assign) unsigned long long
totalExpectedLength; // The total size of the object
.

@end

@interface DownloadRequest : NSObject

@property (nonatomic, copy) NSString * sourceURLString;
// The URL for the download.

@property (nonatomic, copy) NSString * headURLString;
// The URL used to obtain the original information
of the object.

@property (nonatomic, copy) NSString * downloadFilePath;
// The local path to the object.

@property (nonatomic, copy) DownloadProgressBlock
downloadProgress; // The download progress.

@property (nonatomic, copy) DownloadFailureBlock failure;
// The callback that is sent after the download
fails.

@property (nonatomic, copy) DownloadSuccessBlock success;
// The callback that is sent after the download
succeeds.

@property (nonatomic, copy) Checkpoint * checkpoint;
// The checkpoint file that stores the ETag value
of the object.

@end

@interface DownloadService : NSObject

+ (instancetype) downloadServiceWithRequest:(DownloadRequest *) request;

/**
 * Start the download.
 */
- (void) resume;

/**
 * Pause the download.
 */
- (void) pause;

/**
 * Cancel the download.
 */
- (void) cancel;

@end

```

The invocation code for the upper-layer business is as follows:

```

- (void) initDownloadURLs {
    OSSPlainTextAKSKPairCredentialProvider * pCredential
    l = [[ OSSPlainTextAKSKPairCredentialProvider alloc ]

```



```

initWithPl ainTextAcc essKey : OSS_ACCESS_KEY_ID secretKey :
OSS_SECRET_KEY_ID ];
_mClient = [ [ OSSClient alloc ] initWithEndpoint :
OSS_ENDPOINT credentialProvider : pCredential ];

// Generate the signed URL for the GET request .
OSSTask * downloadURLTask = [ _mClient presignConstraintURLWithBucketName : @"aliyun-dhc-shanghai"
withObjectKey : OSS_DOWNLOAD_FILE_NAME withExpirationInterval : 1800 ];
[ downloadURLTask waitUntilFinished ];
_downloadURLString = downloadURLTask.result ;

// Generate the signed URL for the HEAD request .
OSSTask * headURLTask = [ _mClient presignConstraintURLWithBucketName : @"aliyun-dhc-shanghai" withObjectKey :
OSS_DOWNLOAD_FILE_NAME httpMethod : @"HEAD" withExpirationInterval : 1800 withParameters : nil ];
[ headURLTask waitUntilFinished ];

_headURLString = headURLTask.result ;
}

- (IBAction) resumeDownloadClicked:(id) sender {
_downloadRequest = [ DownloadRequest new ];
_downloadRequest.sourceURLString = _downloadURLString ;
// Set the resource URL .
_downloadRequest.headURLString = _headURLString ;
NSString * documentPath = NSSearchPathForDirectoriesInDomains ( NSDocumentDirectory , NSUserDomainMask ,
YES ). firstObject ;
_downloadRequest.downloadFilePath = [ documentPath
stringByAppendingPathComponent : OSS_DOWNLOAD_FILE_NAME ];
// Set the local path to the object you want to
download .

__weak typeof ( self ) weakSelf = self ;
_downloadRequest.downloadProgress = ^( int64_t
bytesReceived , int64_t totalBytesReceived , int64_t
totalBytesExpectToReceive ) {
// totalBytesReceived is the number of Bytes
already cached by the client . totalBytesExpectToRe
ceived is the total number of Bytes that need to
be downloaded .
dispatch_async ( dispatch_get_main_queue () , ^(
__strong typeof ( self ) sSelf = weakSelf ;
CGFloat fProgress = totalBytesReceived * 1 . f
/ totalBytesExpectToReceive ;
sSelf.progressLabel.text = [ NSString stringWithFormat
Format : @"%. 2f %%", fProgress * 100 ];
sSelf.progressBar.progress = fProgress ;
});
};
_downloadRequest.failure = ^( NSError * error ) {
__strong typeof ( self ) sSelf = weakSelf ;
sSelf.checkpoint = error.userInfo [@"checkpoint "];
};
_downloadRequest.success = ^( NSDictionary * result ) {
NSLog (@"The download succeeds .");
};
_downloadRequest.checkpoint = self.checkpoint ;

NSString * titleText = [ [_downloadButton titleLabel]
text ];

```

```

        if ([titleText isEqualToString:@"download"]) {
            [_downloadButton setTitle:@"pause" forState:
            UIControlStateNormal];
            _downloadService = [DownloadService downloadServiceWithRequest:_downloadRequest];
            [_downloadService resume];
        } else {
            [_downloadButton setTitle:@"download" forState:
            UIControlStateNormal];
            [_downloadService pause];
        }
    }

- (IBAction)cancelDownloadClicked:(id)sender {
    [_downloadButton setTitle:@"download" forState:
    UIControlStateNormal];
    [_downloadService cancel];
}

```

**Note:**

The checkpoint file can be obtained from the failure callback when the download is paused or cancelled. When you restart the download, you can upload the checkpoint file to `DownloadRequest`. Then, `DownloadService` uses the checkpoint file to verify the consistency.

10.8 Manage objects

10.8.1 Overview

You can use APIs to perform the following operations on objects in a bucket:

- [Determine whether the specified object exists](#)
- [Obtain Object Meta](#)
- [List objects](#)
- [Copy an object](#)
- [Delete objects](#)

10.8.2 Determine whether the specified object exists

The OSS iOS SDK provides a convenient synchronous API to check whether the specified object exists in the bucket.

Use the following code to determine whether an object exists:

```

NSError * error = nil;
BOOL isExist = [client doesObjectExistInBucket:
TEST_BUCKET withObjectKey:@"file1m" withError:&error];
if (!error) {
    if (isExist) {

```

```

        NSLog(@"File exists.");
    } else {
        NSLog(@"File not exists.");
    }
} else {
    NSLog(@"Error !");
}

```

10.8.3 Obtain Object Meta

The `HeadObject` method can be used to obtain only the Object Meta but not the object content.

Use the following code to obtain Object Meta:

```

OSSHeadObjectRequest * request = [OSSHeadObjectRequest new];
request.bucketName = @"<bucketName>";
request.objectKey = @"<objectKey>";

OSSTask * headTask = [client headObject:request];

[headTask continueWithBlock:^(id(OSSTask *) task) {
    if (!task.error) {
        NSLog(@"head object success!");
        OSSHeadObjectResult * result = task.result;
        NSLog(@"header fields: %@", result.httpResponseHeaderFields);
        for (NSString * key in result.objectMeta) {
            NSLog(@"ObjectMeta: %@ - %@", key, [result.objectMeta objectForKey:key]);
        }
    } else {
        NSLog(@"head object failed, error: %@", task.error);
    }
    return nil;
}];

```

10.8.4 List objects

This topic describes how to list all objects in a bucket.

Use the following code to list all objects in a bucket:

```

OSSGetBucketRequest * getBucket = [OSSGetBucketRequest new];
getBucket.bucketName = @"<bucketName>";

// Set optional parameters. For more information, see
// https://docs.aliyun.com/#/pub/oss/api-reference/bucket/GetBucket.
// getBucket.marker = @"";
// getBucket.prefix = @"";
// getBucket.delimiter = @"";

OSSTask * getBucketTask = [client getBucket:getBucket];

[getBucketTask continueWithBlock:^(id(OSSTask *) task) {

```

```

        if ( ! task . error ) {
            OSSGetBucketResult * result = task . result ;
            NSLog (@" get bucket success !") ;
            for ( NSDictionary * objectInfo in result . contents
        ) {
                NSLog (@" list object : %@", objectInfo );
            }
        } else {
            NSLog (@" get bucket failed , error : %@", task . error
        );
        }
        return nil ;
    }];

```

The following table lists the configurable parameters and their descriptions returned by the list operation:

Name	Description
delimiter	The delimiter used to separate object names. CommonPrefixes specify a set of substrings that start with a specified prefix and end with the first specified delimiter.
marker	The initial object in the list.
maxkeys	The maximum number of objects that can be listed. The default value is 100. The maximum value is 1,000.
prefix	The prefix that must be included in the returned objects. Note that if you use a prefix to make a query, the returned key values still contain the prefix.

10.8.5 Copy an object

You can copy an object from a bucket (source bucket) to another bucket (destination bucket) within the same region.

Use the following code to copy an object:

```

OSSCopyObjectRequest * copy = [ OSSCopyObjectRequest new ];
copy . bucketName = @"< bucketName >";
copy . objectKey = @"< objectKey >";
copy . sourceCopyFrom = [ NSString stringWithFormat :@"%/%@/
%@", @"< bucketName > , @"< objectKey_ copyFrom >"];

OSSTask * task = [ client copyObject : copy ];

[ task continueWithBlock :^ id ( OSSTask * task ) {
    if ( ! task . error ) {
        // ...
    }
}

```

```
        return nil ; s
    }];
```

**Notice:**

- The source and destination objects must belong to the same region.
- During the copy operation, if the source and destination objects have the same IP address, you can modify the Object Meta of the existing objects.
- You cannot use CopyObject to copy an object whose size exceeds 1 GB. To copy an object larger than 1 GB, use multipart upload.

10.8.6 Delete an object

This topic describes how to delete an object.

**Warning:**

Deleted objects cannot be recovered. Use caution when deleting objects.

Use the following code to delete an object:

```
OSSTDeleteObjectRequest * delete = [ OSSTDeleteObjectRequest
    new ];
delete . bucketName = @"< bucketName >";
delete . objectKey = @"< objectKey >";

OSSTask * deleteTask = [ client deleteObject : delete ];

[ deleteTask continueWithBlock:^(id ( OSSTask * task ) {
    if (! task . error ) {
        // ...
    }
    return nil ;
}]);

// [ deleteTask waitUntilFinished ];
```

**Note:**

To delete the object, you must have the write permission on the bucket where the object resides.

10.9 Data security

The iOS SDK provides data integrity checks to secure data during uploads and downloads.

Data integrity checks during file uploads

Due to the complexities in mobile network environments, an error can occur when data is transmitted between the client and the server. The OSS iOS SDK provides end-to-end data integrity checks based on MD5 and CRC64 values.

- MD5 verification

You must provide the Content-MD5 value of the file during the upload. The OSS server performs the MD5 verification after the file is uploaded. The upload can succeed only when the MD5 value of the file received by the OSS server is equal to the MD5 value provided before the upload. In this way, the integrity of uploaded data is guaranteed.

```
OSSPutObjectRequest * request = [OSSPutObjectRequest new];
request.bucketName = BUCKET_NAME;
...
request.contentMd5 = [OSSUtil fileMD5String:filepath];
```

- CRC

Compared with MD5 values, CRC64 values can be calculated during file uploads.

```
// Construct an upload request.
OSSPutObjectRequest * request = [OSSPutObjectRequest new];
request.bucketName = OSS_BUCKET_PRIVATE;
//....
request.crcFlag = OSSRequest_CRCOpen;
// After the CRC is enabled, If data inconsistency
// is found during the upload, OSSClientErrorCodeInvalidCRC
// is thrown.
OSSTask * task = [_client putObject:request];
[[task continueWithBlock:^(id(OSSTask * task) {
    // If the CRC fails, an error is reported.
    XCTAssertNil(task.error);
    return nil;
}]] waitUntilFinished];
```

Data integrity checks during file downloads

The OSS iOS SDK provides end-to-end data integrity checks based on CRC64 values.

When you read a downloaded data stream, the data integrity check is automatically performed if the CRC is enabled.

Use the following code to enable the CRC:

```
OSSGetObjectRequest * request = [ OSSGetObjectRequest new ];
request.bucketName = ... ;
// Enable the CRC .
request.crcFlag = OSSRequest CRCOpen ;

OSSTask * task = [ testProxyClient getObject : request ];

[[ task continueWithBlock:^(id ( OSSTask * task ) {
    // If the CRC is enabled and a data error occurs
    during transmission , OSSClientErrorCodeInvalidCRC is
    thrown .
    XCTAssertNil ( task.error );
    return nil ;
}]] waitUntilFinished ;
```

Note : If the onReceivedData block field is set , you must check the consistency of CRC64 values after the CRC is enabled . Example :

```
OSSGetObjectRequest * request = [ OSSGetObjectRequest new ];
request.bucketName = ....
request.crcFlag = OSSRequest CRCOpen ;
....

__block uint64_t localCrc64 = 0 ;
// If the onReceivedData block field is set ,
NSMutableData * receivedData = [ NSMutableData data ];
request.onReceivedData = ^( NSData * data ) {
    if ( data )
    {
        NSMutableData * mutableData = [ data mutableCopy ];
        void * bytes = mutableData.mutableBytes ;
        localCrc64 = [ OSSUtil crc64ecma : localCrc64 buffer :
        bytes length : data.length ];
        [ receivedData appendData : data ];
    }
};

__block uint64_t remoteCrc64 = 0 ;
OSSTask * task = [ _client getObject : request ];
[[ task continueWithBlock:^(id ( OSSTask * task ) {
    XCTAssertNil ( task.error );
    OSSGetObjectResult * result = task.result ;
    if ( result.remoteCRC64ecma )
    {
        NSScanner * scanner = [ NSScanner scannerWithString :
        result.remoteCRC64ecma ];
        [ scanner scanUnsignedLongLong :& remoteCrc64 ];
        if ( remoteCrc64 == localCrc64 )
        {
            NSLog (@" CRC64 verification succeeded !"); ;
        }
        else
        {
            NSLog (@" CRC64 verification failed !"); ;
        }
    }
    return nil ;
```

```
}] waitUntilFinished];
```

10.10 Signed URLs

The SDK supports the signed URLs with a validity period or public URLs so that the URL can be forwarded to a third party for authorized access.

Generated a signed private resource URL with a validity period

If the bucket or object ACL is Private, you must call the following API to obtain the signed URL:

```
NSString * constrainedURL = nil ;

// Generate the signed URL .
OSSTask * task = [ client presignConstrainedURLWithBucketName :@"< bucket name >"
                                     withObjectKey :@"< object key >"
                                     withExpirationInterval : 30
                                     * 60 ];
if (! task . error ) {
    constrainedURL = task . result ;
} else {
    NSLog (@" error : %@", task . error );
}
```

Generated a signed public URL

If the bucket or object ACL is Public Read, you must call the following API to obtain the URL of the publicly accessible object:

```
NSString * publicURL = nil ;

// Generate the signed URL .
task = [ client presignPublicURLWithBucketName :@"< bucket name >"
        withObjectKey :@"< object key >"];
if (! task . error ) {
    publicURL = task . result ;
} else {
    NSLog (@" sign url error : %@", task . error );
}
```



```
}
```

10.11 Authorized access

The iOS SDK provides STS authentication and self-signed modes to secure mobile devices.

STS authentication mode

You can use Alibaba Cloud STS to authorize temporary access to OSS. STS is a Web service that provides a temporary access token to a cloud computing user. You can use STS to assign a third-party application or a federated user (you can manage the user ID) an access credential with a custom validity period and permissions. The access credential in the application is called a federation token. The third-party application or federated user can use the access credential to directly call the Alibaba Cloud service APIs or use the Alibaba Cloud service SDKs to access the cloud service APIs.

- STS benefits:
 - You do not need to expose your long-term key (AccessKey) to a third-party application. You need only to generate an access token and send the access token to the third-party application. You can customize the access permissions and validity period of this token.
 - The access token automatically becomes invalid after expiration.

- How to use STS

1. Log on to AppServer as an application user.

You can manage your application ID. You can customize the ID management system or use an external Web account or open ID. For each valid application user, AppServer can precisely define their minimum access permissions.

2. AppServer requests a security token from STS.

- a. Before calling STS, AppServer needs to determine the minimum access permissions (described in the policy syntax) of the application user and the expiration time of the authorization.
- b. Then, you can call the [AssumeRole](#) of STS to obtain the security token. For more information about how to manage and use roles, see [Role management](#) in RAM User Guide.

3. STS returns a valid access credential to AppServer. The access credential contains a security token, a temporary AccessKey pair (AccessKey ID and AccessKey Secret), and the expiration time.

4. AppServer returns the access credential to ClientApp.

ClientApp caches this credential. When the credential becomes invalid, ClientApp must request a new valid access credential from AppServer. For example, if the access credential is valid for one hour, ClientApp can request AppServer to update the access credential every 30 minutes.

5. ClientApp uses the locally cached access credential to request Alibaba Cloud service APIs. Cloud services detect the STS access credential. STS verifies the credential, and correctly responds to user requests.

**Notice:**

To use this authentication mode, you need to first activate Alibaba Cloud RAM, as shown in [Alibaba Cloud RAM](#).

- Set an STS token directly

Before you log on to ClientApp, you can use a certain method (such as sending network requests from your AppServer) to obtain an STS token. Then, use the token to initialize the SDK. If this method is adopted, you must take note of the

validity period of the STS token. When the STS token is about to expire, you must manually update the STS token for the SDK.

The initialization code is as follows:

```
id < OSSCredentialProvider > credential = [[ OSSStsTokenCredentialProvider alloc ] initWithAccessKeyId:@"<
StsToken.AccessKeyId>" secretKeyId:@"<StsToken.SecretKeyId>" securityToken:@"<StsToken.SecurityToken>"];
client = [[ OSSClient alloc ] initWithEndpoint: endpoint credentialProvider: credential];
```

When you notice that the token is about to expire, you can rebuild a new OSSClient or use the following method to update the CredentialProvider field value:

```
id < OSSCredentialProvider > credential = [[ OSSStsTokenCredentialProvider alloc ] initWithAccessKeyId:@"<
StsToken.AccessKeyId>" secretKeyId:@"<StsToken.SecretKeyId>" securityToken:@"<StsToken.SecurityToken>"];
client = [[ OSSClient alloc ] initWithEndpoint: endpoint credentialProvider: credential];
```

- Implement a callback to obtain and update an STS token

If you want the SDK to automatically help you manage token updates, you must configure the SDK to obtain the token. To implement a callback, you must obtain a federation token (STS token) and send the token to the SDK. The SDK signs the token. When the token needs to be updated, the SDK calls the callback to obtain the token, as shown in the following figure.

```
id < OSSCredentialProvider > credential = [[ OSSFederationCredentialProvider alloc ] initWithFederationTokenGetter:^(OSSFederationToken *) {
    // Here you need to obtain a federation token, construct it into the OSSFederationToken object, and send the object to the SDK.
    // If the token fails to be obtained for some reason, nil is returned.
    OSSFederationToken * token;
    // The following code is used to obtain the token from AppServer:
    ...
    return token;
}];
client = [[ OSSClient alloc ] initWithEndpoint: endpoint credentialProvider: credential];
```



Note:

Additionally, if you use other means to obtain all fields required by the token, you can directly send the token to the SDK and implement the callback. In this case, you need to manually update the token and then reset the value of `OSSCredentialProvider` of the `OSSClient` instance.

Example:

Assume that your server address is `http://localhost:8080/distribute-token.json`.

The following data is returned after the address is accessed:

```
{
  " StatusCode ": 200 ,
  " AccessKeyId ":" STS . iA645eTOXE qP3cg3VeHf ",
  " AccessKeySecret ":" rV3VQrpFQ4 BsyHSAvi5N VLpPIVffDJ
v4LojUBZCf ",
  " Expiration ":" 2015 - 11 - 03T09 : 52 : 59Z ",
  " SecurityToken ":" CAES7QIIAR KAAZPlqaN9 ILiQZPS + JDks /
GSZN45RLx4 YS / p30gaUC + oJl3XS1bJ7 StKpQ ...."}

```

You can use the following code to implement the `OSSFederationTokenCredentialProvider` instance:

```
id < OSSCredentialProvider > credential 2 = [[ OSSFederationTokenCredentialProvider alloc ] initWithFederationTokenGetter:^( OSSFederationToken * {
    // Construct a request to access your AppServer .
    NSURL * url = [ NSURL URLWithString:@" http ://
localhost : 8080 / distribute - token . json "];
    NSURLRequest * request = [ NSURLRequest requestWith
hURL : url ];
    OSSTaskCompletionSource * tcs = [ OSSTaskCompletionSource
pletionSource taskCompletionSource ];
    NSURLSession * session = [ NSURLSession sharedSession ];
    // Send the request .
    NSURLSessionTask * sessionTask = [ session
dataTaskWithRequest : request
completion
Handler:^( NSData * data , NSURLResponse * response ,
NSError * error ) {
    if ( error )
    {
        [ tcs
setError : error ];
        return ;
    }
    [ tcs
setResult : data ];
    [ sessionTask resume ];
    // Wait until the response is returned .
    [ tcs . task waitUntilFinished ];
    // Parse the result .
    if ( tcs . task . error ) {
        NSLog(@" get token error : %@", tcs . task . error
);
        return nil ;
    }
}];

```

```

    } else {
        // The returned data is in the JSON format
        and you must parse it to obtain all fields of
        the token .
        NSDictionary * object = [ NSJSONSerialization
        JSONObjectWithData : tcs . task . result

options : kNilOptions

error : nil ];
        OSSFederationToken * token = [ OSSFederationToken
        new ];
        token . tAccessKey = [ object objectForKey:@"
        AccessKeyId "];
        token . tSecretKey = [ object objectForKey:@"
        AccessKeySecret "];
        token . tToken = [ object objectForKey:@"
        SecurityToken "];
        token . expirationTimeInGMTFormat = [ object
        objectForKey:@" Expiration "];
        NSLog(@" get token :%@", token );
        return token ;
    }
}];

```

Self-signed mode

```

id < OSSCredentialProvider > credential = [[ OSSCustomSignatureCredentialProvider alloc] initWithImplementedSignature:^ NSString *( NSString * contentToSign, NSError * __autoreleasing * error ) {
    // Here you need to use the signature algorithm
    specified by OSS to sign a string, add the
    AccessKey ID to the signed string, and send the
    signed string to AppServer .
    // Typically, you can upload the string to
    AppServer . AppServer returns the signature .
    // If the string fails to be signed, error
    information is recorded, and nil is returned .
    NSString * signature = [ OSSUtil calculateBase64SHA1WithData :
    contentToSign withSecret:@"< your accessKeySecret >"]; //
    The ossutil tool in the SDK is used to complete
    the local signature . We recommend that you use
    AppServer to implement the remote signature .
    if ( signature != nil ) {
        * error = nil ;
    } else {
        * error = [ NSError errorWithDomain:@"< your domain
        >" code :- 1001 userInfo:@"< your error info >"];
        return nil ;
    }
    return [ NSString stringWithFormat:@" OSS %@:%@", @"< your
    accessKeyId >", signature ];
}];

client = [[ OSSClient alloc] initWithEndpoint : endpoint
credentialProvider : credential ];

```



Notice:

For STS authentication and self-signed modes, you must ensure that the response is correct after you implement the callback. Therefore, if you send a network request to AppServer to obtain the token and signature, we recommend that you call the synchronous APIs of the network library or perform the conversion from async to sync. The callback is run in the subthread of the request from the SDK, which does not block the main thread.

10.12 IMG

Image Processing (IMG) is a secure, cost-effective, and highly reliable image processing service provided by OSS. IMG is capable of processing large amounts of images. After uploading source images to OSS, you can use Representational State Transfer (RESTful) APIs to process the images anytime, anywhere, and on any Internet device. You can create your own IMG service in OSS.

For more information about IMG, see [OSS Image Processing Guide](#).

Basic features

IMG has the following features:

- [Resize images](#)
- [Crop images](#)
- [Rotate images](#)
- [Apply effects](#)
- [Convert formats](#)
- [Add watermarks](#): allows you to add image, text, or image-text watermarks to another image.

How to use IMG

IMG uses the standard HTTP GET request. You can configure IMG parameters in QueryString of a URL.

If the ACL of an image object is Private, only authorized users are allowed access.

- Anonymous access

```
OSSTask * task = [ _client presignPublicURLWith BucketName
                  : _publicBucketName
                  withObject Key:@"hasky.jpeg "
```

```
        withParameters:@{@"x-oss-process": @"image /
resize , w_50 "}]];
```

- **Authorized access**

To use IMG in the SDK, you can call the `request . setxOssProcess ()` method to set IMG parameters when downloading an image. Example:

```
OSSGetObjectRequest * request = [ OSSGetObjectRequest new
];
request . bucketName = @"< bucketName >";
request . objectKey = @" example . jpg ";
// Process the image .
request . xOssProcess = @" image / resize , m_lfit , w_100 ,
h_100 ";
OSSTask * getTask = [ client getObject : request ];
[ getTask continueWithBlock : ^ id ( OSSTask * task ) {
    if ( ! task . error ) {
        NSLog (@" download image success !" ) ;
        OSSGetObjectResult * getResult = task . result ;
        NSLog (@" download image data : %@", getResult .
downloadedData );
    } else {
        NSLog (@" download object failed , error : %@", task
. error );
    }
    return nil ;
}];
// [ getTask waitUntilFinished ];
// [ request cancel ];
```



Note:

To perform other processing operations on the image, you need only to replace relevant parameters in `request . setxOssProcess ()`.

- **Access with the SDK**

```
OSSGetObjectRequest * request = [ OSSGetObjectRequest new
];
request . bucketName = @" bucket - name "; // Set the
bucket name .
request . objectKey = @" image - name "; // Set the
image name .
request . xOssProcess = @" image / resize , m_lfit , w_100 ,
h_100 "; // Process the image .
OSSTask * task = [ ossClient getObject : request ];
```

Implement persistent storage for a processed image

Use the following code to store a processed image in a bucket for persistent storage:

```
OSSImagePersistRequest * request = [ OSSImagePersistRequest
new ];
request . fromBucket = _privateBucketName ;
request . fromObject = OSS_IMAGE_KEY ;
request . toBucket = _privateBucketName ;
request . toObject = @" image_persist_key ";
```

```

Request . action = @ " image / resize , w_100 "; // Process the
image .
// request . action = @" resize , w_100 ";

[[[ ossClient imageActionPersist : request ] continueWithBlock
:^ id _Nullable ( OSSTask * _Nonnull task ) {

    return nil ;
}] waitUntilFinished ];

```

10.13 Exception handling

There are two types of exceptions in the iOS SDK: `ClientError` and `ServerError`.

`ClientError` indicates parameter and network errors. `ServerError` indicates abnormal responses from the OSS server.

Error type	Error domain	Code	UserInfo
ClientError	com.aliyun.oss.clientError	0	<code>OSSClientErrorCodeNetworkingFailWithResponseCode0</code> , indicating that the connection is abnormal.
		1	<code>OSSClientErrorCodeSignFailed</code> , indicating that the signature failure occurs.
		2	<code>OSSClientErrorCodeFileCantWrite</code> , indicating that the object fails to be written to the server.
		3	<code>OSSClientErrorCodeInvalidArgument</code> , indicating that the parameter is invalid.
		4	<code>OSSClientErrorCodeNilUploadid</code> , indicating that the upload ID for the resumable upload task fails to be obtained.
		5	<code>OSSClientErrorCodeTaskCancelled</code> , indicating that the task is cancelled.
		6	<code>OSSClientErrorCodeNetworkError</code> , indicating that the network is abnormal.
		7	<code>OSSClientErrorCodeCannotResumeUpload</code> , indicating that the resumable upload task fails.
		8	<code>OSSClientErrorCodeNotKnown</code> , indicating that an unknown error occurs.

Error type	Error domain	Code	UserInfo
ServerError	com.aliyun.oss.serverError	(-1 * httpResponseBodyCode)	The directory obtained by parsing the corresponding XML file.

11 Node.js

11.1 Installation

This topic describes how to install Node.js. Demos in the official documents are based on SDK 6.X. For SDKs of earlier versions, see [Document for 5.X SDK](#). For more information about upgrade your SDK to 6.X, see [Upgrade document](#).

- [GitHub](#)
- [API document](#)
- [ChangeLog](#)

Environment preparations

- Prerequisites
 - Activate OSS service. If you haven't activated or do not know about Alibaba Cloud OSS, log on to the OSS product page for more information.
 - The Alibaba Cloud account AccessKey gives you permission to access all APIs, thus, we recommend that you follow Alibaba Cloud Best Practices for creating an AccessKeyId and an AccessKeySecret. If you deploy the service on the server, you can use an RAM sub-account or STS for API access or daily O&M management and control. If you deploy the service on the client, we recommend that you use STS for API access. For more information, see [Access control](#).
- Environment requirements

The OSS JavaScript SDK is built on the [Node.js](#) environment.

Usage methods

The OSS JavaScript SDK supports both synchronous and asynchronous use.

- Synchronous mode: Synchronize the asynchronous mode based on `async` and `wait`.
- Asynchronous mode: Similar to the callback mode. The API returns the [Promise](#), uses the `.then()` method to process the returned result and uses the `.catch()` method to process errors.

The `new OSS()` method is used to create a client in both the synchronous mode and asynchronous mode.

Next is to name examples to upload an object first and then download the object respectively.

- **Synchronous mode**

```
let client = new OSS (...);
async function put () {
  try {
    // object indicates the uploaded object name ,
    localfile indicates a local file or file path .
    let r1 = await client . put (' object ', ' localfile ');
    console . log (' put success : % j ', r1 );
    let r2 = await client . get (' object ');
    console . log (' get success : % j ', r2 );
  } catch ( e ) {
    console . error (' error : % j ', err );
  }
}
Put ();
```

- **Asynchronous mode**

```
let client = new OSS (...);

// object indicates the uploaded object name , localfile
// indicates a local file or file path .
client . put (' object ', ' localfile '). then ( function ( r1 ) {
  console . log (' put success : % j ', r1 );
  return client . get (' object ');
}). then ( function ( r2 ) {
  console . log (' get success : % j ', r2 );
}). catch ( function ( err ) {
  console . error (' error : % j ', err );
});
```

Installation

- **Use Node.js**

Supported Node.js versions:

Node.js later than 8.0.0 is supported. Use ali-oss 4.x in Node.js environment earlier than 8.0.0.

Use [npm](#) to install the SDK:

```
npm install ali - oss
```

Use it in your project:

```
let OSS = require (' ali - oss ');

let client = new OSS ({
  region : '< oss region >',
  // The Alibaba Cloud account AccessKey gives you
  permission to access all APIs , so we recommend to
  follow Alibaba Cloud Best Practices for deploying the
```

```

    service on the server using an RAM sub-account
    or STS, or deploying the service on the client
    using STS.
    accessKeyId : '< Your accessKeyId >',
    accessKeySecret : '< Your accessKeySecret >',
    bucket : '< Your bucket name >'
  });

```

- If you face any network problems when using npm, you can use the npm image provided by Taobao: [cnpm](#).

11.2 Options

This topic describes how to configure options.

OSS(options)

- [accessKeyId] {String}: accessKeyId you create on Alibaba Cloud console website
- [accessKeySecret] {String}: accessKeySecret you create
- [bucket] {String}: the default bucket you want to access. If you do not have any bucket, use putBucket() to create one first.
- [endpoint] {String}: oss region domain. It takes priority over region.
- [region] {String}: the bucket data region location. The default value is oss-cn-hangzhou.
- [internal] {Boolean}: access OSS with Alibaba Cloud internal network or not. The default value is false. If your servers are running on Alibaba Cloud too, you can set it to true to reduce costs.
- [secure] {Boolean}: instruct OSS client to use HTTPS (secure: true) or HTTP (secure: false) protocol. For more information, click [here](#).
- [timeout] {String|Number}: instance level timeout for all operations. The default value is 60s.

```

var oss = require('ali-oss');

var store = oss({
  accessKeyId: 'your access key',
  accessKeySecret: 'your access secret',
  bucket: 'your bucket name',
  region: 'oss-cn-hangzhou'
});

```

```
});
```

11.3 Quick start

This topic describes how to quickly perform OSS operations (such as view the bucket list or upload objects) in the Node.js environment.



Note:

To facilitate changes, a file named `app.js` is created to demonstrate the code used to perform the following operations.

View the bucket list

Add the following code to the end of `app.js`, and then call the `listBuckets` interface to view the bucket list.

```
async function listBuckets () {
  try {
    let result = await client.listBuckets();
  } catch (err) {
    console.log(err);
  }
}

listBuckets();
```

You can run `node app.js` to view the result.

For more information about bucket-related APIs, see [GitHub](#).

View the object list

Modify the `app.js` file as follows and call the `list` interface to view the object list.

```
client.useBucket('Your bucket name');
async function list () {
  try {
    let result = await client.list({
      'max-keys': 5
    });
    console.log(result);
  } catch (err) {
    console.log(err);
  }
}
list();
```

You can run `node app.js` to view the result.

Upload an object

Modify the `app.js` file as follows and call the `put` interface to upload a single object.

```
client . useBucket ( ' Your    bucket    name ' );

async  function  put  () {
  try {
    let  result  =  await  client . put ( ' object - name ', ' local
file ' );
    console . log ( result );
  } catch ( err ) {
    console . log ( err );
  }
}

put ();
```

Download an object

Modify the `app.js` file as follows and call the `get` interface to download a single object.

```
async  function  get  () {
  try {
    let  result  =  await  client . get ( ' object - name ' );
    console . log ( result );
  } catch ( err ) {
    console . log ( err );
  }
}

get ();
```

Delete an object

Modify the `app.js` file and call the `delete` interface to delete an object.

```
async  function  delete  () {
  try {
    let  result  =  await  client . delete ( ' object - name ' );
    console . log ( result );
  } catch ( err ) {
    console . log ( err );
  }
}

delete ();
```

For more information about object-related interfaces, see [GitHub](#).

11.4 Initialization

This topic describes how to initialize the Node.js SDK.

Create an object named `app.js` and write the following content:

```
let OSS = require('ali-oss');

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >'
});
```

The `region` field specifies the region that is set when you apply for OSS, such as `oss-cn-hangzhou`. For the complete list of regions, see [Endpoints and regions](#).



Note:

If the desired endpoint is not included in the preceding list, you can use the following parameters to specify the endpoint:

- **internal:** You must set the `region` field if the internal parameter is used. If you set `internal` to `true`, you can access the intranet nodes.
- **secure:** You must set the `region` field if the secure parameter is used. If you set `secure` to `true`, use HTTPS for access.
- **endpoint:** If you specify the `endpoint` field, such as `http://oss-cn-hangzhou.aliyuncs.com`, you do not need to set the `region` field. The `endpoint` field value can be a domain that uses HTTPS or an IP address.
- **cname:** You must set the `endpoint` field if the cname parameter is used. If you set `cname` to `true`, the `endpoint` field value is used as the custom domain that is bound to a bucket.
- **bucket:** If the `bucket` field value is not specified, you must first call `useBucket` to perform operations on the object that belongs to the bucket. You need to call this API only once.
- **timeout:** specifies the timeout period to access OSS APIs. The default value is 60 seconds.

11.5 Manage a bucket

A bucket is a namespace on OSS and a management entity for billing, access control, logging, and other advanced features.

View all buckets

You can use the `listBucket` interface to list all the buckets belonging to the current user. You can specify the `prefix` parameter to list all buckets whose names contain the specified prefix.

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >'
});

async function listBuckets () {
  try {
    const result = await client.listBuckets();
    console.log(result);
    const result2 = await client.listBuckets({
      prefix: 'prefix',
    });
    console.log(result2);
  } catch (err) {
    console.log(err);
  }
}

listBuckets();
```

Create a bucket

You can create a bucket using the `putBucket` interface. You require to specify the bucket name:

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >'
});

// putBucket
async function putBucket () {
  try {
    const result = await client.putBucket('your bucket name');
    console.log(result);
  } catch (err) {
    console.log(err);
  }
}
```



```
}  
  
putBucket ();
```

**Notice:**

- For bucket naming conventions, see [OSS Basic Concepts](#).
- This is because the name of a bucket is globally unique, make sure your bucket name does not match with the other users' .

Delete a bucket

You can use the `deleteBucket` interface to delete a bucket. You require to specify the bucket name:

```
let OSS = require ('ali-oss');  
  
let client = new OSS ({  
  region : '< Your region >',  
  accessKeyId : '< Your AccessKeyId >',  
  accessKeySecret : '< Your AccessKeySecret >'  
});  
  
async function deleteBucket () {  
  try {  
    const result = await client.deleteBucket ('your  
bucket name');  
    console.log (result);  
  } catch (err) {  
    console.log (err);  
  }  
}  
  
deleteBucket ();
```

**Notice:**

- If the bucket to be deleted has objects in it, you must delete the objects first.
- If the bucket to be deleted has unfinished upload requests, you must cancel the requests first using the `listUploads` and `abortMultiPartUpload` interfaces.

Bucket access permissions

You can set the ACL policy of a bucket to allow or prohibit anonymous reads/writes to the bucket. For more information on ACL, see [ACL](#).

- Obtain the ACL of a bucket

You can use `getBucketACL` to view the ACL policy of a bucket:

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >'
});

async function getBucketACL () {
  try {
    const result = await client.getBucketACL('
luozhang002');
    console.log(result);
  } catch (err) {
    console.log(err);
  }
}

getBucketACL();
```

- Set the ACL policy of a bucket

You can use `putBucketACL` to configure the ACL policy of a bucket:

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >'
});

async function putBucketACL () {
  try {
    const result = await client.putBucketACL('bucket
name', 'public-read');
    console.log(result);
  } catch (err) {
    console.log(err);
  }
}

putBucketACL();
```

11.6 Upload objects

11.6.1 Overview

Objects are the basic unit for data operations in OSS. The OSS Node.js SDK provides the following methods to upload objects:

- **Local file upload:** supports the upload of a file up to 5 GB in size.

- **Upload from local memory:** supports the upload of a file up to 5 GB in size.
- **Streaming upload:** supports any file sizes.
- **Multipart upload:** supports the upload of a file up to 48.8 TB in size. This method is suitable for the upload of large files.
- **Resumable upload:** supports the upload of a file up to 48.8 TB in size. This method is suitable for the upload of large files. It supports concurrent upload and you can define the size of each part.

**Note:**

For more information about the scenarios of each upload method, see the "Upload files" section in OSS Developer Guide.

11.6.2 Upload a local file

This topic describes how to upload a local file to OSS.

**Note:**

The following sample code uses the catch syntax. Learn ES6 Promise and ES6 Async/Await on your own. For more information about how to use the SDK, see [#unique_686](#).

You can use put to upload a local file to OSS:

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function put () {
  try {
    let result = await client.put('object-name', 'local-file');
    console.log(result);
  } catch (e) {
    console.log(e);
  }
}
```

```
put ();
```

11.6.3 Upload from local memory

This document describes how to upload content from local memory to OSS.



Note:

The following sample code uses the catch syntax. Learn ES6 Promise and ES6 Async/Await on your own. For more information about how to use the SDK, see [#unique_686](#).

You can use `put` to upload content from local memory to OSS:

```
let OSS = require ('ali-oss');

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function putBuffer () {
  try {
    let result = await client.put ('object-name', new
Buffer ('hello world'));
    console.log (result);
  } catch (e) {
    console.log (e);
  }
}

putBuffer ();
```

11.6.4 Streaming upload

This topic describes how to perform streaming upload.



Note:

The following sample code uses the catch syntax. Learn ES6 Promise and ES6 Async/Await on your own. For more information about how to use the SDK, see [#unique_686](#).

You can use `putStream` to upload the content of a stream. You can set `stream` to specify the types of `ReadableStream`, such as file stream and network stream. When you use `putStream`, the SDK initiates an HTTP PUT request for chunked

encoding. If contentLength is specified in options, chunked encoding is not used.

```
let OSS = require('ali-oss');
let fs = require('fs');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function putStream () {
  try {
    // use 'chunked encoding'
    let stream = fs.createReadStream('local - file');
    let result = await client.putStream('object - name',
    stream);
    console.log(result);

    // don't use 'chunked encoding'
    let stream = fs.createReadStream('local - file');
    let size = fs.statSync('local - file').size;
    let result = await client.putStream(
    'object - name', stream, { contentLength: size });
    console.log(result);
  } catch (e) {
    console.log(e)
  }
}

putStream();
```

11.6.5 Multipart upload

This topic describes how to use multipart upload.



Note:

The following sample code uses the catch syntax. Learn ES6 Promise and ES6 Async/Await on your own. For more information about how to use the SDK, see [#unique_686](#).

To upload a large file, you can use `multipartUpload`. The advantage of multipart upload is to divide the upload request of a large file into multiple small requests and run each small request. This method guarantees that when some of the requests fail, you need only to upload the failed parts instead of the entire file. We recommend that you use multipart upload for files that are larger than 100 MB.

If `Connection TimeoutError` occurs during the use of `multipartUpload`, the business side must process the timeout logic on its own. You can scale in the part

size, prolong the timeout period, retry the request, or analyze the specific cause by capturing `Connection TimeoutErr` or `.`

Relevant parameters are described as follows:

- `name {String}`: specifies the object name.
- `file {String|File}`: specifies the file path or the HTML5 file.
- `[options] {Object}`: specifies optional parameters.
 - `[checkpoint] {Object}`: specifies the checkpoint information. If this parameter is set, the upload starts from the recorded checkpoint information. If this parameter is not set, the entire file needs to be uploaded.
 - `[parallel] {Number}`: specifies the number of parts that are to be uploaded simultaneously.
 - `[partSize] {Number}`: specifies the size of each part.
 - `[progress] {Function}`: specifies the async function mode. The callback function involves three parameters:
 - `(percentage {Number})`: specifies the percentage of the upload progress (or a decimal from 0 to 1).
 - `checkpoint {Object}`: specifies the checkpoint information.
 - `res {Object}`: specifies the response to a part upload request.
 - `[meta] {Object}`: specifies the Object Meta you can set in the header. The prefix of Object Meta in the header is `x-oss-meta-`.
 - `[mime] {String}`: specifies the `Content-Type` value you can define in the header.
 - `[headers] {Object}`: specifies other field information in the header. For more information, see [RFC 2616](#).
 - `'Cache-Control'`: specifies the cache mechanism that is added to HTTP request or response headers. The cache mechanism is implemented through commands. Example: `Cache-Control: public, no-cache`.
 - `'Content-Disposition'`: specifies the form in which the reply content is displayed. The reply content can be displayed on a Web page, or downloaded

and saved locally as an attachment. Example: `Content - Disposition : somename .`

- **'Content-Encoding':** compresses the data of a specific medium type, such as

`Content - Encoding : gzip .`

- **'Expires':** specifies the validity period, such as `Expires : 3600000 .`

Use the following code for multipart upload:

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function multipartUpload() {
  try {
    let result = await client.multipartUpload('object - name', 'local - file', {
      progress,
      meta: {
        year: 2017,
        people: 'test'
      }
    });
    console.log(result);
    let head = await client.head('object - name');
    console.log(head);
  } catch (e) {
    // Catch the timeout exception.
    if (e.code === 'Connection TimeoutError or ') {
      console.log("Woops, timeout exception!");
      // do Connection TimeoutError or operation
    }
    console.log(e)
  }
}
```

`progress` is a progress upload callback function that is used to obtain the upload progress. `progress` can be an async function:

```
const progress = async function(p) {
  console.log(p);
};
```

`meta` specifies Object Meta. You can use `head` to obtain Object Meta.

11.6.6 Resumable upload

You can use resumable upload to split a file you want to upload into several parts and upload them simultaneously. After all parts are uploaded, you can combine these parts into a complete object. Thus, you can complete uploading the entire object.



Note:

The following sample code uses the catch syntax. Learn ES6 Promise and ES6 Async/Await on your own. For more information about how to use the SDK, see [#unique_686](#).

For more information about resumable upload, see the "[Resumable upload](#)" section in OSS Developer Guide.

Multipart upload provides `progress` to facilitate making upload progress callback requests. The SDK uses the upload progress and the checkpoint information as parameters in the callback requests. To implement resumable upload, you can save the checkpoint information during the upload process. When an error occurs, you can pass the saved checkpoint information as a parameter to `multipartUpload`. The upload continues from where it failed earlier.

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

let checkpoint;
async function resumeUpload() {
  // Set the number of retry attempts to 5.
  for (let i = 0; i < 5; i++) {
    try {
      const result = await client.multipartUpload('object - name', filePath, {
        checkpoint,
        async progress (percentage, cpt) {
          checkpoint = cpt;
        },
      });
      console.log(result);
      break; // break if success
    } catch (e) {
      console.log(e);
    }
  }
}
```



```
resumeUpload ();
```

The preceding code saves the checkpoint information in the variable. If the program stops responding, the checkpoint information is lost. We recommend that you save the checkpoint information in a file. After the program restarts, the checkpoint information can be read from the file.

11.7 Download objects

11.7.1 Overview

The OSS Node.js SDK provides the following methods to download an object:

- [Download to a local file](#)
- [Download to local memory](#)
- [Streaming download](#)
- [HTTP-based download](#)

11.7.2 Download to a local file

This topic describes how to download an object from OSS to a local file.

You can use `get` to download an object from OSS to a local file:

```
let OSS = require ('ali-oss');

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : 'Your bucket name '
});

async function get () {
  try {
    let result = await client . get (' object - name ', ' local
- file ');
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

get ();
```

11.7.3 Download to local memory

This topic describes how to download an object to local memory.

You can use `get` to download an object from OSS to local memory:

```
let OSS = require ('ali-oss');

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function getBuffer () {
  try {
    let result = await client.get ('object-name');
    console.log (result.content);
  } catch (e) {
    console.log (e);
  }
}

getBuffer ();
```

11.7.4 Streaming download

If you have a large object to download or it is time-consuming to download the entire object at a time, you can use streaming download. Streaming download enables you to download part of the object content each time until you download the entire object.

When you use `getStream` to download an object, `ReadableStream` is returned and used to stream the object content.

```
let OSS = require ('ali-oss');
let fs = require ('fs');

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function getStream () {
  try {
    let result = await client.getStream ('object-name');
    console.log (result);
    let writeStream = fs.createWriteStream ('local-file');
    result.stream.pipe (writeStream);
  } catch (e) {
    console.log (e);
  }
}
```

```

    }
  }

  getStream ()

```

11.7.5 HTTP-based download

Aside from using the OSS Node.js SDK, you can directly use HTTP to download objects stored in OSS.

HTTP-based download supports browsers and command line interface (CLI) tools such as `wget` and `curl`. The object URL is generated through the SDK. Use `signatureUrl` to generate an HTTP-based address that can be used for downloads. The validity period of the URL is half an hour by default.

Use the following code to download an object from OSS over HTTP:

```

let OSS = require ('ali-oss');

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : 'Your bucket name '
});

let url = client . signatureUrl (' object - name ');
console . log ( url );

let url = client . signatureUrl (' object - name ', { expires
: 3600 });
console . log ( url );

// Generate the signed URL for PUT .
let url = client . signatureUrl (' object - name ', { method :
' PUT '});
console . log ( url );

```

11.8 Manage objects

11.8.1 Overview

You can use APIs to perform

the following operations on objects in a bucket:

- [Determine whether an object exists](#)
- [Manage Object Meta](#)
- [Lists all objects in a bucket](#)
- [List the objects and subfolders of a folder](#)
- [Copy an object](#)

- [Delete objects](#)

11.8.2 Determine whether an object exists

This topic describes how to determine whether an object exists.

Use `get` to determine whether an object exists:

```
client . get ( object ). then ( ( result ) => {  
    if ( result . res . status == 200 ) {  
        return true  
    }  
}). catch ( ( e )=> {  
    if ( e . code == ' NoSuchKey ' ) {  
        return false  
    }  
})
```



Note:

For more information about the use of `get`, see [Download an object](#).

11.8.3 Manage Object Meta

Object Meta contains HTTP headers and User Meta. For more information, see the "[Object Meta](#)" section in OSS Developer Guide.

When you upload a file to OSS, you can specify the information of certain properties in addition to the content. The property information is called Object Meta. Object Meta is stored together with its object during uploads, and returned with the object when the object is downloaded.

Object Meta is stored in HTTP headers during object uploads or downloads.

Therefore, Object Meta cannot contain complex characters in accordance with HTTP.



Note:

Object Meta can only contain simple characters in the ASCII format and cannot contain line breaks. The total size of Object Meta cannot exceed 8 KB.

You can set `meta` to specify Object Meta when you use `put`, `putStream`, or `multipartUpload`.

```
let OSS = require ( ' ali - oss ' )  
  
let client = new OSS ({  
    region : '< Your region >',  
    accessKeyId : '< Your AccessKeyId >',  
    accessKeySecret : '< Your AccessKeySecret >',  
    bucket : ' Your bucket name '  
});
```

```

async function put () {
  try {
    let result = await client . put (' object - name ', ' local
- file ', {
      meta : {
        year : 2016 ,
        people : ' mary '
      }
    });
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

put ();

```

You can also use `putMeta` to update Object Meta:

```

let OSS = require (' ali - oss ')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : ' Your bucket name '
});

async function putMeta () {
  try {
    let result = await client . putMeta (' object - name ', {
      meta : {
        year : 2015 ,
        people : ' mary '
      }
    });
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

putMeta ();

```

11.8.4 List all objects in a bucket

This topic describes how to list all objects in a bucket.

Use `list` to list all objects in the current bucket. Parameters are described as follows:

- **prefix:** only lists objects with the specified prefix.
- **marker:** only lists objects whose names come after the marker value.
- **delimiter:** separates object names. CommonPrefixes specify a set of substrings that start with a specified prefix and end with the first specified delimiter.

- **max-keys:** the maximum number of objects to be returned.

Use the following code to list all objects in a bucket:

```
let OSS = require ('ali-oss');
let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : 'Your bucket name '
});
async function list () {
  {
    // If no parameter is specified, a maximum of 1,000
    // objects can be returned by default.
    let result = await client.list ();
    console.log (result);
    // Continue to list objects based on the nextMarker
    // field value.
    if (result.isTruncated) {
      let result = await client.list ({
        marker : result.nextMarker
      });
    }
    // List all objects that are prefixed with "my-."
    let result = await client.list ({
      prefix : 'my-'
    });
    console.log (result);
    // List all objects whose names are prefixed with "
    // my-" and come after the "my-object" marker.
    let result = await client.list ({
      prefix : 'my-',
      marker : 'my-object'
    });
    console.log (result);
  } catch (e) {
    console.log (e);
  }
}
list ();
```

11.8.5 List the objects and subfolders of a folder

This topic describes how to list objects and subfolders of a folder.

OSS is an object-based storage service. It has no directories. All objects stored in a bucket are identified by unique object keys in a non-hierarchical structure. OSS provides CommonPrefixes for object names. You can easily simulate the directory structure and store objects in different directories like traditional files. For more information about the concept of CommonPrefixes, see [List objects](#).

Assume that a bucket contains the following objects:

```
foo / x
foo / y
foo / bar / a
```

```
foo / bar / b
foo / hello / C / 1
foo / hello / C / 2
...
foo / hello / C / 9999
```

To list the objects and subfolders in a folder, call the `listDir` function:

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function listDir(dir) {
  let result = await client.list({
    prefix: dir,
    delimiter: '/'
  });

  result.prefixes.forEach(function(subDir) {
    console.log('SubDir: %s', subDir);
  });
  result.objects.forEach(function(obj) {
    => console.log('Object: %s', obj.name);
  });
}
```

A similar output is displayed:

```
> await listDir('foo /')
=> SubDir: foo / bar /
    SubDir: foo / hello /
    Object: foo / x
    Object: foo / y

> await listDir('foo / bar /')
=> Object: foo / bar / a
    Object: foo / bar / b

> await listDir('foo / hello / C /')
=> Object: foo / hello / C / 1
    Object: foo / hello / C / 2
    ...
    Object: foo / hello / C / 9999
```

11.8.6 Copy an object

This topic describes how to copy an object.

You can use `copy` to copy an object in either of the following situations:

- The source and destination objects are in the same bucket.
- The source and destination objects are in different buckets within the same region
 - The source object name must be in the format of `'/Bucket/object.'`

Additionally, you can process Object Meta as follows when you copy an object:

- If `meta` is not specified, the Object Meta of the source object is copied.
- If `meta` is specified, the specified Object Meta overwrites the Object Meta of the source object.

Use the following code to copy an object:

```
let OSS = require ('ali-oss')

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function copy () {
  try {
    // Copy the object from a bucket to another
    bucket.
    let result = await client.copy ('to', '/from - bucket
/ from');
    console.log (result);

    // Copy Object Meta.
    let result = await client.copy ('to', 'from');
    console.log (result);

    // Overwrite Object Meta.
    let result = await client.copy ('to', 'from', {
      meta: {
        year: 2015,
        people: 'mary'
      }
    });
    console.log (result);
  } catch (e) {
    console.log (e);
  }
}
```

11.8.7 Delete objects

This topic describes how to delete objects.

Delete an object

Use `delete` to delete an object:

```
let OSS = require ('ali-oss')

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});
```



```

async function delete () {
  try {
    let result = await client . delete (' object - name ');
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

delete ();

```

Delete multiple objects

Use `deleteMulti` to delete multiple objects and use `quiet` to specify whether to return the deletion results:

```

let OSS = require (' ali - oss ')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : ' Your bucket name '
});

async function deleteMulti () {
  try {
    let result = await client . deleteMulti ([' obj - 1 ', '
obj - 2 ', ' obj - 3 ']);
    console . log ( result );
    let result = await client . deleteMulti ([' obj - 1 ', '
obj - 2 ', ' obj - 3 '], {
      quiet : true
    });
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

deleteMulti ();

```

11.9 CNAME

This topic describes how to bind a custom domain name.

OSS supports binding custom domain names to OSS, so that you can seamlessly migrate data storage to OSS. For example, if your domain name is "my-domain.com" and all your images were previously located at `http://img.my-domain.com/x.jpg`, after you've moved your storage to OSS and linked your custom domain name, the images can still be accessed at the original address.

- Activate OSS and create a bucket.

- Modify the DNS configuration, add a CNAME record, and direct `img.my-domain.com` to an endpoint of OSS (for example, `my-bucket.oss-cn-hangzhou.aliyuncs.com`).
- On the [OSS console](#), bind `img.my-domain.com` to the created bucket.
- Upload images to the created bucket in OSS.

By following the preceding steps, you can use the original address `http://img.my-domain.com/xx.jpg` to access the images on OSS. For more information on how to bind custom domain names, see [Bind Custom Domain Names](#).

The SDK supports using a CNAME as the endpoint. Therefore, set the `cname` parameter to `true`.

```
let OSS = require('ali-oss')

let client = new OSS({
  endpoint: '< Your endpoint >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  cname: true
});

client.useBucket('my-bucket')
```



Notice:

The `list_buckets` interface is unavailable when a CNAME is used as the endpoint, because the custom domain name has been bound to a specific bucket.

11.10 Authorized access

This topic describes how to authorize temporary access to OSS.

Use STS for temporary access authorization

You can use Alibaba Cloud STS to authorize temporary access to OSS. STS is a Web service that provides a temporary access token to a cloud computing user. You can use STS to assign a third-party application or a federated user (you can manage the user ID) an access credential with a custom validity period and permissions. For more information, see [STS introduction](#).

STS benefits:

- You do not need to expose your long-term key (AccessKey) to a third-party application. You need only to generate an access token and send the access token

to the third-party application. You can customize the access permissions and validity period of this token.

- The access token automatically expires after the validity period.

For more information about how to use STS to access OSS, see the "[Use STS to authorize temporary access to OSS](#)" section in OSS Developer Guide.

To use STS to access OSS, you must set the `stsToken` parameter. Example:

```
let OSS = require('ali-oss');
let STS = OSS.STS;
let sts = new STS({
  accessKeyId: '< The AccessKey ID of the RAM user >',
  accessKeySecret: '< The AccessKey Secret of the RAM user >'
});
async function assumeRole() {
  try {
    let token = await sts.assumeRole(
      '< role - arn >', '< policy >', '< expiration >', '< session - name >');
    let client = new OSS({
      region: '< region >',
      accessKeyId: token.credentials.accessKeyId,
      accessKeySecret: token.credentials.accessKeySecret,
      stsToken: token.credentials.securityToken,
      bucket: '< bucket - name >'
    });
  } catch (e) {
    console.log(e);
  }
}
assumeRole();
```

You can customize an STS policy when applying for a temporary token from STS.

The temporary permission you apply for is owned by your role and specified by the policy at the same time. The following example specifies the STS policy used to apply for the read-only permission on `my-bucket`, and sets the validity period of the temporary token to 15 minutes:

```
let OSS = require('ali-oss');
let STS = OSS.STS;
let sts = new STS({
  accessKeyId: '< The AccessKey ID of the RAM user >',
  accessKeySecret: '< The AccessKey Secret of the RAM user >'
});
let policy = {
  "Statement": [
    {
      "Action": [
        "oss:Get*"
      ],
      "Effect": "Allow",
    }
  ]
}
```

```

    " Resource ": [" acs : oss :*: *: my - bucket /*"]
  },
  " Version ": " 1 "
};
async function assumeRole () {
  try {
    let token = await sts . assumeRole (
      '< role - arn >', policy , 15 * 60 , '< session - name >');
    let client = new OSS ({
      region : '< region >',
      accessKeyId : token . credential s . AccessKeyId ,
      accessKeySecret : token . credential s . AccessKeySecret
    },
    {
      stsToken : token . credential s . SecurityToken ,
      bucket : '< bucket - name >'
    });
  } catch ( e ) {
    console . log ( e );
  }
}
assumeRole ();

```

Use a signed URL to authorize temporary access

- Generate a signed URL

You can provide a signed URL to authorize temporary access for a visitor. When a signed URL is generated, you can specify the validity period for the URL to restrict the period of access from visitors.

- Generate a signed URL for an object

Use the following code to generate a signed URL for an object:



Note:

`name {String}` indicates the name of the object stored in OSS. `[expires] {Number}` indicates the validity period of the URL. The default value is 1,800 seconds. For more information about other parameters, see [GitHub](#).

```

let OSS = require (' ali - oss ');
let store = new OSS ({
  bucket : '< your bucket >',
  region : '< your region >',
  accessKeyId : '< your accessKeyId >',
  accessKeySecret : '< your accessKeySecret >'
});
const url = store . signatureUrl (' ossdemo . txt ');
console . log ( url );
// -----
const url = store . signatureUrl (' ossdemo . txt ', {
  expires : 3600 ,
  method : ' PUT '
});
console . log ( url );

```

```
// put object with signatureUrl
// -----
const url = store.signatureUrl('ossdemo.txt', {
  expires: 3600,
  method: 'PUT',
  'Content-Type': 'text/plain; charset=UTF-8',
});
console.log(url);

// -----
const url = store.signatureUrl('ossdemo.txt', {
  expires: 3600,
  response: {
    'content-type': 'text/custom',
    'content-disposition': 'attachment'
  }
});
console.log(url);

// put operation
```

- **Generated a signed URL for IMG**

```
let OSS = require('ali-oss');
let store = new OSS({
  bucket: '<your bucket>',
  region: '<your region>',
  accessKeyId: '<your accessKeyId>',
  accessKeySecret: '<your accessKeySecret>'
});
const url = store.signatureUrl('ossdemo.png', {
  process: 'image/resize,w_200'
});
console.log(url);

// -----
const url = store.signatureUrl('ossdemo.png', {
  expires: 3600,
  process: 'image/resize,w_200'
});
console.log(url);
```

11.11 Set ACLs

This topic describes how to set ACLs.

OSS allows you to set ACLs for buckets and objects respectively. This helps to easily control external access to your resources. You can set three ACLs for a bucket:

- **public-read-write:** Anonymous users are allowed to create/retrieve/delete objects in the bucket.
- **public-read:** Anonymous users are allowed to retrieve objects in the bucket.
- **private:** Anonymous users are not allowed to access objects in the bucket.

Signature is required for all accesses.

When a bucket is created, the private ACL applies by default. You can use

`putBucketA CL` to set bucket ACLs and use `getBucketA CL` to get the bucket ACL.

```
let OSS = require ('ali - oss ')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function bucketACL () {
  try {
    let result = await client . getBucketA CL (' bucket - name ');
    console . log ( result );

    let result = await client . putBucketA CL (' bucket - name ', ' acl ');
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

bucketACL ();
```

You can set four ACLs for an object.

- **default:** The object inherits the ACL for the bucket it belongs to, that is, the ACL for the object is the same as that for the bucket where the object is stored.
- **public-read-write:** Anonymous users are allowed to read/write the object.
- **public-read:** Anonymous users are allowed to read the object.
- **private:** Anonymous users are not allowed to access objects in the bucket.

Signature is required for all accesses.

When an object is created, the default ACL applies by default. You can use `putACL` to set the object ACL.

```
let OSS = require ('ali - oss ')
let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});
async function bucketACL () {
  try {
    let result = await client . getACL (' my - object ');
    console . log ( result . acl ); // default
    await client . putACL (' my - object ', ' public - read ');
```

```
    result = yield client . getACL ( ' my - object ' );  
    console . log ( result . acl ); // public - read  
  } catch ( e ) {  
    console . log ( e );  
  }  
}
```

**Notice:**

1. If an object is configured with an ACL policy (not default), the object ACL takes priority during permission authentication when it is accessed. The bucket ACL is ignored.
2. If anonymous access is allowed (public-read or public-read-write is configured for the object), you can directly access the object using a browser.

For example : `http://bucket-name.oss-cn-hangzhou.aliyuncs.com/object.jpg`.

For more information about ACL, see [Access Control](#).

11.12 Manage lifecycle rules

OSS allows you to set lifecycle rules for buckets to automatically remove expired objects and save the storage space.

You can configure lifecycle rules for OSS buckets or objects to automatically delete expired objects and parts or change the storage class of expired objects to IA or Archive, saving storage costs. A lifecycle rule includes the following elements:

- **Rule ID:** Identifies a lifecycle rule. A rule ID is unique in a bucket.
- **Strategy:** You can set one of the following two application strategies for a rule. The strategy for all rules set for a bucket must be the same.
 - The rule applies to objects with specified prefixes. You can create multiple rules with this strategy. Prefixes specified in different rules must be different.
 - The rule applies to the entire bucket. You can create only one rule with this strategy.
- **Expiration period:**
 - Specifies the expiration period (N days) of a lifecycle rule. An object expires N days after it is modified for the last time.
 - Specifies an expiration time. Objects modified before the time expire.

- **Status:** Indicates whether the rule is enabled or disabled.

For more information, see [Lifecycle rules](#).

Configure lifecycle rules

You can use `putBucketLifecycle` to configure lifecycle rules.

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: '< Your bucket name >'
});

async function putBucketLifecycle () {
  try {
    let result = yield client.putBucketLifecycle('bucket-name', [
      {
        id: 'rule1',
        status: 'Enabled',
        prefix: 'foo/',
        days: 3
      },
      {
        id: 'rule2',
        status: 'Disabled',
        date: '2022-10-11T00:00:00.000Z'
      }
    ]);
    console.log(result);
  } catch (e) {
    console.log(e);
  }
}

putBucketLifecycle();
```

View lifecycle rules

You can use `getBucketLifecycle` to view lifecycle rules:

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: '< Your bucket name >'
});

async function getBucketLifecycle () {
  try {
    let result = await client.getBucketLifecycle('bucket-name');
    console.log(result);
  }
}
```



```

    } catch ( e ) {
      console . log ( e );
    }
  }

  getBucketLifecycle ();

```

Clear lifecycle rules

You can use `deleteBucketLifecycle` to set a null Rule array to clear the lifecycle rules:

```

let OSS = require ( 'ali-oss' )

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function deleteBucketLifecycle () {
  try {
    let result = await client.deleteBucketLifecycle (
      bucketName );
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

deleteBucketLifecycle ();

```

11.13 Set logging

OSS allows you to configure bucket access logging to store the bucket access logs in a specified bucket for future analysis. The logs are stored as objects in the specified bucket.

The object name format is as follows:

```

< TargetPrefix >< SourceBucket >- YYYY - mm - DD - HH - MM - SS -
UniqueString

```

The `TargetPrefix` needs to be specified by the user during configuration. For more information about access logging, see [Bucket Access Logging](#).

Enable bucket logging

Use `putBucketLogging` to enable bucket logging:

```

let OSS = require ( 'ali-oss' )

```

```

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});
async function putBucketLogging () {
  try {
    let result = await client.putBucketLogging (' bucket -
name ', ' logs /');
    console.log ( result )
  } catch ( e ) {
    console.log ( e )
  }
}
putBucketLogging ();

```

View bucket logging settings

Use `getBucketLogging` to view bucket logging settings:

```

let OSS = require (' ali - oss ')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function getBucketLogging () {
  try {
    let result = await client.getBucketLogging (' bucket -
name ');
    console.log ( result );
  } catch ( e ) {
    console.log ( e );
  }
}

getBucketLogging ();

```

Disable bucket logging

Use `deleteBucketLogging` to disable bucket logging:

```

let OSS = require (' ali - oss ')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function deleteBucketLogging () {
  try {
    let result = await client.deleteBucketLogging (' bucket
- name ');
    console.log ( result );
  } catch ( e ) {

```

```

        console . log ( e );
    }

    deleteBucket . setLogging ();

```

11.14 Static website hosting

You can set your bucket to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

As mentioned in [CNAME](#), OSS allows you to direct your domain name to the OSS service address. By this means, OSS bucket is accessed when you access your website. You must specify the names of the objects in the bucket corresponding to the homepage (index page) and error page of the website respectively.

For more information, see [Static Website Hosting](#).

Set website hosting

Use `putBucketWebsite` to set website hosting:

```

let OSS = require ('ali-oss')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function putBucketWebsite () {
  try {
    let result = await client . putBucketWebsite (' bucket -
name ', {
      index : ' index . html ',
      error : ' error . html '
    });
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

putBucketWebsite ();

```

View website hosting

Use `getBucketWebsite` to view website hosting:

```

let OSS = require ('ali-oss')

let client = new OSS ({
  region : '< Your region >'

```

```

    accessKeyId : '< Your AccessKeyId >',
    accessKeySecret : '< Your AccessKeySecret >',
    bucket : '< Your bucket name >'
  });

  async function getBucketWebsite () {
    try {
      let result = await client . getBucketWebsite (' bucket -
name ');
      console . log ( result );
    } catch ( e ) {
      console . log ( e );
    }
  }

  getBucketWebsite ();

```

Clear website hosting

Use `deleteBucketWebsite` to clear website hosting:

```

let OSS = require ('ali-oss')

let client = new OSS ({
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function deleteBucketWebsite () {
  try {
    let result = await client . deleteBucketWebsite (' bucket
- name ');
    console . log ( result );
  } catch ( e ) {
    console . log ( e );
  }
}

deleteBucketWebsite ();

```

11.15 Anti-leech

This topic describes how to use anti-leech.

To prevent your data on OSS from being leached, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about anti-leach, see [Anti-leech settings](#).

Configure a referer whitelist

Use `putBucketReferer` to set a referer whitelist:

```
let OSS = require ('ali-oss')

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: '< Your bucket name >'
});

async function putBucketReferer () {
  try {
    let result = await client.putBucketReferer ('bucket -
name', true, [
  'my-domain.com',
  '*.example.com'
]);
    console.log (result);
  } catch (e) {
    console.log (e);
  }
}

putBucketReferer ();
```

View a referer whitelist

Use `getBucketReferer` to view a referer whitelist:

```
let OSS = require ('ali-oss')

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: '< Your bucket name >'
});

async function getBucketReferer () {
  try {
    let result = await client.getBucketReferer ('bucket -
name');
    console.log (result);
  } catch (e) {
    console.log (e);
  }
}

getBucketReferer ();
```

Clear a referer whitelist

Use `deleteBucketReferer` to clear a referer whitelist:

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: '< Your bucket name >'
});

async function deleteBucketReferer () {
  try {
    let result = await client.deleteBucketReferer('bucket - name');
    console.log(result);
  } catch (e) {
    console.log(e);
  }
}

deleteBucketReferer();
```

11.16 Image processing

Alibaba Cloud OSS provides image processing service featuring massive processing power, security, low costs, and high reliability. By uploading and storing source images in OSS, you can process images anytime, anywhere, and on any Internet device through a simple RESTful interface.

Image processing basic features

Image processing provides the following features:

- [Retrieve dominant image tones](#)
- [Format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Customize image processing styles](#)

- Calling multiple image processing features through [Image service access rules](#).

Usage of image processing

Image processing uses the standard HTTP GET request for access. All processing parameters are encoded in the QueryString in the URL.

If the ACL for an image is private read and write, only authorized users are allowed can access the image.

- Anonymous access

If the access permission of the image object is `public-read`, as shown in the following table, you can access image processing anonymously.

Bucket ACL	Object ACL
Public-read or Public-read-write	Default
Any ACL	Public-read or Public-read-write

Access image processing anonymously using the third-level domain in the following format:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >?  
&x-oss-process=image/< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Name of the user' s bucket
endpoint	The access domain name for the data center of the user' s bucket
object	The image object uploaded to the OSS
image	The identifier reserved by the image processing
action	Operations on images, such as scaling, cropping, and rotating

Parameter	Description
param	Corresponding to the operations on the image

- Basic operations

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Custom style.

Access image processing anonymously using the third-level domain in the following format:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

■ **Style:** The identifier reserved by the system for the user's custom style

■ **yourStyleName:** Name of the custom style, that is, the rule name of the console-defined style

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Through Image Service access rules,

you can implement multiple operations on an image. The format is as follows:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Image processing also supports HTTPS access.

For example:

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Authorized access

For objects with private permissions, as shown in the following table, you must be authorized to access image processing:

Bucket ACL	Object ACL
private	default
Any ACL	private

The image processing URL code generated with signature is as follows:

```
let OSS = require('ali-oss');
let client = new OSS({
  accessKeyId: '<accessKeyId>',
  accessKeySecret: '<accessKeySecret>',
  bucket: '<bucketName>',
  endpoint: '<endpoint>', for example, http://oss-cn-hangzhou.aliyuncs.com
});
// Expired for 10 minutes. The image processing style
// is: "image/resize,w_300".
let signUrl = client.signatureUrl('example.jpg', {
  expires: 600, 'process': 'image/resize,w_300'
});
console.log('signUrl=' + signUrl);
```



Note:

- Authorized access supports custom styles, HTTPS, and cascading processing.
- Expiration time (expires) is measured in seconds.

• SDK access

You can directly use SDK to access and process image objects with any permissions

.

Image object processing through SDK supports custom styles, HTTPS, and Image Service access rules.

- Basic operations

Run the following code to perform basic operations on an image:

```
let OSS = require('ali-oss');
let client = new OSS({
  accessKeyId: '<accessKeyId>',
  accessKeySecret: '<accessKeySecret>',
  bucket: '<bucketName>',
  endpoint: '<endpoint>', for example, http://oss-cn-hangzhou.aliyuncs.com
});
// Scale the image.
```

```

async function scale () {
  try {
    let result = await client . get (' example . jpg ', './
example - resize . jpg ', { process : ' image / resize , m_fixed
, w_100 , h_100 '});
  } catch ( e ) {
    console . log ( e );
  }
}
// Crop the image .
async function cut () {
  try {
    let result = await client . get (' example . jpg ',
'./ example - crop . jpg ', { process : ' image / crop , w_100 ,
h_100 , x_100 , y_100 , r_1 '});
  } catch ( e ) {
    console . log ( e )
  }
}
// Rotate the image .
async function rotate () {
  try {
    let result = await client . get (' example . jpg ', './
example - rotate . jpg ', { process : ' image / rotate , 90 '});
  } catch ( e ) {
    console . log ( e );
  }
}
// Sharpen the image .
async function sharpen () {
  try {
    let result = yield client . get (' example . jpg ', './
example - sharpen . jpg ', { process : ' image / sharpen , 100
'});
  } catch ( e ) {
    console . log ( e );
  }
}
// Add watermarks .
async function watermark () {
  try {
    let result = yield client . get (' example . jpg ', './
example - watermark . jpg ', { process : ' image / watermark ,
text_SGVsb_G8g5Zu - 54mH5pyN5Y_qhIQ '});
  } catch ( e ) {
    console . log ( e );
  }
}
// Convert the image format .
async function format () {
  try {
    let result = await client . get (' example . jpg ', './
example - format . jpg ', { process : ' image / format , png
'});
  } catch ( e ) {
    console . log ( e );
  }
}
// Obtain image information .
async function info () {
  try {
    let result = await client . get (' example . jpg ', './
example - info . txt ', { process : ' image / info '});
  } catch ( e ) {

```

```

        console . log ( e );
    }
}

```

- Custom styles



Note:

You must first log on to the OSS console to add the custom style 'example-style'.

You can run the following code to customize an image style:

```

let OSS = require (' ali - oss ');
let client = new OSS ({
    accessKeyId : '< accessKeyId >',
    accessKeySecret : '< accessKeySecret >',
    bucket : '< bucketName >',
    endpoint : '< endpoint , for example , http :// oss - cn - hangzhou . aliyuncs . com >'
});
// Custom styles
async function style () {
    try {
        let result = await client . get (' example . jpg ', './
example - custom - style . jpg ', { process : ' style / example -
style "});
    } catch ( e ) {
        console . log ( e );
    }
}
style ();

```

- Cascading processing

Run the following code to perform cascade operations on an image:

```

let OSS = require (' ali - oss ');
let client = new OSS ({
    accessKeyId : '< accessKeyId >',
    accessKeySecret : '< accessKeySecret >',
    bucket : '< bucketName >',
    endpoint : '< endpoint , for example , http :// oss - cn - hangzhou . aliyuncs . com >'
});
// Cascading processing
async function cascade () {
    try {
        let result = await client . get (' example . jpg ', './
example - cascade . jpg ', { process : ' image / resize , m_fixed
, w_100 , h_100 / rotate , 90 '});
    } catch ( e ) {
        console . log ( e );
    }
}

```

```
cascade ();
```

11.17 Exception handling

If a request error occurs when you use OSS Node.js SDK usage, an exception is thrown. Detailed information about the error is recorded in the log (`oss_sdk . log` by default).

ClientError

ClientErrors indicate exceptions occurred inside the SDK, for example, parameter setting errors and errors occurred when objects are modified in resumable upload or download tasks.

ServerError

ServerErrors are the parsing results of the error messages generated by the server and indicate errors occurred on the server. ServerErrors include the following attributes:

- **status:** Indicates the HTTP status code of the error request.
- **code:** Indicates the OSS error code.
- **message:** Indicates the OSS error message.
- **requestId:** Indicates the UUID that uniquely identifies the request. Seek help from OSS development engineers by providing this requestId if necessary.

For more information about common error messages in OSS, see [OSS Error Response](#).

Debugging

If ClientErrors or ServerErrors occur when you use OSS Node.js SDK, enable debugging by setting the `DEBUG` environment variable as follows:

```
DEBUG = ali - oss    node    app . js
```

In a browser environment, set the `localStorage . debug` variable in the console to enable debugging:

```
localStorage . debug = ' ali - oss '
```

11.18 FAQ

This topic describes common problems that may occur in the use of OSS Node.js SDK and the solutions for the problems.

How to enable HTTPS access

To enable HTTPS access, specify the value of the `secure` parameter to `true` when initializing the SDK.

How to obtain the upload progress

You can specify the `progress` parameter to obtain the upload progress when performing [multipart upload](#) operations.

How to obtain the download progress

The download progress is calculated based on the size of the download traffic in OSS Node.js SDK.

How to upload Base64-encoded images

Convert the images to File objects and upload them to the OSS server by calling APIs.

```
function dataURLtoFile ( dataurl , filename ) {
    let arr = dataurl . split (','), mime = arr [ 0 ]. match
(/:(. *) ;/)[ 1 ],
    bstr = atob ( arr [ 1 ]), n = bstr . length , u8arr =
new Uint8Array ( n );
    while ( n --){
        u8arr [ n ] = bstr . charCodeAt ( n );
    }
    return new File ([ u8arr ], filename , { type : mime });
}

let file = dataURLtoFile ('< base64 content >', '');

client . multipartU pload ('< oss file name >', file ). then
( ( res )=> {
    console . log ( res )
}). catch (( err ) => {
    console . log ( err )
});
```

How to upload a file to a specified directory

You only need to add the prefix of the directory to the name of the object that you want to upload. For more information, see: [Comparison of OSS and file system](#).

```
let OSS = require ('ali - oss ')
let client = new OSS ({
```

```
    region : '< Your    region >',
    accessKeyId : '< Your    AccessKeyId >',
    accessKeySecret : '< Your    AccessKeySecret >',
    bucket : 'Your    bucket    name '
  });

  client.multipartUpload('base-dir/' + 'object-name', '
local-file', {
    progress : async function ( p ) {
      console.log('Progress : ' + p);
    }
  });
  console.log(result);
}).then((res) => {
  console.log(res)
}).catch((err) => {
  console.log(err);
});
```

How to obtain the signed URL of an object

You can call the `signatureUrl` method to obtain the signed URL used to download an object. For more information, see the [signatureUrl](#) part in the OSS SDK document.

Common errors

- [#unique_728](#)
- [#unique_220](#)

12 Browser.js

12.1 Installation

This topic describes how to install OSS Browser.js SDK.

Download the SDK

Demos in official documents are based on SDK 6.X. For SDKs earlier than 6.X, see [Documents for 5.X SDK](#). For more information about how to upgrade your SDK to 6.X, see [Upgrade document](#).

- [GitHub](#)
- [API documentation](#)
- [ChangeLog](#)

Environment preparations

- Requirements
 - You must have activated Alibaba Cloud OSS. If you do not know about Alibaba Cloud OSS, log on to the OSS Product Homepage. For more information, see [Product Overview](#).
 - You must have created an AccessKeyID and an AccessKeySecret. Since the Alibaba Cloud account AccessKey gives you permission to access all APIs, we recommend that you follow the Alibaba Cloud Security Best Practices. If you deploy the service on the server, you can use an RAM sub-account or STS for API access or daily O&M management and control. If you deploy the service on the client, we recommend you use STS for API access. For more information, see [Access control](#).

- Environment requirements

The OSS BrowserJS-SDK is built on the [Node.js](#) environment. It generates codes that apply to browsers through [Browserify](#) and [Babel](#).

However, because of the special natures of the browser environment, some features cannot be used:

- Stream upload: The chunked encoding cannot be set in browsers. The multipart upload can be used instead.
- Operations on local files: You cannot directly operate on local file systems in the browser. You can download the file using a signed URL.
- Bucket-related operations (listBuckets/BucketACL/BucketLogging). Since OSS does not support bucket-related cross-origin requests currently, you can perform bucket-related operations in the console.

Usage

The OSS BrowserJS-SDK supports both synchronous and asynchronous usages.

- Synchronous mode: Synchronize the asynchronous mode based on `async / await` in the ES7 standard.
- Asynchronous mode: Similar to callback. The API returns [Promise](#). The `. then ()` method is used to process the returned result and the `. catch ()` method is used to process errors.

`new OSS ()` is used to create a client both in the synchronous mode and asynchronous mode.

Here are the examples of uploading a file and then downloading the file immediately in the two modes:

- Synchronous mode

```
// Create an OSS client instance . For detailed
parameters , see the configurat ion items described in
the documents .
let client = new OSS (...);

async function put () {
  try {
    // object indicates the name of the uploaded
    object , which can be customized .
    // file indicates the file that needs to be
    upload in the browser . The HTML and Blob formats
    are supported .
    let r1 = await client . put (' object ', file );
```



```

        console . log ( ' put    success : % j ', r1 );
        let    r2 = await    client . get ( ' object ' );
        console . log ( ' get    success : % j ', r2 );
    } catch ( e ) {
        console . error ( ' error : % j ', e );
    }
}

put ();

```

- **Asynchronous mode**

```

// Create an OSS client instance . For detailed
parameters , see the configurat ion items described in
the documents .
let    client = new    OSS (...);

// object indicates the name of the uploaded object
, which can be customized .
// file indicates the file that needs to be upload
in the browser . The HTML and Blob formats are
supported .
client . put ( ' object ', file ) . then ( function ( r1 ) {
    console . log ( ' put    success : % j ', r1 );
    return    client . get ( ' object ' );
} ) . then ( function ( r2 ) {
    console . log ( ' get    success : % j ', r2 );
} ) . catch ( function ( err ) {
    console . error ( ' error : % j ', err );
} );

```

Installation

- **Usage in browsers**

Supported browsers:

- IE (>=10) and Edge
- Mainstream Chrome/Firefox/Safari versions
- Mainstream Android/iOS/WindowsPhone versions

Browser.js SDK can be installed in the following methods:

- **Browser introduction:**



Notice:

Some browsers do not support promise, so a promise compatibility library must be introduced. For example, promise-polyfill must be introduced for IE10 and IE11.

```
<!-- Introduce online resources -->
```

```
< script   src =" http :// gosspublic . alicdn . com / aliyun -
oss - sdk - x . x . x . min . js "></ script >
```

```
-- Introduce   local   resources   -->
< script   src ="./ aliyun - oss - sdk - x . x . x . min . js "></
script >
```



Note:

- x.x.x represents the version number. For more information about the specific version, see [here](#).
- The introduction of online resources depends on the stability of the CDN server. Users are recommended to introduce resources offline, that is, through local resources or self-built.

You can use the `OSS . Wrapper` object in the code:

```
< script   type =" text / javascript ">
  let   client   = new   OSS ({
    region : '< oss   region >',
    // Since the Alibaba Cloud account AccessKey gives
    you permission to access all APIs , we recommend
    that you access an API using STS by following
    the Alibaba Cloud Security Best Practices .
    accessKeyId : '< Your   accessKeyId ( STS )>',
    accessKeySecret : '< Your   accessKeySecret ( STS )>',
    stsToken : '< Your   securityToken ( STS )>',
    bucket : '< Your   bucket   name >'
  });
</ script >
```

- Use [npm](#) to install the SDK:

```
npm   install   ali - oss
```

After the installation is completed, you can reference it using “import” or “require”. The `require` mode is not supported by browsers. Therefore, you must use packaging tools, such as `webpack` and `browserify` in the development environment.

```
let   OSS   = require (' ali - oss ');

let   client   = new   OSS ({
  region : '< oss   region >',
  // Since the Alibaba Cloud account AccessKey gives
  you permission to access all APIs , we recommend
  that you follow the Alibaba Cloud Security Best
  Practices . If you deploy the service on the
  server , use an RAM sub - account or STS , and if
  you deploy the service on the client , use STS .
  accessKeyId : '< Your   accessKeyId >',
  accessKeySecret : '< Your   accessKeySecret >',
  bucket : '< Your   bucket   name >'
});
```

```
});
```

- Use [Bower](#):

For web projects, you can install the SDK through the [Bower](#):

```
bower install ali-oss
```

Then you can add the `<script>` tag in the webpage:

```
<script src="bower_components/ali-oss/dist/aliyun-oss-sdk.min.js"></script>
```

12.2 Quick start

This topic describes how to quickly access the OSS service in the BrowserJS-SDK, including uploading/downloading files and viewing file lists.



Notice:

Since the Alibaba Cloud account AccessKey gives you permission to access all APIs, we recommend that you not use the AK that can cause configuration leak by following the Alibaba Cloud Security Best Practices. Therefore, you must create and use STS (temporary account permission) to authorize the client.

For more information about browser usage, see [Browser Application](#).

For demo project, see [example](#).

Bucket settings

The [CORS settings](#) of the bucket must be enabled to access OSS directly in a browser:

- Set allowed origins to `*`
- Set allowed methods to `PUT` , `GET` , `POST` , `DELETE` , and `HEAD` .
- Set allowed headers to `*`.
- Set expose headers to `ETag` `x-oss-request-id` .



Notice:

Set the CORS rule as the first option.

来源	允许 Methods	允许 Headers
*	GET POST PUT DELETE HEAD	*

*** 来源**

来源可以设置多个，每行一个，每行最多能有一个通配符「*」

*** 允许 Methods** ☒ GET ☒ POST ☒ PUT ☒ DELETE ☒ HEAD

允许 Headers

允许 Headers 可以设置多个，每行一个，每行最多能有一个通配符「*」

暴露 Headers ☒ etag ☒ x-oss-request-id

暴露 Headers 可以设置多个，每行一个，不允许出现通配符「*」

缓存时间 (秒)

Use the SDK

Currently, browsers do not support direct bucket-related operations (for example, list buckets, and get/set bucket logging/referer/website/cors), but support object-related operations (for example, upload/download files, and view file lists).

- Include the SDK

First, include the following tags in the `< head >` of the HTML file:

```
< script src =" http :// gosspublic . alicdn . com / aliyun - oss
- sdk - x . x . x . min . js "></ script >
```

For more introduction methods, see [Installation](#).

Create a client using `new OSS . Wrapper ()`. `OSS.Wrapper` provides an asynchronous API similar to the callback method to process the returned results in `. then ()` and process errors in `. catch ()`.

- Build an STS Server and obtain temporary authorization information from the client

1. Activate the STS service for your account. If the service has been activated, you can skip this step. To activate the STS service, see [STS Activation](#). For more

information on how to configure OSS authorization policy, see [RAM and STS authorization policy configuration](#).

2. Apply the exclusive STS signature service through the STS SDK or API. To help you quickly get started with the STS service, we provide demo programs in multiple languages:

- NodeJS: [Download address](#)
- PHP: [Download address](#)
- Java: [Download address](#)
- Ruby: [Download address](#)



Note:

The demo program is only used as reference for STS feature implementation. To implement STS features in the production environment, you must develop codes required for authentication and other functions.

3. Initiate requests to the STS service built in the frontend (browser) to obtain the STS temporary authorization information.

```
< script    type =" text / javascript ">
  // OSS . urlib  is  the  logic  of  sending  requests
  encapsulat ed  within  the  SDK . Developers  can  use
  any  library  that  sends  requests  to  send  requests
  to  ' sts - server '.
  OSS . urllib . request (" http :// your_sts_s erver /", {
  method : ' GET '}, ( err , response ) => {
    if ( err ) {
      return  alert ( err );
    }
    try {
      result = JSON . parse ( response );
    } catch ( e ) {
      errmsg = ' parse  sts  response  info  error : '
+ e . message ;
      return  alert ( errmsg );
    }
    // console . log ( result )
  })
</ script >
```

- View the object list

```
< script    type =" text / javascript ">
OSS . urlib . request (" http :// your_sts_s erver /", { method :
' GET '}, ( err , response ) => {
  if ( err ) {
    return  alert ( err );
  }

  try {
    result = JSON . parse ( response );
```

```

    } catch ( e ) {
        return alert ( ' parse sts response info error : ' +
e . message );
    }

    let client = new OSS ({
        accessKeyId : result . AccessKeyId ,
        accessKeySecret : result . AccessKeySecret ,
        stsToken : result . SecurityToken ,
        endpoint : '< oss endpoint >',
        bucket : '< Your bucket name >'
    });

    client . list ({
        'max-keys' : 10
    }). then ( function ( result ) {
        console . log ( result );
    }). catch ( function ( err ) {
        console . log ( err );
    });
});
</ script >

```

The parameter `region` refers to the region used when you applied for OSS, such as `oss-cn-hangzhou`. The complete region list can be viewed in [OSS Nodes](#).

Open the HTML file in the browser, and open the Chrome Developer Console to view the result in List files.

The screenshot displays a web browser window with the address bar showing 'localhost:63343'. The browser's developer console is open, showing the response of the 'listFiles' API call. The response is a JSON object with the following structure:

```

{
  "LastModified": "2017-12-04T08:12:47.000Z",
  "ETag": "7F6D22ED37A73B157F03255A00000000",
  "Type": "Appendable",
  "Size": 1024000,
  "StorageClass": "Standard",
  "Owner": {
    "ID": "1855202418087745",
    "DisplayName": "1855202418087745"
  }
}

```

The 'List files' section shows a table with the following data:

Key	Size	LastModified
img1.jpg	2255940	2017-12-22T03:11:06.000Z
object	169110	2017-12-21T02:52:17.000Z
file10m	10240000	2017-12-13T09:35:17.000Z

The 'Content' section shows 'Hello, OSS!'. The 'Store as' section shows 'object'. The 'Download file' section shows 'object' and 'filename'.



Notice:

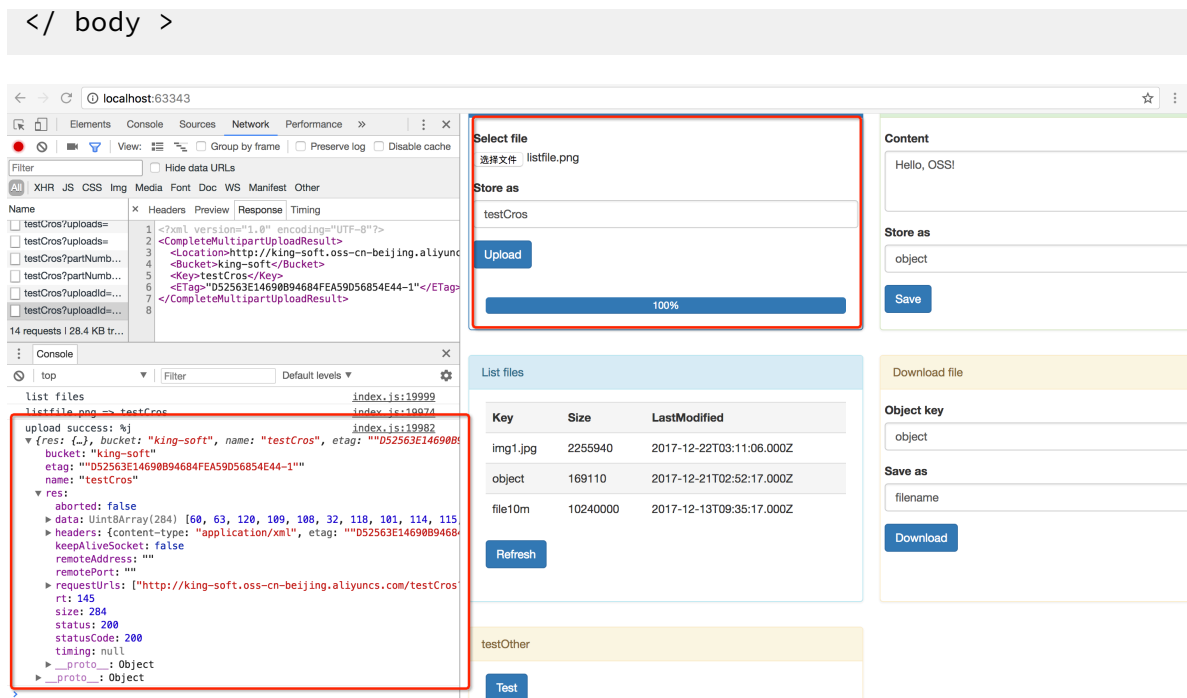
A browser is an untrusted environment. An extremely high risk occurs if you store the AccessKeyID and AccessKeySecret in a browser. We recommend that you only use plaintext settings for testing purposes. STS authentication mode or self-signed mode is recommended for business applications. For more

information, see [Resource Access Management](#) and [Direct Transfer on a Mobile Device](#).

- Upload an object

We use the `multipartUpload` API to upload objects. For more information about `multipartUpload`, see [Multipart upload](#).

```
< body >
  < input type="file" id="file" />
  < script type="text/javascript">
    document.getElementById('file').addEventListener('
change', function (e) {
      let file = e.target.files[0];
      let storeAs = 'upload-' + file.name;
      console.log(file.name + ' => ' + storeAs);
      // OSS.urlLib is the logic of sending requests
      encapsulated within the SDK. Developers can use any
      library that sends requests to send requests to '
      sts-server'.
      OSS.urlLib.request("http://your_sts_server/", {
        method: 'GET'}, (err, response) => {
        if (err) {
          return alert(err);
        }
        try {
          result = JSON.parse(response);
        } catch (e) {
          return alert('parse sts response info error: ' + e
            .message);
        }
        let client = new OSS({
          accessKeyId: result.AccessKeyId,
          accessKeySecret: result.AccessKeySecret,
          stsToken: result.SecurityToken,
          endpoint: '<oss endpoint>',
          bucket: '<Your bucket name>'
        });
        // storeAs indicates the uploaded object name
        and file indicates the uploaded file.
        client.multipartUpload(storeAs, file).then(function
        (result) {
          console.log(result);
        }).catch(function (err) {
          console.log(err);
        });
      });
    });
  </ script >
```



- Download objects

You cannot directly operate on files in the browser. Therefore, the download link of a file is generated using the signed URL method, and you can download the file by clicking the link.

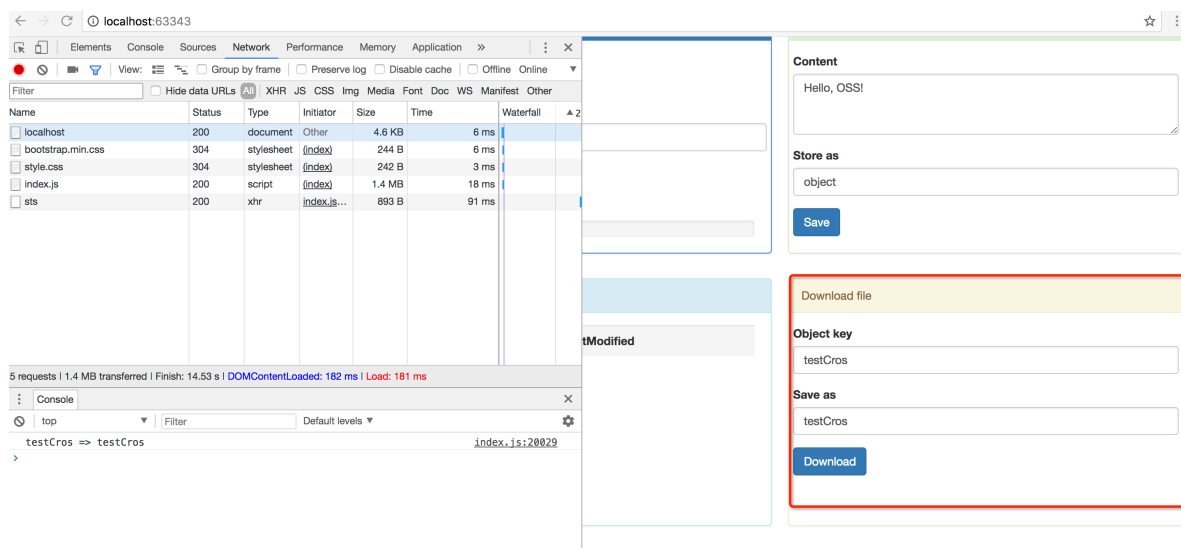
```
< body >
< input type = " button " id = " download " value = " Download
" />
< script type = " text / javascript ">
    document . getElement ById ( ' download ' ). addEventLi stener
( ' click ' , function ( e ) {
    let objectKey = ' test / download_f ile ' ;
    let saveAs = ' download_f ile ' ;
    console . log ( objectKey + ' => ' + saveAs );
    // OSS . urllib is the logic of sending requests
encapsulat ed within the SDK . Developers can use any
library that sends requests to send requests to '
sts - server ' .
    OSS . urllib . request ( " http :// your_sts_s erver / " , {
method : ' GET ' } , ( err , response ) => {
    if ( err ) {
    return alert ( err );
    }
    try {
    result = JSON . parse ( response );
    } catch ( e ) {
    return alert ( ' parse sts response info error : ' + e
. message );
    }
    let client = new OSS ( {
accessKeyI d : result . AccessKeyI d ,
accessKeyS ecret : result . AccessKeyS ecret ,
stsToken : result . SecurityTo ken ,
endpoint : ' < oss endpoint > ' ,
bucket : ' < Your bucket name > '
});
```



```

    let result = client.signatureUrl ( objectKey , {
    expires : 3600 ,
    response : {
        'content-disposition': 'attachment; filename="' +
        saveAs + '"'
    }
    });
    console.log ( result );
    window.location = result ;
});
});
</ script >
</ body >

```



12.3 Options

This topic describes how to configure options.

OSS(options) introduction

- [accessKeyId] {String}: AccessKey you create on Alibaba Cloud console website
- [accessKeySecret] {String}: access secret you create
- [stsToken] {String}: use temporary authorization method. For more information, see [Use STS to access OSS](#).
- [bucket] {String}: the bucket you create on the console.
- [endpoint] {String}: oss region domain.
- [region] {String}: the bucket data region location. The default region is oss-cn-hangzhou.
- [internal] {Boolean}: access OSS with Alibaba Cloud internal network or not, default is false. If your servers are running on Alibaba Cloud too, you can set true to save lot of money.

- [secure] {Boolean} instruct OSS client to use HTTPS (secure: true) or HTTP (secure: false) protocol. For more information, click [here](#).
- [timeout] {String|Number} instance level time-out for all operations. The default value is 60s.

Example:

```
// Assume that an OSS object has been introduced to
// the browser environment in the 'script' or 'npm'
// method.
let store = new OSS ({
  accessKeyId: 'your access key ',
  accessKeySecret: 'your access secret ',
  bucket: 'your bucket name ',
  region: 'oss-cn-hangzhou'
});
```

12.4 Upload objects

You can upload a file to OSS using any of the following methods:

- Upload Blob data
- Resumable upload

Upload Blob Data

Binary Large Object (Blob) indicates a large object of the binary type. The concept of blob is used in some databases. For example, the BLOB type used in MySQL indicates a container for binary data.

You can also use the `put` interface to upload content in a Blob to OSS:

```
let OSS = require ('ali-oss');

let client = new OSS ({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name '
});

async function putBlob () {
  try {
    let result = await client.put ('object-key', new
    Blob (['content'], { type: 'text/plain' }));
    console.log (result);
  } catch (e) {
    console.log (e);
  }
}
```

```
putBlob ();
```

Resumable upload

When the file to be uploaded is large, you can use the multipartUpload interface for multipart upload. Multipartupload is to divide a large request into multiple small requests for execution. As a result, when some of the requests fail, you do not need to upload the entire file again, but only to upload the failed parts. Generally for a file larger than 100 MB, we recommend that you use the preceding multipart upload approach and create a new OSS instance before each multipart upload.

If a `Connection TimeoutErr` or error occurs when you use the multipartUpload interface, you must construct your own method to handle time-out issues. You can reduce the part size, increase the time-out period and allowed retry times, or catch the `Connection TimeoutErr` or error and return it to the user. For more information, see [network error handling](#).

For the usage of multipartUpload API, see [MultipartUpload](#).

Related parameters:

- name {String}: Object name
- file {File}: HTML5 Web File or Blob data

- [options] {Object}: Additional parameters
 - [checkpoint] {Object}: Endpoint checkpoint used in resumable upload. If this parameter is set, the upload starts from the endpoint. If it is not set, the upload restarts.
 - file {File}: File object selected by the user. Users must manually set this parameter if the browser is restarted.
 - name {String}: Uploaded object key
 - fileSize {Number}: File size
 - partSize {Number}: Part size
 - uploadId {String}: Upload Id
 - doneParts {Array}: Array of uploaded parts, including the following objects:
 - number {Number}: Part number
 - etag {String}: Part etag
 - [parallel] {Number}: Number of parts uploaded simultaneously
 - [partSize] {Number}: Part size
 - [progress] {Function}: A generator function or a thunk. The callback function contains the following three parameters: `function`, `async`, and `promise`.
 - (percentage {Number}: Percentage of upload progress (a decimal range from 0 to 1)
 - checkpoint {Object}: Endpoint checkpoint
 - res {Object}: Response returned after a single part is successfully uploaded
 - [meta] {Object}: Header meta information defined by users with a prefix `x-oss-meta-`
 - [mime] {String}: Custom `Content-Type` header
 - [headers] {Object}: Extra headers. See [RFC 2616](#) for more information.
 - 'Cache-Control' : General header used to implement cache mechanisms by specifying a command in HTTP requests and responses. For example: `Cache-Control: public, no-cache`
 - 'Content-Disposition' : Used to indicate the disposition form of a response, which can be an internal reference (a part of a webpage or a page) or

an attachment downloaded and saved locally. For example: `Content -`

`Disposition : somename`

■ **'Content-Encoding'** : Used to compress data of specific media type. For example: `Content - Encoding : gzip`

■ **'Expires'** : Expiration time. For example: `Expires : 3600000`

- [callback] {Object}: Callback settings For more information, see [Callback](#).

■ **url {String}**: The address of the callback server communicated with the OSS server. It corresponds to `callbackUrl` in the `CallBack` parameter. Required

■ **body {String}**: The value of callback request. It is in the JSON format and corresponds to `callbackBody` in the `Callback` parameter. Required

■ **[host] {String}**: The value of the Host header in the callback request. It corresponds to `callbackHost` in the `CallBack` parameter.

■ **[contentType] {String}**: The Content-Type of the callback request. It corresponds to `callbackBodyType` in the `CallBack` parameter.

■ **[customValue] {Object}**: Custom parameters in the callback request. It corresponds to `callback-var` in the `CallBack` parameter.

Example:

1. The upload progress is recorded. When the upload is initiated again, the recorded checkpoint parameter is passed in.
2. We recommend that you create a new OSS instance for each multipart upload task.

```
let OSS = require ('ali-oss')

let ossConfig = {
  region : '< Your region >',
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : 'Your bucket name'
}

let client = new OSS (ossConfig);

let tempCheckpoint;

// Define the upload method
async function multipartUpload () {
  try {
    let result = await client.multipartUpload ('object - key', 'local - file', {
      progress : async function (p, checkpoint) {
        // Record the checkpoint. If you close the browser and continue the upload after restarting the browser, the upload cannot be continued. For more information, see the description of file objects.
      }
    });
  }
}
```

```

        tempCheckpointoint = checkpoint ;
    }
    meta : { year : 2017 , people : ' test ' },
    mime : ' image / jpeg '
})
} catch ( e ){
    console . log ( e );
}
}

// Start uploading
multipartUpload ();

// Pause multipart upload
client . cancel ();

// Resume multipart upload
let resumeclient = new OSS ( ossConfig );
async function resumeUpload () {
    try {
        let result = await resumeclient . multipartUpload (
object - key ' , ' local - file ' , {
    progress : async function ( p , checkpoint ) {
        tempCheckpointoint = checkpoint ;
    },
    checkpoint : tempCheckpointoint
    meta : { year : 2017 , people : ' test ' },
    mime : ' image / jpeg '
})
    } catch ( e ) {
        console . log ( e );
    }
}

resumeUpload ();

```

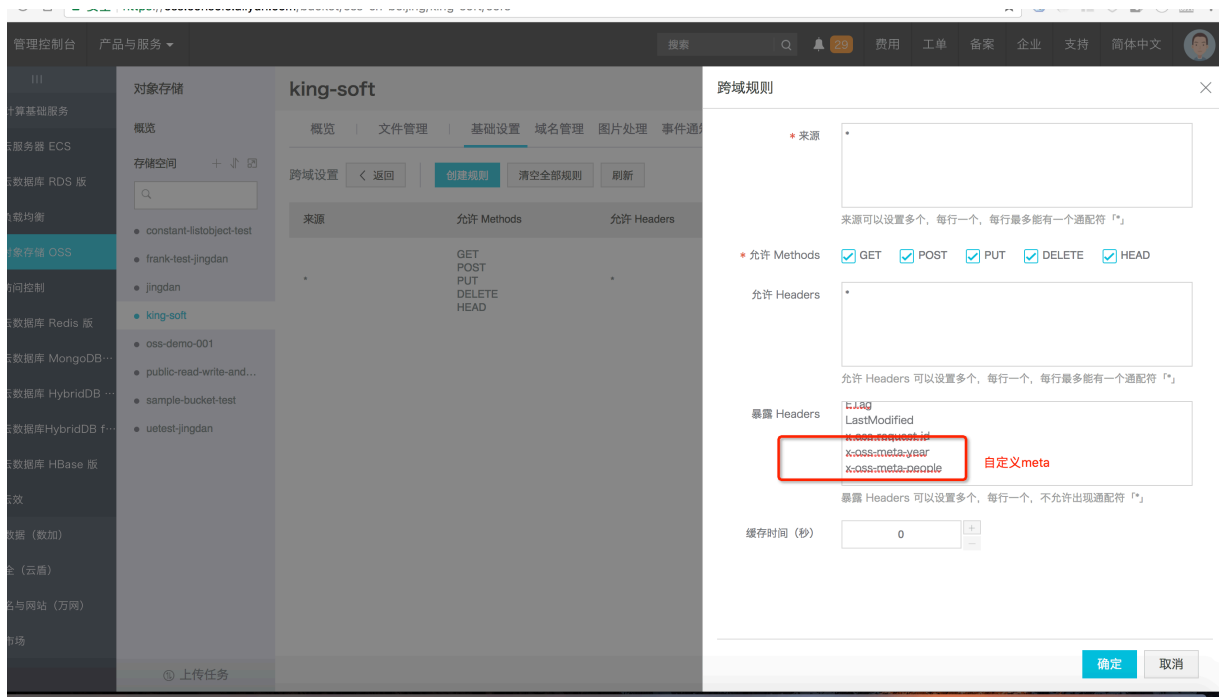
The preceding `progress` parameter is a progress callback function used to get the upload progress.

```

const progress = async function progress ( p , checkpoint )
{
    console . log ( p )
};

```

The preceding `meta` parameter is a user-defined metadata. You can obtain the meta value of the object using the head API, but the meta header must be set properly in the Exposed Headers of the Cross-region Settings in the console, as shown in the following figure:



Returned results of successful requests are as follows:

Request	Status	Method	Response	Size	Time
object?partNumber=39&uploadId=...	200	xhr	Other	534 B	20 ms
object?partNumber=40&uploadId=...	200	xhr	index.js:31702	519 B	6 ms
object?partNumber=40&uploadId=...	200	xhr	Other	533 B	16 ms
object?uploadId=14F3988C71614C...	200	xhr	index.js:31702	568 B	5 ms
object?uploadId=14F3988C71614C...	200	xhr	Other	812 B	107 ms
object	200	xhr	index.js:31702	519 B	5 ms
object	200	xhr	Other	710 B	8 ms

94 requests | 53.0 KB transferred | Finish: 16.76 s | DOMContentLoaded: 321 ms | Load: 335 ms

```

{res: {...}, bucket: "king-soft", name: "object", etag: "'54D4394E73439292340BBE232614A8D6-40'"}
  > {meta: {...}, res: {...}, status: 200}
    > meta:
      > people: "test"
      > year: "2017"
      > __proto__: Object
    > res: {status: 200, statusCode: 200, headers: {...}, size: 0, aborted: false, ...}
    > status: 200
    > __proto__: Object
  
```

You can add `callback` in the option parameter to send callback information.

```

callback : {
  url : ' http :// oss - demo . aliyuncs . com : 23450 ',
  host : ' oss - cn - hangzhou . aliyuncs . com ',
  /* eslint no - template - curly - in - string : [ 0 ] */
  body : ' bucket =${ bucket }& object =${ object }& var1 =${ x :
var1 }',
  contentType : ' application / x - www - form - urlencoded ',
  customValue : {
    var1 : ' value1 ',
    var2 : ' value2 ',
  },
}
  
```

```
    },
  },
}
```

12.5 Download objects

You can download an object from OSS using any of the following methods:

HTTP download (using a browser)

You can use the `signatureUrl` method in the browser to generate the downloadable HTTP address. The validity period of the URL is half an hour by default:

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '<Your region>',
  accessKeyId: '<Your AccessKeyId>',
  accessKeySecret: '<Your AccessKeySecret>',
  bucket: 'Your bucket name'
});

let url = client.signatureUrl('object-key');
console.log(url);

let url = client.signatureUrl('object-key', { expires: 3600 });
console.log(url);
```

12.6 Manage objects

A bucket may have a lot of objects. SDK provides a series of interfaces to help in managing the objects easily.

View all objects

The following code uses `list` to enumerate all objects in the current bucket. Main parameters are described as follows:

- **Prefix:** Specifies only list objects with specific prefixes.
- **Marker:** Specifies list objects whose names are greater than a specified marker.
- **Delimiter:** Obtains the common prefix of a group of objects.
- **Max-keys:** Specifies the maximum number of objects returned.

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '<Your region>',
  accessKeyId: '<Your AccessKeyId>',
  accessKeySecret: '<Your AccessKeySecret>',
```



```

    bucket : ' Your    bucket    name '
  });

  async function list ( dir ) {
    try {
      // Without any parameters . A maximum of 1 , 000
      objects can be returned by default .
      let result = await client . list ();
      console . log ( result );

      // Continue to list objects based on nextMarker
      if ( result . isTruncated ) {
        let result = await client . list ({ marker : result .
nextMarker });
      }

      // List all objects whose names contain the " my -"
      prefix
      let result = await client . list ({
        prefix : ' my -'
      });
      console . log ( result );

      // List all objects whose names are greater than "
      my - object " marker and contain the " my -" prefix
      let result = await client . list ({
        prefix : ' my -',
        marker : ' my - object '
      });
      console . log ( result );
    } catch ( e ) {
      console . log ( e );
    }
  }

  list ();

```

Simulate the directory structure

OSS is an object-based storage service. It has no directories. All objects stored in a bucket are uniquely identified by the key of the object. These objects does not follow any hierarchical structure amongst themselves. Such structure allows OSS to store objects efficiently. However, you may want to put different types of objects under different “directories” to manage objects in a conventional way. OSS provides a “Public Prefix” feature that helps to create a stimulated directory structure. For more information about the public prefix concept, see [View the object list](#).

Assume that the bucket already contains the following objects:

```

foo / x
foo / y
foo / bar / a
foo / bar / b
foo / hello / C / 1
foo / hello / C / 2
...

```

```
foo / hello / C / 9999
```

Next we implement a function called `listDir` to enlist all the objects and subdirectories under the specified directory:

```
let OSS = require('ali-oss');

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function listDir (dir) {
  try {
    let result = await client.list({
      prefix: dir,
      delimiter: '/'
    });

    result.prefixes.forEach(function (subDir) {
      console.log('SubDir: %s', subDir);
    });
    result.objects.forEach(function (obj) {
      console.log('Object: %s', obj.name);
    });
  } catch (e) {
    console.log(e);
  }
}
```

The execution results are as follows:

```
> listDir('foo/')
=> SubDir: foo / bar /
    SubDir: foo / hello /
    Object: foo / x
    Object: foo / y

> listDir('foo / bar /')
=> Object: foo / bar / a
    Object: foo / bar / b

> listDir('foo / hello / C /')
=> Object: foo / hello / C / 1
    Object: foo / hello / C / 2
    ...
    Object: foo / hello / C / 9999
```

Object Meta

When uploading an object to OSS, except for the object content, you have to specify certain attribute information for the object. This attribute information is called meta information. The metadata is stored together with the object during uploading.

This is because the object meta information is stored in HTTP headers during object uploading/downloading, the meta information cannot contain any complex characters according to HTTP.



Note:

Therefore, the meta information must only contain simple printable ASCII characters and cannot contain line breaks. The maximum size of all meta information cannot exceed 8KB.

You can specify the object meta information by specifying the `meta` parameter when using the `put`, `putStream` and `multipartUpload`:

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function multipartUploadWithMeta () {
  try {
    let result = await client.multipartUpload('object - key', 'local - file', {
      meta: {
        year: 2017,
        people: 'mary'
      }
    });
    console.log(result);
  } catch (e) {
    console.log(e);
  }
}

multipartUploadWithMeta();
```

Delete an object

You can use `delete` to delete the object:

```
let OSS = require('ali-oss')

let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

async function delete () {
  try {
    let result = await client.delete('object - key');
    console.log(result);
  }
}
```

```

    } catch ( e ) {
        console . log ( e );
    }
}

delete ();

```

Delete multiple objects

The following code uses `deleteMulti` to delete multiple objects. You can set the `quiet` parameter to determine whether to return the deletion results:

```

let OSS = require ( ' ali - oss ' )

let client = new OSS ( {
    region : '< Your region >',
    accessKeyId : '< Your AccessKeyId >',
    accessKeySecret : '< Your AccessKeySecret >',
    bucket : ' Your bucket name '
} );

async function deleteMulti () {
    try {
        let result = await client . deleteMulti ( [ ' obj - 1 ', '
obj - 2 ', ' obj - 3 ' ] );
        console . log ( result );

        let result = await client . deleteMulti ( [ ' obj - 1 ', '
obj - 2 ', ' obj - 3 ' ], {
            quiet : true
        } );
        console . log ( result );
    } catch ( e ) {
        console . log ( e );
    }
}

deleteMulti ();

```

12.7 CNAME

This topic describes how to bind a custom domain name.

OSS supports binding custom domain names to OSS to seamlessly migrate data storage to the OSS. For example, your domain name is `my-domain.com`, and all your previous image resources are in a format similar to `http://img.my-domain.com/x.jpg`. After you migrate the image storage to OSS, you can still access the images with the original address by binding your custom domain name to OSS:

- Activate OSS and create a bucket.

- Modify the DNS configuration, add a CNAME record, and direct `img.my-domain.com` to an endpoint of OSS (for example, `my-bucket.oss-cn-hangzhou.aliyuncs.com`).
- On [OSS console](#), bind `img.my-domain.com` to the created bucket.
- Upload images to the created bucket in OSS.

Follow the preceding steps to use the original address `http://img.my-domain.com/xx.jpg` to access images on OSS. For more information, see [Bind custom domain names \(CNAME\)](#).

SDK supports the use of a CNAME as the endpoint. To do this, set the `cname` parameter to `true`. For example:

```
let OSS = require('ali-oss')

let client = new OSS({
  endpoint: '< Your endpoint >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  cname: true
});
```



Notice:

The `list_buckets` interface is unavailable when CNAME is used because the custom domain name is bound already to a specific bucket.

12.8 Authorized access

This topic describes how to authorize access to OSS.

Use STS for temporary access authorization

You can use Alibaba Cloud Security Token Service (STS) to authorize temporary access to OSS. STS is a Web service that provides a temporary access token to a cloud computing user. You can use STS to grant a third-party application or a RAM user (whose user ID is managed by yourself) an access credential with customized validity period and permissions. For more information about STS, see [STS introduction](#).

STS has the following advantages:

- You can generate an access token and send it to the third-party application for authorization without exposing your long-term key (AccessKey). You can customize the access permissions and validity period of the token.

- The access token automatically becomes invalid after expiration.

For more information about how to use STS to access OSS, see [Access control](#) in OSS Developer Guide.

Run the following code to create a signed request by using STS:

```
const express = require('express');
const { STS } = require('ali-oss');
const fs = require('fs');

const app = express();
const path = require('path');
const conf = require('./config');

app.get('/sts', (req, res) => {
  console.log(conf);
  let policy;
  if (conf.PolicyFile) {
    policy = fs.readFileSync(path.resolve(__dirname, conf.PolicyFile)).toString('utf-8');
  }

  const client = new STS({
    accessKeyId: conf.AccessKeyId,
    accessKeySecret: conf.AccessKeySecret
  });

  client.assumeRole(conf.RoleArn, policy, conf.TokenExpirationTime).then((result) => {
    console.log(result);
    res.set('Access-Control-Allow-Origin', '*');
    res.set('Access-Control-Allow-METHOD', 'GET');
    res.json({
      AccessKeyId: result.credentials.AccessKeyId,
      AccessKeySecret: result.credentials.AccessKeySecret,
      SecurityToken: result.credentials.SecurityToken,
      Expiration: result.credentials.Expiration
    });
  }).catch((err) => {
    console.log(err);
    res.status(400).json(err.message);
  });
});

app.use('/static', express.static('public'));
app.get('/', (req, res) => {
  res.sendFile(path.resolve(__dirname, '../index.html'));
});

app.listen(9000, () => {
  console.log('App started.');
```

```
});
```

Sign a URL to authorize temporary access

- Sign a URL

You can provide a signed URL to a visitor for temporary access. When you sign a URL, you can specify the expiration time of the URL to restrict the period of access from visitors.

- Sign a URL for an object



Note:

`name { String }` indicates the name of the target object. `[ expires ] { Number }` indicates the expiration period of the URL, which is 1,800 seconds by default. For more information about other parameters, see [Github](#).

Run the following code to sign a URL for an object:

```
const url = store . signatureU rl (' ossdemo . txt ');
console . log ( url );
// -----
const url = store . signatureU rl (' ossdemo . txt ', {
  expires : 3600 ,
  method : ' PUT '
});
console . log ( url );

// put object with signatureU rl
// -----

const url = store . signatureU rl (' ossdemo . txt ', {
  expires : 3600 ,
  method : ' PUT ',
  ' Content - Type ': ' text / plain ; charset = UTF - 8 ',
});
console . log ( url );

// -----
const url = store . signatureU rl (' ossdemo . txt ', {
  expires : 3600 ,
  response : {
    ' content - type ': ' text / custom ',
    ' content - dispositio n ': ' attachment '
  }
});
console . log ( url );

// put operation
```

- Run the following code to sign a URL for image processing:

```
const url = store . signatureU rl (' ossdemo . png ', {
  process : ' image / resize , w_200 '
});
console . log ( url );
```

```
// -----
const url = store . signatureUrl ( ' ossdemo . png ', {
  expires : 3600 ,
  process : ' image / resize , w_200 '
});
console . log ( url );
```

12.9 Set ACLs

OSS allows you to set access permissions for objects. This helps to easily control external access to your resources.

You can set four ACLs for an object:

- **default:** The object inherits the ACL for the bucket it belongs to, that is, the ACL for the object is the same as that for the bucket where the object is stored.
 - **public-read-write:** Anonymous users are allowed to read/write the object.
 - **public-read:** Anonymous users are allowed to read the object.
 - **private:** Anonymous users are not allowed to access objects in the bucket.
- Signature is required for all accesses.

When an object is created, the default permission applies by default. You can use `putACL` to set the ACL for the object.

```
let OSS = require ( ' ali - oss ' )

let client = new OSS ({
  region : '< Your region >'
  accessKeyId : '< Your AccessKeyId >',
  accessKeySecret : '< Your AccessKeySecret >',
  bucket : '< Your bucket name >'
});

async function getACL () {
  try {
    let result = await client . getACL ( ' my - object ' );
    console . log ( result . acl ); // default

    await client . putACL ( ' my - object ', ' public - read ' );
    let result = await client . getACL ( ' my - object ' );
    console . log ( result . acl ); // public - read
  } catch ( e ) {
    console . log ( e );
  }
}

getACL ();
```



Note:

1. If an object is configured with an ACL policy (not default), the object ACL takes priority during permission authentication when it is accessed. The bucket ACL is ignored.
2. If anonymous access is allowed (public-read or public-read-write is configured for the object), you can directly access the object using a browser.

```
For example : http://bucket-name.oss-cn-hangzhou.aliyuncs.com/object.jpg.
```

For more information about ACL, see [Access Control](#).

12.10 Image processing

OSS Image Service (IMG) is an image processing service that features massive capacity, high security, low costs, and high reliability. By uploading and storing original images in OSS, you can process images anytime, anywhere, on any Internet device through a simple RESTful API. IMG offers image processing APIs.

For more information, see [Image processing](#).

Basic features of Image Service

OSS IMG provides the following features:

- [Obtain image information](#)
- [Image format conversion](#)
- [Resize images](#)
- [Incircle](#)
- [Adaptive orientation](#)
- [Brightness](#)
- [Add watermarks](#): allows you to add images, texts, or image-text watermarks to another image.
- [Customize image processing styles](#)
- Calling multiple image processing features through [Cascading Processing](#)

Usage of Image Service

Image Service uses the standard HTTP GET request for access. All processing parameters are encoded in the QueryString in the URL.

If the ACL for an image is private read and write, only authorized users are allowed can access the image.

- Anonymous access

If the access permission of the image object is `public read`, as shown in the following table, you can access Image Service anonymously.

Bucket ACL	Object ACL
Public-read or Public-read-write	Default
Any ACL	Public-read or Public-read-write

Access Image Service anonymously using the third-level domain in the following format:

```
http ://< yourBucket Name >.< yourEndpoi nt >/< yourObject Name >? x - oss - process = image /< yourAction >,< yourParamV alue >
```

Parameter	Description
bucket	Name of the user' s bucket
endpoint	The access domain name for the data center of the user' s bucket
object	The image object uploaded by the user to the OSS
image	The identifier reserved by Image Service
action	Operations by the user on images, such as scaling, cropping, and rotating

Parameter	Description
param	Parameters corresponding to the operations by the user on images

- Basic operations

For example, scale an image to a width of 100 px and adjust the height based on the ratio.

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- Customize image styles

You can use a level-3 domain name in the following format to access a processed image:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=style/<yourStyleName>
```

- **style**: The identifier reserved by the system for the user's custom style
- **yourStyleName**: specifies the name of a custom image style. It is the name specified in the rule that is created on the OSS console.

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=style/oss-pic-style-w-100
```

- Cascade operations

With the cascading processing, you can implement multiple operations on an image. The format is as follows:

```
http://<yourBucketName>.<yourEndpoint>/<yourObjectName>?x-oss-process=image/<yourAction1>,<yourParamValue1>/<yourAction2>,<yourParamValue2>/...
```

For example:

```
http://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100/rotate,90
```

- Image Service also supports HTTPS access.

For example,

```
https://image-demo.oss-cn-hangzhou.aliyuncs.com/example.jpg?x-oss-process=image/resize,w_100
```

- **Authorized access**

For objects with private permissions, as shown in the following table, you must be authorized to access Image Service.

Bucket ACL	Object ACL
private	default
Any ACL	private

The Image Service URL code generated with signature is as follows:

```
let OSS = require('ali-oss');
let client = new OSS({
  accessKeyId: '<accessKeyId>',
  accessKeySecret: '<accessKeySecret>',
  bucket: '<bucketName>',
  endpoint: '<endpoint>', // for example, http://oss-cn-hangzhou.aliyuncs.com
});
// Expire after 10 minutes. The image processing style is: "image/resize,w_300".
let signUrl = client.signatureUrl('example.jpg', {
  expires: 600, // process: 'image/resize,w_300'
});
console.log("signUrl=" + signUrl);
```



Note:

- Authorized access supports custom styles, HTTPS, and cascading processing.
- Expiration time (expires) is measured in seconds.

- **SDK access**

You can directly use SDK to access and process image objects with any ACLs.



Note:

The image object processing using SDKs supports custom styles , HTTPS , and cascading processing .

- Basic operations

Run the following code to perform basic operations on an image:

```
let OSS = require (' ali - oss ');

let client = new OSS ({
  accessKeyId : '< accessKeyId >',
  accessKeySecret : '< accessKeySecret >',
  bucket : '< bucketName >',
  endpoint : '< endpoint , for example , http :// oss - cn - hangzhou . aliyuncs . com >'
});

// Scale the image .
async function scale () {
  try {
    let result = await client . signatureUrl (' example . jpg ', { expires : 3600 , process : ' image / resize , m_fixed , w_100 , h_100 ' })
  } catch ( e ) {
    console . log ( e );
  }
}

scale ();

// Crop the image .
async function crop () {
  try {
    let result = await client . signatureUrl (' example . jpg ', { expires : 3600 , process : ' image / crop , w_100 , h_100 , x_100 , y_100 , r_1 ' })
  } catch ( e ) {
    console . log ( e );
  }
}

crop ();

// Rotate the image .
async function rotate () {
  try {
    let result = await client . signatureUrl (' example . jpg ', { expires : 3600 , process : ' image / rotate , 90 ' })
  } catch ( e ) {
    console . log ( e );
  }
}

rotate ();

// Sharpen the image .
async function sharpen () {
  try {
    let result = await client . signatureUrl (' example . jpg ', { expires : 3600 , process : ' image / sharpen , 100 ' })
  } catch ( e ) {
```

```

        console . log ( e );
    }
}

sharpen ();

// Add watermarks .
async function watermark () {
    try {
        let result = await client . signatureU rl (' example
. jpg ', { expires : 3600 , process : ' image / watermark ,
text_SGVsb G8g5Zu - 54mH5pyN5Y qhIQ '})
    } catch ( e ) {
        console . log ( e );
    }
}

watermark ();

// Convert the image format .
async function format () {
    try {
        let result = await client . signatureU rl (' example .
jpg ', { expires : 3600 , process : ' image / format , png '})
    } catch ( e ) {
        console . log ( e );
    }
}

format ();

// Obtain image information .
async function info () {
    try {
        let result = await client . signatureU rl (' example .
jpg ', { expires : 3600 , process : ' image / info })
    } catch ( e ) {
        console . log ( e )
    }
}

info ();

```

- Customize image styles

You can run the following code to customize an image style:

```

let OSS = require (' ali - oss ');

let client = new OSS ({
    accessKeyId : '< accessKeyId >',
    accessKeySecret : '< accessKeySecret >',
    bucket : '< bucketName >',
    endpoint : '< endpoint , for example , http :// oss - cn -
hangzhou . aliyuncs . com >'
});

// Custom styles
async function style () {
    try {

```

```

        let result = await client.signatureUrl('example .
jpg ', { expires : 3600 , process : ' style / example - sytle
' });
    } catch ( e ) {
        console . log ( e );
    }
}

style ();

```

**Note:**

You must first log on to the OSS console to add the custom style 'example-style'.

- Cascade operations

Run the following code to perform cascade operations on an image:

```

let OSS = require (' ali - oss ');

let client = new OSS ({
    accessKeyId : '< accessKeyId >',
    accessKeySecret : '< accessKeySecret >',
    bucket : '< bucketName >',
    endpoint : '< endpoint , for example , http :// oss - cn -
hangzhou . aliyuncs . com >'
});

// Cascading processing
async function resize () {
    try {
        let result = await client.signatureUrl (' example .
jpg ', { expires : 3600 , process : ' image / resize , m_fixed
, w_100 , h_100 / rotate , 90 ' })
    } catch ( e ) {
        console . log ( e );
    }
}

resize ();

```

12.11 Browser

This topic describes the application of browsers.

Supported browser

- IE (>=10) and Edge
- Mainstream Chrome/Firefox/Safari versions
- Mainstream Android/iOS/WindowsPhone versions

**Notice:**

For IE, you must introduce the promise library before introducing the SDK:

Support for browsers of earlier versions

The OSS SDK perform operations through File APIs so errors may occur in browsers of earlier versions. You can use third-party plug-ins to implement OSS APIs to perform operations, such as upload and download.

• Settings

- Bucket settings

To access OSS through browser, you must set the **CORS** of the bucket as follows:

- Set Source to `*`.
- Set Method to `GET , POST , PUT , DELETE , HEAD`.
- Set Allowed Headers to `*`.
- Set Expose Header to `ETag`.



Notice:

Set the preceding CORS rule to the first option.

file:///Users/rockuw/freeman/work/oss-in-browser/index.html

OSS in Browser

Upload file

Select file

选择文件 未选择任何文件

Store as

object

Upload

0%

Upload content

Content

Hello, OSS!

Store as

object

Save

List files

Key	Size	LastModified
sts/hello	1964	2016-01-19T07:46:09.000Z
oss	11	2016-01-19T06:55:44.000Z
object	11	2016-01-19T06:55:31.000Z

Refresh

Download file

Object key

object

Save as

filename

Download

Powered by OSS & ali-oss

- STS settings

To prevent exposing the AccessKeyId and AccessKeySecret in the webpage, Alibaba Cloud adopts STS for temporary authorization. The authorization server

is maintained by users. Only authenticated users are allowed to apply for a temporary permission from the authorization server. For more information on how to set RAM users and STS policies, see [Overview](#). For information about how to establish your own authorization servers, see [Node.js STS AppServer](#).

- For example:

Here, we use SDK to develop a webpage app which offers four features:

- Upload objects
- Upload text
- List objects
- Download objects

For complete code, see [oss-in-browser](#). The expected result is as follows:

Let's take upload objects as an example in the following text:

1. Create a webpage

- An object choice box to select the object to be uploaded
- A textbox to input the name of the object to be uploaded to the OSS.
- A button to start the upload.
- A progress bar to display the upload progress

The following code creates four controls. `class = " xxx "` serves to apply the [Bootstrap](#) style and can be ignored here.

```
< div class = " form - group ">
  < label > Select file </ label >
  < input type = " file " id = " file " />
</ div >
< div class = " form - group ">
  < label > Store as </ label >
  < input type = " text " class = " form - control " id = "
object - key - file " value = " object " />
</ div >
< div class = " form - group ">
  < input type = " button " class = " btn btn - primary " id
= " file - button " value = " Upload " />
</ div >
< div class = " form - group ">
  < div class = " progress ">
    < div id = " progress - bar "
      class = " progress - bar "
      role = " progressbar "
      aria - valuenow = " 0 "
      aria - valuemin = " 0 "
      aria - valuemax = " 100 " style = " min - width : 2em
; ">
    0 %
  </ div >
```

```

    </ div >
  </ div >

```

2. Add a style

You can add some styles to the webpage to enhance its look. Here, we use [Bootstrap](#). Add the style in the `< head >` tag as follows:

```

< head >
  < title > OSS   in   Browser </ title >
  < link   rel =" stylesheet "   href =" bootstrap . min . css " />
</ head >

```

`bootstrap . min . css` can be downloaded from the official website of [Bootstrap](#).

3. Add a script

Before adding a script, the webpage is static and does not respond to any click. We can add some JavaScript so that it can upload/download/list objects.

First, introduce the SDK in the `< head >` tag:

```

< head >
  < title > OSS   in   Browser </ title >
  < link   rel =" stylesheet "   href =" bootstrap . min . css " />
  < script   type =" text / javascript "   src =" http ://
gosspublic . alicdn . com / aliyun - oss - sdk - x . x . x . min
. js "></ script >
  < script   type =" text / javascript "   src =" app . js "></
script >
</ head >

```

`app . js` contains the code that implements the object uploading. The content is as follows:

```

const   appServer   = ' http :// localhost : 3000 ' ;
const   bucket   = ' js - sdk - bucket - sts ' ;
const   region   = ' oss - cn - hangzhou ' ;

const   urllib   =   OSS . urllib ;

const   applyToken Do = function ( func ) {
  const   url   =   appServer ;
  return   urllib . request ( url , {
    method : ' GET '
  } ). then ( function ( result ) {
    const   creds = JSON . parse ( result . data ) ;
    const   client = new   OSS ( {
      region :   region ,
      accessKeyId :   creds . AccessKeyId ,
      accessKeySecret :   creds . AccessKeySecret ,
      stsToken :   creds . SecurityToken ,
      bucket :   bucket
    } );
    return   func ( client );
  } );
}

```

```

    });
};

let currentChe ckpoint ;
const progress = async function progress ( p ,
checkpoint ) {
    currentChe ckpoint = checkpoint ;
    const bar = document . getElement ById ( ' progress - bar
    ');
    bar . style . width = `${ Math . floor ( p * 100 )}%`;
    bar . innerHTML = `${ Math . floor ( p * 100 )}%`;
};

let uploadFile Client ;
const uploadFile = function ( client ) {
    if ( ! uploadFile Client || Object . keys ( uploadFile
Client ). length === 0 ) {
        uploadFile Client = client ;
    }

    const file = document . getElement ById ( ' file '). files
[ 0 ];
    const key = document . getElement ById ( ' object - key -
file '). value . trim () || ' object ';
    console . log ( `${ file . name } => ${ key }` );

    const options = {
        progress ,
        partSize : 100 * 1024 ,
        meta : {
            year : 2017 ,
            people : ' test ',
        },
    };

    return client . multipartU pload ( key , file , options ).
then ( ( res ) => {
        console . log ( ' upload success : % j ', res );
        currentChe ckpoint = null ;
        uploadFile Client = null ;
    }). catch (( err ) => {
        if ( uploadFile Client && uploadFile Client . isCancel
()) {
            console . log ( ' stop - upload !' );
        } else {
            console . error ( err );
        }
    });
};

window . onload = function () {
    document . getElement ById ( ' file - button '). onclick =
function () {
        applyToken Do ( uploadFile );
    }
};

```

The following are the steps to uploading an object:

- a. Apply for a temporary permission from the authorization server. The authorization server is a service built by the website developer

to provide temporary authorization for `end` users. Developers can provide different permissions for different users during authorization (see [Overview](#)). For authorization server information, see [GitHub](#). To make it more convenient, the authorization server in the example returns the temporary credential to the user directly.

- b. Create an OSS client using a temporary key.
- c. Upload the selected files through the `multipartUpload` interface and set the progress bar through the `progress` parameter.
- d. Other features

For text content uploading, object list retrieval, and object downloading features, see the example code: [OSS in Browser](#).

12.12 Exception handling

If a request error occurs during SDK usage, an exception is thrown containing the following information:

- `status`: the HTTP status code of the error request
- `code`: OSS error code
- `message`: OSS error message
- `requestId`: the UUID that uniquely identifies the request. Seek help from OSS development engineers by providing this `RequestId` if necessary.

For more information about common error messages in OSS, see [OSS Error Response](#).

Debugging

In a browser environment, set the `localStorage.debug` variable in the console to enable debugging:

```
localStorage.debug = 'ali-oss'
```

12.13 FAQ

This topic describes common questions and answers.

How to call STS

A browser is an untrusted environment. It brings an extremely high risk if you store an `AccessKeyID` and `AccessKeySecret` in a browser.

for OSS interface calls in a browser environment.

After obtaining the STS token, you can initialize the SDK.

```
< script type = " text / javascript ">
$. ajax ( " http : // your_sts_s erver / " , { method : ' GET ' } ,
function ( err , result ) {
let client = new OSS ( {
accessKeyId : result . AccessKeyId ,
accessKeySecret : result . AccessKeySecret ,
stsToken : result . SecurityToken ,
endpoint : '< oss endpoint >' ,
bucket : '< Your bucket name >'
} );
} );
< / script >
```

How to enable HTTPS access

When you initialize the SDK, you can input the following parameters:

- **region:** This parameter is the region used when you apply for the OSS service, such as `oss - cn - hangzhou`. Complete region lists can be viewed in [Regions and endpoints](#).
- **internal:** Used in combination with `region`. If `internal` is set to `true`, visit the intranet node.
- **secure:** Used in combination with `region`. If `secure` is set to `true`, use the HTTPS for access.
- **endpoint:** For example, `http : // oss - cn - hangzhou . aliyuncs . com`. If `endpoint` is specified, the `region` is ignored. You can specify the HTTPS or IP address when using `endpoint`.

How to solve the cross-domain issues when using a browser

Set CORS attributes of buckets before using SDK in a browser. For more information, see bucket settings in [Quick start](#).

How to set the user-defined data (meta), file type (mime), and request header (header) of the file to be uploaded

For more information, see [Multipart upload](#).

Instruction for resumable upload on browser side

Save the checkpoint to a browser localStorage. You can directly use the checkpoint parameter next time to achieve resumable upload.

How to obtain the upload progress

You can obtain upload process when using multipart upload. For more information, see [Upload objects](#).

How to obtain the download progress

The browser does not provide download progress. You can call the `signatureUrl` method to obtain the download address. For more information, see [Download objects](#).

How to upload files to a specified directory

Add the specified directory prefix to the object name to be uploaded. For more information, see [Comparison between OSS and file system](#).

```
let OSS = require('ali-oss')
let client = new OSS({
  region: '< Your region >',
  accessKeyId: '< Your AccessKeyId >',
  accessKeySecret: '< Your AccessKeySecret >',
  bucket: 'Your bucket name'
});

client.multipartUpload('base-dir/' + 'object-key', '
local-file', {
  progress: async function(p) {
    console.log('Progress: ' + p);
  }
});
console.log(result);
}).catch((err) => {
  console.log(err);
});
```

How to upload a Base64-encoded image

Base64 first transcodes images into specified formats, and then calls OSS Upload interface to upload the images. For more information, see [Github examples](#).

```
/**
 * base64 to file
 * @param dataurl base64 content
 * @param filename set up a meaningful suffix, or
you can set mime type in options
 * @returns {File}
 */
const dataURLtoFile = function(dataurl, filename) {
  const arr = dataurl.split(',');
  const mime = arr[0].match(/:(.*?);/)[1];
  const bstr = atob(arr[1]);
  let n = bstr.length;
  const u8arr = new Uint8Array(n);
```

```

    while ( n --) {
        u8arr [ n ] = bstr . charCodeAt ( n );
    }
    return new Blob ([ u8arr ], { type : mime }); // if env
    support File , also can use this : return new File ([
    u8arr ], filename , { type : mime });
};

// client indicates OSS client instances
const uploadBase 64Img = function uploadBase 64Img ( client )
{
    // base64 format content
    const base64Cont ent = ' data : image : xxxxxxxxxx xxx ' ;
    const filename = ' img . png ' ;
    const imgfile = dataURLtoF ile ( base64Cont ent , filename
);
    // key indicates uploaded object key and imgFile
    indicates the returned images after processing by
    dataURLtoF ile
    client . multipartU pload ( key , imgfile ). then (( res ) => {
        console . log ( ' upload success : % j ' , res );
    } ). catch (( err ) => {
        console . error ( err );
    } );
};
};

```

How to limit the size of the uploaded file

The size (in bytes) of the uploaded file can be obtained in the browser according to `document.getElementById( "file" ).files[0].size`. See the web post request.

How to obtain the signed URL of an object

You can call the `signatureU rl` method to obtain the download address. For more information, see [Download objects](#).

How to use a signed URL generated by SDK to upload resources

Signed URLs are often used to authorize third parties to download and upload resources. For the download operations, see the answer for the next question. The `signatureUrl` API provided by SDK is used to return a signed URL and users can use this URL to upload or download resources directly. For more information about how to upload resources using signed URLs, see [Upload resources using signed URLs](#).

How to upload resources to OSS servers using Form-based File Upload

For more information, see [Overview of direct transfer on Web client](#).

Common errors

- [JS-SDK exception](#)
- [OSS error response](#)

13 Ruby

13.1 Installation

This topic describes how to install Ruby SDK. Install SDK with `gem` directly:

- [GitHub](#)
- [API documentation](#)
- [ChangeLog](#)

Prerequisites

- You must have an active Alibaba Cloud OSS account. You must create an AccessKey ID and an AccessKey Secret.
- If you want more information about OSS service and want to activate the OSS service, log on to [OSS Product Homepage](#) to learn more.
- If you have not created an AccessKey ID and an AccessKey Secret, go to [Alibaba Cloud Access Key Management](#) to create an AccessKey.

Installation

Install SDK with `gem` directly:

```
gem install aliyun - sdk
```

If the website <https://rubygems.org> is inaccessible, use the image source provided by Taobao.

```
gem install aliyun - sdk -- clear - sources -- source https  
:// ruby . taobao . org
```

You can also install SDK by using [Bundler](#). Initially, add the following content to the `Gemfile` of your app:

```
gem ' aliyun - sdk ', '~> 0 . 3 . 0 '
```

Run the following command:

```
# ( Optional ). Use the image source provided by Taobao .  
bundle config mirror . https :// rubygems . org https :// ruby  
. taobao . org
```



```
bundle install
```

**Note:**

<https://ruby.taobao.org> is the complete image of rubygems.org and automatically synchronizes with the official source. You can access <https://ruby.taobao.org> if rubygems.org is inaccessible.

Dependencies

- Ruby 1.9.3 and later
- Windows, Linux, or OS X system that supports Ruby

**Notice:**

- Some gems on which the SDK depends are local extensions, and you must install `ruby-dev` to compile locally extended gems.
- The environment for running the SDK-dependent gem (`nokogiri`) for processing XML must have the `zlib` library.

Linux

The following method is used to install the preceding dependencies through Ubuntu in Linux:

```
sudo apt - get install ruby ruby - dev zlib1g - dev
```

The installation methods for various Linux releases (each with a package management tool) are similar.

Windows

1. Access [RubyInstaller](#) to download RubyInstaller. Double-click the installer to install the tool. Select “Add Ruby Executables to Your PATH” during the installation process.

**Note:**

Download RubyInstaller 2.1 or an earlier version. RubyInstaller 2.2 cannot be installed due to [Issues](#).

2. Access [RubyInstaller](#) to download the development kit. Make sure you select the right version. The downloaded kit is a compressed package. First create a folder

named `RubyDev` in the `C:\` root directory, and decompress the package to the folder.

3. Press the shortcut key `Windows + R`, input `cmd`, and press Enter to open the CLI. Then run the following commands:

```
cd C :\ RubyDev
ruby dk . rb init
ruby dk . rb install
```

If “config.yml configuration error” is displayed during the last step “install”, use the text editor to open the `C :\ RubyDev \ config . yml` file and modify the file content as follows:

```
---
- C :/ Ruby21
```

Save the config.yml file and continue `ruby dk . rb install`. “Ruby21” is the Ruby installation directory, which must be specified according to the actual name. Close the CLI once the operation is completed.

4. Press the shortcut key `Windows + R`, input `cmd`, and press Enter to open the CLI. Run the following commands:

```
gem install aliyun - sdk
```

When installation is completed, input `irb` to open the Ruby interactive CLI, and input `require ' aliyun / oss '`. The SDK is successfully installed if “true” is displayed.

OS X

1. OS X has Ruby installed by default. To facilitate proper usage and management, we recommend you to install a development version.
2. Run `xcode - select -- install` on your terminal to install “Xcode command line tools”. If installation fails, you can manually download and install the tools (see the steps to download).
3. Download Xcode command line tools from [Apple’ s Developer Center](#) using your Apple ID. Make sure that your selected version matches your system version. After download, double-click the dmg file to load the file. Double-click the installation program in the dialog box to install the tools. Enter your system password during the installation process.

4. Run the following command on your terminal to install Brew:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

5. Run the following command on your terminal to install Ruby:

```
brew install ruby
exec $SHELL -l
```

6. Run the following command on your terminal to install SDK:

```
gem install aliyun-sdk
```

7. Run the following command on your terminal to check the installation status of SDK.

```
irb
> require 'aliyun/oss'
=> true
```

13.2 Quick start

This topic describes how to use the OSS Ruby SDK to access OSS services to view the bucket list and object list, to upload, download, and delete the objects.

For usage convenience, the following operations are performed in `irb` which is Ruby's interactive command-line interface (CLI).

Initialize the client

Input the following content on the CLI and press the Enter:

```
irb
```

Enter the Ruby interactive CLI mode. Introduce the SDK package with `require` :

```
> require 'aliyun/oss'
=> true
```



Note:

In the following demonstration, the `>` symbol introduces the command that the user enters, and the `=>` symbol introduces the content returned by a program.

Run the following command to create an OSSClient:

```
> client = Aliyun::OSS::Client.new (
>   endpoint : 'endpoint',
>   access_key_id : 'AccessKeyId',
```

```
> access_key _secret : ' AccessKeyS ecret ' )
=> #< Aliyun :: OSS :: Client ...
```

Replace the parameters with the actual endpoint, AccessKeyID, and AccessKeySecret.

View the bucket list

Run the following command to view the bucket list:

```
> buckets = client . list_bucke ts
=> #< Enumerator ...
> buckets . each { | b | puts b . name }
=> bucket - 1
=> bucket - 2
=> ...
```

If the bucket list is empty, run the following command to create a bucket:

```
> client . create_buc ket (' my - bucket ')
=> true
```



Note:

- For bucket naming conventions, see [Basic OSS Concepts](#).
- The bucket name cannot be the same with existing bucket names of other users in OSS. To create a bucket successfully, select a unique bucket name.

View the object list

You can run the following command to view the objects in a bucket:

```
> bucket = client . get_bucket (' my - bucket ')
=> #< Aliyun :: OSS :: Bucket ...
> objects = bucket . list_objec ts
=> #< Enumerator ...
> objects . each { | obj | puts obj . key }
=> object - 1
=> object - 2
=> ...
```

Upload an object

You can run the following command to upload a file to a bucket:

```
> bucket . put_object (' my - object ', : file => ' local - file ')
=> true
```

Specifically, `local - file` indicates the path to the local file to be uploaded. When the file is uploaded, you can view it using `list_objec ts`:

```
> objects = bucket . list_objec ts
=> #< Enumerator ...
> objects . each { | obj | puts obj . key }
```

```
=> my - object  
=> ...
```

Download an object

You can run the following command to download an object from a bucket:

```
> bucket . get_object (' my - object ', : file => ' local - file ')  
=> #< Aliyun :: OSS :: Object ...
```

Specifically, `local - file` is the path to the downloaded object. After the object is downloaded, you can open the object to view its content.

Delete an object

You can run the following command to delete an object from a bucket:

```
> bucket . delete_obj ect (' my - object ')  
=> true
```

After deletion, you can run `list_objec ts` to check whether the object has been successfully deleted from the bucket.

```
> objects = bucket . list_objec ts  
=> #< Enumerator ...  
> objects . each { | obj | puts obj . key }  
=> object - 1  
=> ...
```

Learn more

- [Manage a bucket](#)
- [Upload objects](#)
- [Download objects](#)
- [Manage objects](#)
- [Integration with Rails](#)
- [CNAME](#)
- [Use STS to access OSS](#)
- [Set access permissions](#)
- [Manage lifecycle rules](#)
- [Set logging](#)
- [Static website hosting](#)
- [Anti-leech](#)
- [CORS](#)

- [Exception handling](#)

13.3 Manage buckets

A bucket is a namespace on the OSS and a management entity for billing, access control, logging, and other advanced features.

View all buckets

The `Client # list_buckets` interface is used to list all buckets of the current user. Set the `: prefix` parameter to list only the buckets with the specific prefix in the name.

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

buckets = client.list_buckets
buckets.each { |b| puts b.name }

buckets = client.list_buckets(:prefix => 'my -')
buckets.each { |b| puts b.name }
```

Create a bucket

The following code uses the `Client # create_bucket` interface to create a bucket. Specify the name of the bucket to be created:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

client.create_bucket('my - bucket')
```



Note:

- For bucket naming conventions, see [Basic OSS Concepts](#).
- The bucket name has to be globally unique. Make sure your bucket name is different from other users' .

Delete a bucket

The following code uses the `Client # delete_bucket` interface to delete a bucket. Specify the name of the bucket to be deleted:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

client.delete_bucket('my - bucket')
```



Note:

- If the bucket to be deleted has objects in it, delete the objects first.
- If the bucket has an incomplete upload request, use `list_uploads` and `abort_upload` to cancel the request before you delete the bucket. For the usage, see [API Documentation](#).

Check whether a bucket exists

The following code uses the `Client # bucket_exists` interface to check whether a specified bucket of the current user exists:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

puts client.bucket_exists?('my - bucket')
```

Bucket access permissions

You can set the ACL policy of a bucket to allow or prohibit anonymous reads/writes to the bucket. For more information on ACL, see [ACL](#).

- Obtain the ACL of a bucket

The following code uses `Bucket # acl` to view the ACL of a bucket:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my - bucket')
```

```
puts bucket . acl
```

- Set the ACL policy of a bucket

The following code uses `Bucket # acl` to configure the ACL of a bucket:

```
require 'aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key _id : ' AccessKeyI d ', access_key _secret : '
AccessKeyS ecret ')

bucket = client . get_bucket (' my - bucket ')
bucket . acl = Aliyun :: OSS :: ACL :: PUBLIC_REA D
puts bucket . acl
```

13.4 Upload objects

This topic describes how to upload objects.

The OSS Ruby SDK provides rich interfaces for object upload. You can upload an object from OSS through any of the following methods:

- Upload a local file to the OSS
- Stream upload
- Resumable upload
- Append upload
- Upload callback

Upload a local file

The following code uses the `Bucket # put_object` interface with the `: file` parameter specified to upload a local file to OSS:

```
require 'aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key _id : ' AccessKeyI d ', access_key _secret : '
AccessKeyS ecret ')

bucket = client . get_bucket (' my - bucket ')
bucket . put_object (' my - object ', : file => ' local - file ')
```

Stream upload

During uploading a large file, we often need to upload the file in a streaming way instead of processing and uploading all the content at the same time. We want to upload a part of content at a time. In particular, when the content to be uploaded

comes from the Internet and cannot be retrieved at the same time, stream upload is the only option left. Through `Bucket#put_object` interface and specify `block` parameter to upload the stream-generated content to the OSS:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new(
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
bucket.put_object('my-object') do |stream|
  100.times { |i| stream << i.to_s }
end
```

Resumable upload

If the network jitters or the program crashes when a large object is being uploaded, the whole upload operation fails. You have to re-upload the objects, wasting resources. When the network is unstable, you may need to make multiple retries. The `Bucket#resumable_upload` interface can be used to achieve resumable upload. The interface has the following parameters:

- **key:** The name of the object on OSS mapped to the uploaded file.
- **file:** The local path of the file to be uploaded.
- **opts:** An optional parameter. It mainly includes:
 - **:cpt_file:** The path of the checkpoint file. If not specified, the path of the checkpoint object is by default the `file.cpt` under the same directory of the local file. The `file` is the name of the local file.
 - **:disable_cpt:** If this parameter is set to true, the upload progress is not recorded during upload, and resumable upload is not performed when upload fails.
 - **:part_size:** The size of each part. The default size is 4 MB.
 - **&block:** If a block function is imported when the `Bucket#resumable_upload` interface is called, the upload progress is handled by the block function.

For more information about the parameters, see [API documentation](#).

The principle of resumable upload is to divide the object to be uploaded into multiple parts and upload them separately. When all the parts are uploaded, the upload of the entire object is completed. The progress of the current upload is recorded during the uploading process(in the checkpoint object). If the upload of any part fails during the process, the next upload attempt starts from the recorded position in the checkpoint

object. This requires that the same checkpoint object with the previous upload must be used in the next call. When the upload is completed, the checkpoint object is deleted.

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new(
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
bucket.resumable_upload('my-object', 'local-file') do
  |p|
    puts "Progress: #{p}"
end

bucket.resumable_upload(
  'my-object', 'local-file',
  :part_size => 100 * 1024, :cpt_file => '/tmp/x.cpt')
{ |p|
  puts "Progress: #{p}"
}
```



Note:

- The SDK records the upload intermediate states in the cpt object. Therefore, make sure that you have the write permission on the cpt object.
- The cpt object records the intermediate state information of the upload and has a self-checking function. You cannot edit the object. Upload fails if the cpt object is corrupted. When the upload is completed, the checkpoint object is deleted.
- Upload fails if the local file is changed during the upload process.

Append upload

OSS supports the appendable object type. The `Bucket#append_object` interface is used to upload appendable objects. When calling this method, you must specify the append position of the object. If the object is newly created, the append position is 0. If the object already exists, the append position must be set to the object length before the append.

- If the object to be appended does not exist, calling `append_object` creates an appendable object.
- If the object to be appended exists, calling `append_object` appends the content to the end of the object.

```
require 'aliyun / oss'
```

```

client = Aliyun::OSS::Client.new (
  endpoint : 'endpoint ',
  access_key_id : 'AccessKeyId ', access_key_secret : '
AccessKeySecret ')

bucket = client.get_bucket('my-bucket')
# Create an appendable object.
bucket.append_object('my-object', 0) {}

# Append content to the end of the object.
next_pos = bucket.append_object('my-object', 0) do |
  stream |
    100.times { |i| stream << i.to_s }
  end
next_pos = bucket.append_object('my-object', next_pos, :
  file => 'local-file-1')
next_pos = bucket.append_object('my-object', next_pos, :
  file => 'local-file-2')

```

**Note:**

- You can append content to appendable object (objects created with `append_object`) only.
- You cannot copy appendable objects.

Upload callback

You can specify an “upload callback” when uploading a file. After the file is successfully uploaded to OSS, OSS initiates an HTTP POST request to the server address that you have provided. The request is equal to a notification mechanism. You can perform operations according to the received callback results. For more information about upload callback, see [OSS Upload Callback](#).

Currently, only the `put_object` and `resumable_upload` interfaces of the OSS support upload callback.

```

require 'aliyun/oss'

client = Aliyun::OSS::Client.new (
  endpoint : 'endpoint ',
  access_key_id : 'AccessKeyId ', access_key_secret : '
AccessKeySecret ')

bucket = client.get_bucket('my-bucket')

callback = Aliyun::OSS::Callback.new (
  url : 'http://10.101.168.94:1234/callback ',
  query : { user : 'put_object' },
  body : 'bucket=${bucket}&object=${object}'
)

begin

```

```

    bucket . put_object (' files / hello ', file : '/ tmp / x ',
    callback : callback )
  rescue Aliyun :: OSS :: CallbackError => e
    puts " Callback failed : #{ e . message }"
  end

```

In the preceding code, a file is uploaded through the `put_object` interface with an upload callback specified. The callback body contains the bucket and object information for this upload. Upon receiving the callback, the application server determines that the file has been successfully uploaded to the OSS.

The usage of the `resumable_upload` interface is similar:

```

require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ' )

bucket = client . get_bucket (' my - bucket ')

callback = Aliyun :: OSS :: Callback . new (
  url : ' http :// 10 . 101 . 168 . 94 : 1234 / callback ',
  query : { user : ' put_object ' },
  body : ' bucket =#{ bucket }& object =#{ object }'
)

begin
  bucket . resumable_upload (' files / hello ', '/ tmp / x ',
  callback : callback )
rescue Aliyun :: OSS :: CallbackError => e
  puts " Callback failed : #{ e . message }"
end

```



Note:

- The callback URL must not contain the query string which must be specified in the `: query` parameter.
- In the event that the file is successfully uploaded but callback execution fails, the client throws `CallbackError`. To ignore the error, you must explicitly catch the exception.
- For detailed examples, see [callback.rb](#).
- For servers that support callback, see [callback_server.rb](#).

13.5 Download objects

This topic describes how to download an object from OSS to a local file.

The OSS Ruby SDK provides rich interfaces for object download. You can download an object from OSS through any of the following methods:

- Download to a local file
- Stream download
- Resumable download
- HTTP download (by a browser)

Download an object to a local file

The following code uses the `Bucket # get_object` interface with the `: file` parameter specified to download an object as a local file:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my - bucket')
bucket.get_object('my - object', :file => 'local - file')
```

Stream download

When uploading a large file, we often need to upload the file in a streaming way instead of processing and uploading all the content at the same time. We want to upload a part of content at a time. The following code uses the `Bucket # get_object` interface with the `block` parameter specified to process downloaded content through streaming:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my - bucket')
bucket.get_object('my - object') do |chunk|
  # handle_data(chunk)
  puts "Got a chunk, size: #{chunk.size}."
end
```

```
end
```

Resumable download

If the network jitters or the program crashes when a large object is being downloaded, the whole download operation fails. You have to re-download the objects, wasting resources. When the network is unstable, you may need to make multiple retries. The `Bucket#resumable_download` interface can be used to achieve resumable download. The interface has the following parameters:

- **key**: The name of the object to be downloaded.
- **file**: The local path to the downloaded object.
- **opts**: An optional parameter. It mainly includes:
 - **:cpt_file**: The path of the checkpoint file. If not specified, the path of the checkpoint object is by default the `file . cpt` under the same directory of the local `file`. The file is the name of the local file.
 - **:disable_cpt**: If this parameter is set to true, the download progress is not recorded during download, and resumable upload is not performed when upload fails.
 - **:part_size**: The size of each part. The default size is 10 MB.
 - **&block**: If a block function is imported when the `Bucket#resumable_download` interface is called, the download progress is handled by the block function.

For more information about the parameters, see [API Documentation](#).

The principle is to divide the object to be downloaded into multiple parts and download them separately. When all the parts are downloaded, the download of the entire object is completed. The progress of the current download task and downloaded parts is recorded (recorded in the checkpoint file) and downloaded parts (saved as `file . part . N`, among which `file` is the name of the object downloaded to a local path). If one part fails to be downloaded, the re-download starts from the recorded position in the checkpoint object. This requires that the same checkpoint object with the previous download must be used in the next call. The `file.part.N` files and checkpoint object is deleted when download is completed.

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: '
  AccessKeySecret')
```

```

bucket = client . get_bucket ( ' my - bucket ' )
bucket . resumable_ download ( ' my - object ' , ' local - file ' ) do
  | p |
  puts " Progress : #{ p }"
end

bucket . resumable_ download (
  ' my - object ' , ' local - file ' ,
  : part_size => 100 * 1024 , : cpt_file => '/ tmp / x . cpt ' )
{ | p |
  puts " Progress : #{ p }"
}

```

**Note:**

- The SDK records the download intermediate states in the cpt object; therefore, make sure that you have the write permission on the cpt object.
- The SDK saves downloaded parts as file.part.N files; therefore, make sure that you have the file creation permission on the directory where the file.part.N files are stored.
- The cpt object records the intermediate state information of the download and has a self-checking function. You cannot edit the object. Download fails if the cpt object is corrupted.
- If the object to be downloaded is changed during download (the ETag is changed) or a part is missing or modified, an error is reported.

HTTP download

Objects stored in OSS can be downloaded through HTTP directly without using the SDK. HTTP download supports browsers and command-line interface (CLI) tools (such as `wget` and `curl`). The URL of the object to be downloaded is generated by the SDK. The `Bucket # object_url` method is used to generate an HTTP download address which supports the following parameters:

- **key:** The name of the object to be downloaded.
- **sign:** Whether to generate a signed URL. An object with the public-read or public-read-write permission can be accessed without a signed URL. An object with the private permission must be accessed with a signed URL.
- **expiry:** The validity period of the URL. The default value is 60 seconds.

```

require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ' ,

```

```

    access_key_id : 'AccessKeyId', access_key_secret : '
    AccessKeySecret')

    bucket = client.get_bucket('my-bucket')
    # Generate a URL. By default, the generated URL is
    # a signed URL valid for 60 seconds.
    puts bucket.object_url('my-object')
    # http://my-bucket.oss-cn-hangzhou.aliyuncs.com/my-object?Expires=1448349966&OSSAccessKeyId=5vxxxx&Signature=
    # aM2HpBLeMq1aec6JCd7BBAKYiwI%3D

    # Generate an unsigned URL
    puts bucket.object_url('my-object', false)
    # http://my-bucket.oss-cn-hangzhou.aliyuncs.com/my-object

    # Set the URL validity period to one hour (3,600
    # seconds)
    puts bucket.object_url('my-object', true, 3600)

```

13.6 Manage objects

A bucket may contain a lot of objects. The SDK provides a series of interfaces for you to conveniently manage the objects.

View all objects

The following code uses `Bucket#list_objects` to list all objects in the current bucket. Main parameters are described as follows:

- **prefix:** Specify to only list objects with specific prefixes.
- **marker:** Specify to list objects whose names are greater than a specified marker.
- **delimiter:** Obtain the common prefix of objects

```

require 'aliyun/oss'

client = Aliyun::OSS::Client.new(
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: '
  AccessKeySecret')

bucket = client.get_bucket('my-bucket')
# List all objects
objects = bucket.list_objects
objects.each { |o| puts o.key }

# List all objects with the "my-" prefix in the
# name
objects = bucket.list_objects(:prefix => 'my-')
objects.each { |o| puts o.key }

# List all objects that follow the "my-object" marker
# and whose names contain the "my-" prefix
objects = bucket.list_objects(:prefix => 'my-', :marker
=> 'my-object')

```



```
objects . each { | o | puts o . key }
```

Simulate the directory structure

OSS is an object-based storage service. It has no directory. All objects stored in a bucket are uniquely identified by the key of the object. No hierarchical structure is established among these objects. Such a structure allows OSS to efficiently store objects. However, you may want to put different types of objects under different “directories” to manage objects in a conventional way. OSS provides a “Public Prefix” feature for you to conveniently create a stimulated directory structure. For more information about the concept of a public prefix, see [View Object List](#).

Suppose that the bucket already contains the following objects:

```
foo / x
foo / y
foo / bar / a
foo / bar / b
foo / hello / C / 1
foo / hello / C / 2
...
foo / hello / C / 9999
```

Next we will implement a function named `list_dir` to list all objects and subdirectories in a specified directory:

```
require 'aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key _id : ' AccessKeyI d ', access_key _secret : '
AccessKeyS ecret ')

bucket = client . get_bucket (' my - bucket ')

def list_dir ( dir )
  objects = bucket . list_objec ts (: prefix => dir , :
delimiter => '/')
  objects . each do | obj |
    if obj . is_a ?( OSS :: Object ) # object
      puts " Object : #{ obj . key }"
    else # common prefix
      puts " SubDir : #{ obj }"
    end
  end
end
```

The execution results are as follows:

```
> list_dir (' foo /')
=> SubDir : foo / bar /
   SubDir : foo / hello /
   Object : foo / x
   Object : foo / y
```

```
> list_dir (' foo / bar /')
=> Object : foo / bar / a
    Object : foo / bar / b

> list_dir (' foo / hello / C /')
=> Object : foo / hello / C / 1
    Object : foo / hello / C / 2
    ...
    Object : foo / hello / C / 9999
```

Object Meta

When uploading an object to OSS, except for the object content, you can specify certain attribute information for the object. The attribute information is called Object Meta. The metadata is stored together with the object during uploading, and returned together with the object during downloading.

In the SDK, the metadata of an object is expressed by a `Hash`, and the object key and value are both of the `String` type, and only contain simple printable ASCII characters and cannot contain line breaks. The total size of all metadata cannot exceed 8 KB.

When using `Bucket # put_object`, `Bucket # append_object`, and `Bucket # resumable_upload`, you can specify the Object Meta by specifying the `: metas` parameter.

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint : ' endpoint ',
  access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ')

bucket = client.get_bucket (' my - bucket ')

bucket.put_object (
  ' my - object - 1 ',
  : file => ' local - file ',
  : metas => {' year ' => ' 2016 ', ' people ' => ' mary '})

bucket.append_object (
  ' my - object - 2 ', 0,
  : file => ' local - file ',
  : metas => {' year ' => ' 2016 ', ' people ' => ' mary '})

bucket.resumable_upload (
  ' my - object ',
  ' local - file ',
  : metas => {' year ' => ' 2016 ', ' people ' => ' mary '})
```

Run the `Bucket # update_object_metas` command to update Object Meta.

```
require 'aliyun / oss'
```

```

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')

bucket.update_object_metas('my-object', {'year' => '2017'})

```

Copy objects

The following code uses `Bucket#copy_object` to copy an object. When an object is copied, two methods can be used to process the Object Meta. The method used is specified by the `:meta_directive` parameter:

- Method 1: Stay consistent with the source object metadata through the COPY operation.
- Method 2: Overwrite the source object metadata with the new metadata.

```

require 'aliyun/oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')

# Copy the source object metadata
bucket.copy_object(
  'my-object', 'copy-object',
  :meta_directive => Aliyun::OSS::MetaDirective::COPY)

# Overwrite the source object metadata
bucket.copy_object(
  'my-object', 'copy-object',
  :metas => {'year' => '2017'},
  :meta_directive => Aliyun::OSS::MetaDirective::REPLACE)

```

Delete an object

The following code uses `Bucket#delete_object` to delete an object:

```

require 'aliyun/oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')

```

```
bucket . delete_obj  ect (' my - object ')
```

Delete objects in batches

The following code uses `Bucket # batch_delete_object` to delete objects in batches. You can set the `: quiet` parameter to determine whether to return the deletion results:

```
require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key _id : ' AccessKeyI d ', access_key _secret : '
AccessKeyS ecret ')

bucket = client . get_bucket (' my - bucket ')

objs = [' my - object - 1 ', ' my - object - 2 ']
result = bucket . batch_delete_object ( objs )
puts result #[' my - object - 1 ', ' my - object - 2 ']. Names
of successful ly deleted objects are returned by
default .

objs = [' my - object - 3 ', ' my - object - 4 ']
result = bucket . batch_delete_object ( objs , : quiet => true
)
puts result #[]. The deletion result is not returned .
```

13.7 CNAME

This topic describes how to bind a custom domain name.

OSS supports binding custom domain names to OSS to seamlessly migrate data storage to the OSS. For example, your domain name is `my-domain.com`, and all your previous image resources are in a format similar to `http://img.my-domain.com/x.jpg`. After you migrate the image storage to OSS, you can still access the images with the original address by binding your custom domain name to OSS:

1. Activate OSS and create a bucket
2. Bind `img.my-domain.com` to the created bucket.
3. Upload images to the created bucket in OSS.
4. Modify the DNS configuration, add a CNAME record, and direct `img.my-domain.com` to an endpoint of OSS (for example, `my-bucket.oss-cn-hangzhou.aliyuncs.com`)

Follow the preceding steps to use the original address `http://img.my-domain.com/xx.jpg` to access images on OSS. For more information, see [Bind custom domain names \(CNAME\)](#).

SDK supports the use of a CNAME as the endpoint. To do this, set the `:cname` parameter to true. For example:

```
require 'aliyun/oss'

include Aliyun::OSS

client = Client.new(
  endpoint: 'ENDPOINT',
  access_key_id: 'ACCESS_KEY_ID',
  access_key_secret: 'ACCESS_KEY_SECRET',
  cname: true)

bucket = client.get_bucket('my-bucket')
```



Note:

The `list_buckets` interface is unavailable when CNAME is used, because the custom domain name is bound already to a specific bucket.

13.8 Use STS to access OSS

OSS can temporarily grant authorization for access through the Alibaba Cloud STS service.

For more information about STS, see [Concepts](#).

Follow these steps to use the STS:

1. Create a RAM user in the console of the official website. For more information, see [Overview](#).
2. Create an STS role in the console and grant permission to the role of the RAM user. For more information, see [Overview](#).
3. Use the AccessKeyID/AccessKeySecret of the RAM user to apply for a temporary token from STS.
4. Use the authentication information in the temporary token to create an OSS client.
5. Use the OSS client to access OSS.

You must set the `: sts_token` parameter to access OSS through STS. See the following example:

```
require 'aliyun / sts'
require 'aliyun / oss'

sts = Aliyun::STS::Client.new (
  access_key_id: '< AccessKeyId of the RAM user >',
  access_key_secret: '< AccessKeySecret of the RAM user >')

token = sts.assume_role('< role - arn >', '< session - name >')

client = Aliyun::OSS::Client.new (
  endpoint: '< endpoint >',
  access_key_id: token.access_key_id,
  access_key_secret: token.access_key_secret,
  sts_token: token.security_token)

bucket = client.get_bucket('my - bucket')
```

You can customize an STS policy when applying for a temporary token from STS. The requested temporary permission is the intersection of the permission assigned to the role and the permission specified by the STS policy. The following code applies for the read-only permission on `my - bucket` using a specified STS policy and sets the temporary token validity period to 15 minutes:

```
require 'aliyun / sts'
require 'aliyun / oss'

sts = Aliyun::STS::Client.new (
  access_key_id: '< The AccessKeyId of the RAM user >',
  access_key_secret: '< The AccessKeySecret of the RAM user >')

policy = Aliyun::STS::Policy.new
policy.allow(['oss: Get *'], ['acs: oss ::*: my - bucket / *'])

token = sts.assume_role('< role - arn >', '< session name >', policy, 15 * 60)

client = Aliyun::OSS::Client.new (
  endpoint: 'ENDPOINT',
  access_key_id: token.access_key_id,
  access_key_secret: token.access_key_secret,
  sts_token: token.security_token)

bucket = client.get_bucket('my - bucket')
```

For detailed usage and parameters, see [API documentation](#).

13.9 Set ACLs

OSS allows you to set ACLs for buckets and objects respectively. This helps you to conveniently control external access to your resources.

You can set three ACLs for a bucket:

- **public-read-write:** Anonymous users are allowed to create/retrieve/delete objects in the bucket.
 - **public-read:** Anonymous users are allowed to retrieve objects in the bucket.
 - **private:** Anonymous users are not allowed to access objects in the bucket.
- Signature is required for all accesses.

When a bucket is created, the private permission applies by default. You can use

`bucket . acl =` to set the ACL of the bucket.

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
puts bucket.acl
```

You can set four ACLs for an object.

- **default:** The object inherits the ACL for the bucket it belongs to, that is, the ACL for the object is same as that for the bucket where the object is stored.
 - **public-read-write:** Anonymous users are allowed to read/write the object.
 - **public-read:** Anonymous users are allowed to read the object.
 - **private:** Anonymous users are not allowed to access objects in the bucket.
- Signature is required for all accesses.

When an object is created, the default permission applies by default. You can use

`bucket . set_object_acl` to configure the ACL of the object.

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
```

```
acl = bucket.get_object_acl('my-object')
puts acl # default
bucket.set_object_acl('my-object', Aliyun::OSS::ACL::PUBLIC_READ)
acl = bucket.get_object_acl('my-object')
puts acl # public-read
```

**Note:**

- If an object is configured with an ACL policy (not default), the object ACL takes priority during permission authentication when the object is accessed. The bucket ACL is ignored.
- If anonymous access is allowed (public-read or public-read-write is configured for the object), you can directly access the object using a browser.

For example: `http://bucket-name.oss-cn-hangzhou.aliyuncs.com/object.jpg`

- A bucket or an object with the public ACL can be accessed by an anonymous client:

```
require 'aliyun/oss'

# If access_key_id and access_key_secret are not
# specified, an anonymous client will be created.
# The client can access only
# the buckets and objects with the public ACL.
client = Aliyun::OSS::Client.new(endpoint: '
endpoint')
bucket = client.get_bucket('my-bucket')

bucket.get_object('my-object', :file => 'local_file')
```

- For more information about ACL, see [Access Control](#).

13.10 Manage lifecycle rules

OSS allows you to set lifecycle rules for buckets to automatically remove expired objects and save the storage space.

You can configure lifecycle rules for OSS buckets or objects to automatically delete expired objects and parts or change the storage class of expired objects to IA or Archive, saving storage costs. A lifecycle rule includes the following elements:

- **Rule ID:** Identifies a lifecycle rule. A rule ID is unique in a bucket.

- **Strategy:** You can set one of the following two application strategies for a rule. The strategy for all rules set for a bucket must be the same.
 - The rule applies to objects with specified prefixes. You can create multiple rules with this strategy. Prefixes specified in different rules must be different.
 - The rule applies to the entire bucket. You can create only one rule with this strategy.
- **Expiration period:**
 - Specifies the expiration period (N days) of a lifecycle rule. An object expires N days after it is modified for the last time.
 - Specifies an expiration time. Objects modified before the time expire.
- **Status:** Indicates whether the rule is enabled or disabled.

For more information, see [Lifecycle rules](#).

Configure lifecycle rules

You can use `Bucket # lifecycle` to configure lifecycle rules:

```
require 'aliyun / oss '

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket.lifecycle = [
  LifecycleRule.new (
    :id => 'rule1', :enabled => true, :prefix => 'foo /', :
    expiry => 3 ),
  LifecycleRule.new (
    :id => 'rule2', :enabled => false, :prefix => 'bar
  /', : expiry => Date.new ( 2016, 1, 1 ))
]
```

View lifecycle rules

You can use `Bucket # lifecycle` to view lifecycle rules:

```
require 'aliyun / oss '

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

rules = bucket.lifecycle
puts rules
```

Clear lifecycle rules

You can use `Bucket # lifecycle =` to configure an empty rule array to clear lifecycle rules:

```
require 'aliyun / oss '

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket.lifecycle = []
```

13.11 Set logging

OSS allows you to set access logging for a bucket. After the configuration, accesses to the bucket are logged.

The logs are then stored in a specified bucket on OSS in the format of:

```
< TargetPrefix >< SourceBucket >- YYYY - mm - DD - HH - MM - SS -
UniqueString
```

You must specify the `TargetPrefix`. A logging rule consists the following:

- **enable:** Whether to enable access logging.
- **target_bucket:** The bucket where logs are stored.
- **target_prefix:** The prefix of the log object

For more information, see [Set access logging](#).

Enable bucket logging

Use `Bucket # logging =` to enable logging:

```
require 'aliyun / oss '

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')

bucket.logging = BucketLogging.new (
```

```
enable : true , target_bucket : ' logging_bucket ',
target_prefix : ' my - log ')
```

View bucket logging settings

Use `Bucket # logging` to view logging settings:

```
require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ')

bucket = client . get_bucket ( ' my - bucket ' )
log = bucket . logging
puts log . to_s
```

Disable bucket logging

Use `Bucket # logging = to disable logging`.

```
require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ')

bucket = client . get_bucket ( ' my - bucket ' )
bucket . logging = BucketLogging . new ( enable : false )
```

13.12 Static website hosting

You can set your bucket to the static website hosting mode. After the configuration takes effect, you can access this static website with the bucket domain and be redirected to a specified index page or error page.

In [CNAME](#), we mentioned that the OSS allows you to direct your domain name to the OSS service address. In this way, the OSS bucket is accessed when you access your website. You must specify the names of the objects in the bucket corresponding to the homepage (index) and error page (error) of the website respectively.

For more information, see [Static website hosting](#).

Configure website hosting

Use `Bucket # website` to configure hosted pages:

```
require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
```

```

    endpoint : ' endpoint ',
    access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ' )

bucket = client . get_bucket ( ' my - bucket ' )
bucket . website = BucketWebsite . new ( index : ' index . html
', error : ' error . html ' )

```

View website hosting

Use `Bucket # website` to display hosted pages:

```

require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ' )

bucket = client . get_bucket ( ' my - bucket ' )
web = bucket . website
puts web . to_s

```

Clear website hosting

Use `Bucket # website =` to clear hosted pages:

```

require ' aliyun / oss '

client = Aliyun :: OSS :: Client . new (
  endpoint : ' endpoint ',
  access_key_id : ' AccessKeyId ', access_key_secret : '
AccessKeySecret ' )

bucket = client . get_bucket ( ' my - bucket ' )
bucket . website = BucketWebsite . new ( enable : false )

```

13.13 Anti-leech

OSS is a Pay-As-You-Go service. To prevent users' data on the OSS from being leeches, OSS supports anti-leech based on the referer field in the HTTP header.

To prevent your data on OSS from being leeches, OSS supports anti-leeching through the referer field settings in the HTTP header, including the following parameters:

- **Referer whitelist:** Used to allow access only for specified domains to OSS data.
- **Empty referer:** Determines whether the referer can be empty. If it is not allowed, only requests with the referer filed in their HTTP or HTTPS headers can access OSS data.

For more information about the OSS anti-leech features, see [Anti-leech settings](#).

Configure a referer whitelist

The following code uses `Bucket # referer =` to configure a referer whitelist:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
bucket.referer = BucketReferer.new (
  allow_empty: true, whitelist: ['my-domain.com', '*.example.com'])
```

View a referer whitelist

The following code uses `Bucket # referer` to set a referer whitelist:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
ref = bucket.referer
puts ref.to_s
```

Clear a referer whitelist

The following code uses `Bucket # referer =` to clear a referer whitelist:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
bucket.referer = BucketReferer.new (allow_empty: true,
whitelist: [])
```

13.14 CORS

This topic describes how to perform CORS.

Cross-origin resource sharing (CORS) allows web applications to access resources that belong to another region. OSS provides CORS APIs for convenient cross-origin access control.

For more information, see [Cross-origin resource sharing](#) in OSS Developer Guide.

OSS CORS settings contain one or more CORS rules. Each CORS rule includes the following parameters:

- **allowed_origins:** The origins allowed for cross-origin requests, for example, `www.my-domain.com`, `*`.
- **allowed_methods:** The HTTP methods (PUT/POST/GET/DELETE/HEAD) allowed for cross-origin requests.
- **allowed_headers:** The headers allowed in a prefetch command (OPTIONS), for example, `x-oss-test`, `*`.
- **expose_headers:** The headers allowed for the user to access in the application.
- **max_age_seconds:** The cache time for the returned result of a browser prefetch (OPTIONS) request to a specific resource.

Configure CORS rules

The following code uses `Bucket # cors =` to configure a CORS rule:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new(
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
bucket.cors = [
  CORSRule.new(
    :allowed_origins => ['aliyun.com', 'http://www.taobao.com'],
    :allowed_methods => ['PUT', 'POST', 'GET'],
    :allowed_headers => ['Authorization'],
    :expose_headers => ['x-oss-test'],
    :max_age_seconds => 100)
```

```
]
```

View CORS rules

The following code uses `Bucket # cors` to display CORS rules:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
cors = bucket.cors
puts cors.map(&:to_s)
```

Clear CORS rules

The following code uses `Bucket # cors =` to clear CORS rules:

```
require 'aliyun / oss'

client = Aliyun::OSS::Client.new (
  endpoint: 'endpoint',
  access_key_id: 'AccessKeyId', access_key_secret: 'AccessKeySecret')

bucket = client.get_bucket('my-bucket')
bucket.cors = []
```

13.15 Exception handling

When a request error occurs during SDK usage, an exception is thrown, and the error details are logged (saved to the program runtime directory `oss_sdk.log` by default).

The two types of OSS Ruby SDK exceptions are: `ClientError` and `ServerError` and both are the subclasses of `RuntimeError`.

ClientError

`ClientError` is an internal SDK error, such as incorrect parameter setting or object modification during the resumable upload or download.

ServerError

`ServerError` indicates a server error, generated by parsing a server error message. `ServerError` has the following attributes:

- `http_code`: The HTTP status code of the error request.
- `error_code`: OSS error code.

- **message**: OSS error message.
- **request_id**: The UUID that uniquely identifies the request. When you cannot solve the problem, you can seek help from OSS development engineers by providing this RequestId.

For more information, see [OSS error response](#).

13.16 Application of Rails

This topic describes the application of Rails.

Integration with Rails

To use the OSS Ruby SDK in Rails, add the following dependency to `Gemfile` :

```
gem 'aliyun - sdk', '~> 0.3.0
```

Then introduce the dependency when you use OSS:

```
require 'aliyun / oss '
```

The “rails/ “ directory of the SDK provides helper code to facilitate your operation.

The following describes how to use the SDK to implement a simple OSS object manager (oss-manager) which provides the following functions:

- List all the buckets of a user.
- List all the objects in a bucket by level of directory.
- Upload objects.
- Download objects.

The integration process is described as follows:

1. Create a project

Install Rails and create a Rails application oss-manager:

```
gem install rails
rails new oss - manager
```

We recommend, you manage your project code in GitHub:

```
cd oss - manager
git init
git add .
```



```
git commit -m "init project"
```

2. Add the SDK dependency

Add the SDK dependency to `oss - manager / Gemfile` :

```
gem 'aliyun - sdk', '~> 0.3.0'
```

Run the following command in `oss - manager /:`

```
bundle install
```

Save the change in this step:

```
git add .
git commit -m "add aliyun - sdk dependency"
```

3. Initialize an OSSClient

To avoid initializing an OSSClient instance each time you use it in the project, you can add an initialization file to your project.

```
# oss - manager / config / initialize rs / aliyun_oss _init . rb
require 'aliyun / oss'

module OSS
  def self . client
    unless @ client
      Aliyun :: Common :: Logging . set_log_file ('./ log /
oss_sdk . log ')

      @ client = Aliyun :: OSS :: Client . new (
        endpoint :
          Rails . applicatio n . secrets . aliyun_oss [' endpoint
'],
        access_key _id :
          Rails . applicatio n . secrets . aliyun_oss ['
access_key _id '],
        access_key _secret :
          Rails . applicatio n . secrets . aliyun_oss ['
access_key _secret '])
    end

    @ client
  end
end
```

```
end
```

The preceding code can be found in the “rails/” directory of the SDK. The initialization file simplifies the use of the OSSClient in your project:

```
buckets = OSS . client . list_buckets
```

The endpoint, AccessKeyID, and AccessKeySecret are stored in `oss - manager / conf / secrets . yml`. For example:

```
development :
  secret_key_base : xxxx
  aliyun_oss :
    endpoint : xxxx
    access_key_id : aaaa
    access_key_secret : bbbb
```

Save the code:

```
git add .
git commit -m "add aliyun - sdk initialize r"
```

4. Implement the “List Buckets” feature

Firstly, use Rails to generate a controller for bucket management:

```
rails g controller buckets index
```

The following objects are generated in `oss - manager`:

- `app/controller/buckets_controller.rb`: The bucket-related logic code
- `app/views/buckets/index.html.erb`: The bucket-related demonstration code
- `app/helpers/buckets_helper.rb`: Some helper functions

Edit `buckets_controller.rb`. Call the OSSClient to save the `list_buckets` results to the `@buckets` variable:

```
class BucketsController < ApplicationController
  def index
    @buckets = OSS . client . list_buckets
  end
end
```

Edit `views/buckets/index.html.erb` to display the bucket list:

```
< h1 > Buckets </ h1 >
< table class =" table table - striped ">
  < tr >
    < th > Name </ th >
    < th > Location </ th >
    < th > CreationTime </ th >
  </ tr >
```

```

<% @ buckets . each do | bucket | %>
< tr >
  < td ><%= link_to bucket . name , bucket_obj ects_path (
bucket . name ) %></ td >
  < td ><%= bucket . location %></ td >
  < td ><%= bucket . creation_t ime . localtime . to_s %></ td
>
</ tr >
<% end %>
</ table >

```

The `bucket_obj ects_path` is a helper function is stored in `app/helpers/buckets_helper.rb`:

```

module BucketsHel per
  def bucket_obj ects_path ( bucket_nam e )
    "/ buckets /#{ bucket_nam e }/ objects "
  end
end

```

This is how all the buckets are listed. Before you implement the function, configure a Rails route so that the correct logic can be called after an address is entered in the address bar of the browser. Edit `config / routes . rb` and add the following:

```

resources : buckets do
  resources : objects
end

```

Now, input `rails s` in `oss-manager/` to start the Rails server, and input `http :// localhost : 3000 / buckets /` in the address bar of the browser. The bucket list is displayed.

Finally, save the code.

```

git add .
git commit - m " add list buckets feature "

```

5. Implement the “List Objects” feature

First generate a controller for object management:

```

rails g controller objects index

```

Edit `app/controllers/objects_controller.rb` as follows:

```

class ObjectsCon troller < Applicatio nControlle r
  def index
    @ bucket_nam e = params [: bucket_id ]
    @ prefix = params [: prefix ]
    @ bucket = OSS . client . get_bucket (@ bucket_nam e )
  end
end

```

```

    @ objects = @ bucket . list_objec ts ( : prefix => @ prefix
    , : delimiter => '/')
  end
end

```

The preceding code obtains the bucket name from the URL parameters first. An additional prefix is required to list objects by directory level. Call the `list_objects` interface of the `OSSClient` to obtain the object list. Note that the preceding procedure provide a list of the objects which are delimited by the slash (/) and whose names contain a specified prefix. This also aims to display the object list by directory level. See [Manage objects](#).

Next, edit `app/views/objects/index.html.erb` as follows:

```

< h1 > Objects in <%= @ bucket_name %></ h1 >
< p > <%= link_to 'Upload file ', new_object_path (
  bucket_name , @ prefix ) %></ p >
< table class = " table table - striped ">
  < tr >
    < th > Key </ th >
    < th > Type </ th >
    < th > Size </ th >
    < th > LastModified </ th >
  </ tr >

  < tr >
    < td ><%= link_to '../', with_prefix ( upper_dir ( @ prefix
    ) ) %></ td >
    < td > Directory </ td >
    < td > N / A </ td >
    < td > N / A </ td >
  </ tr >

  <%= @ objects . each do | object | %>
  < tr >
    <%= if object . is_a ?( Aliyun :: OSS :: Object ) %>
    < td ><%= link_to remove_prefix ( object . key , @ prefix
    ) ,
      @ bucket . object_url ( object . key ) %></ td >
    < td ><%= object . type %></ td >
    < td ><%= number_to_human_size ( object . size ) %></ td >
    < td ><%= object . last_modified . localtime . to_s %></ td >
  >
    <%= else %>
    < td ><%= link_to remove_prefix ( object , @ prefix ) ,
    with_prefix ( object ) %></ td >
    < td > Directory </ td >
    < td > N / A </ td >
    < td > N / A </ td >
    <%= end %>
  </ tr >
  <%= end %>

```

```
</ table >
```

In the preceding code, the main logic for listing objects according to the directory structure is as follows:

- a. “../ “ always appears at the beginning to indicate the upper-level directory.
- b. Objects with a common prefix in their names are displayed as directories.
- c. Objects are displayed as files.

The preceding code uses `with_prefix`, `remove_prefix`, and other helper functions which are defined in `app/helpers/objects_helper.rb`:

```
module ObjectsHelper
  def with_prefix ( prefix )
    "? prefix=#{ prefix }"
  end

  def remove_prefix ( key , prefix )
    key . sub (/^#{ prefix }/, '')
  end

  def upper_dir ( dir )
    dir . sub (/^[^\/]+\$/, '') if dir
  end

  def new_object_path ( bucket_name , prefix = nil )
    "/ buckets /#{ bucket_name }/ objects / new /#{ with_prefix (
    prefix )}"
  end

  def objects_path ( bucket_name , prefix = nil )
    "/ buckets /#{ bucket_name }/ objects /#{ with_prefix (
    prefix )}"
  end
end
```

Once you run the preceding code, run `rails s`, and enter `http://localhost:3000/buckets/my-bucket/objects/` in the address bar of the browser. The object list is displayed.

Save the code as usual:

```
git add .
git commit -m "add list objects feature"
```

6. Download objects

Note the the preceding code adds a link to each listed object:

```
< td ><%= link_to remove_prefix ( object . key , @ prefix ),
```

```
@ bucket . object_url ( object . key ) %></ td >
```

The `Bucket # object_url` method is used to generate a temporary URL for an object. For more information, see [Download objects](#).

7. Upload files

You can use either of the following methods to upload files with a server app such as Rails:

- a. Upload a file to the Rails server that uploads the object to the OSS. In this method, the Rails server works as a transit server by copying the object to the OSS. The upload process is inefficient.
- b. Upload a file to the OSS directly with the form and temporary credentials generated by the Rails server.

The first method is relatively simple, similar to the common file upload method. The following describes how to use the second method to upload a file:

Add a `# new` method to `app/controllers/objects_controller.rb` to generate an upload form.

```
def new
  @ bucket_nam e = params [: bucket_id ]
  @ prefix = params [: prefix ]
  @ bucket = OSS . client . get_bucket ( @ bucket_nam e )
  @ options = {
    : prefix => @ prefix ,
    : redirect => ' http :// localhost : 3000 / buckets / '
  }
end
```

Then edit `app/views/objects/new.html.erb` as follows:

```
< h2 > Upload   object </ h2 >

<%= upload_for m ( @ bucket , @ options ) do %>
  < table   class = " table   table - striped " >
    < tr >
      < td >< label > Bucket :</ label ></ td >
      < td ><%= @ bucket . name   %></ td >
    </ tr >
    < tr >
      < td >< label > Prefix :</ label ></ td >
      < td ><%= @ prefix   %></ td >
    </ tr >

    < tr >
      < td >< label > Select   file :</ label ></ td >
      < td >< input   type = " file "   name = " file "   style = "
display : inline " /></ td >
    </ tr >

    < tr >
```

```

      < td    colspan =" 2 ">
        < input    type =" submit "    class =" btn    btn - default "
value =" Upload " />
        < span >& nbsp ;& nbsp </ span >
        <%= link_to ' Back ', objects_pa th (@ bucket_nam e , @
prefix ) %>
      </ td >
    </ tr >
  </ table >
<% end %>

```

Specifically, the `upload_for` `m` is a helper function provided by SDK to generate an upload form. The function is stored in `rails/aliyun_oss_helper.rb` of the SDK.

You must copy the function to the `app/helpers/` directory, run `rails s`, and input the `http://localhost:3000/buckets/my-bucket/objects/new` in the address bar of the browser. The file is uploaded.

Do not forget to save the code:

```

git add .
git commit -m "add upload object feature"

```

8. Add a style

You can add some styles (CSS) to refine the page outlook.

Download [bootstrap](#). Unzip the package and then copy `bootstrap.min.css` to `app/assets/stylesheets/`.

Open `app/views/layouts/application.html.erb` and modify the `yield` line as follows:

```

< div id =" main ">
  <%= yield %>
</ div >

```

In this way, each page is added with a `<div>` with `main` as the ID. Then modify `app/assets/stylesheets/application.css` and add the following content:

```

body {
  text-align : center ;
}

div # main {
  text-align : left ;
  width : 1024px ;
  margin : 0 auto ;
}

```

```
}
```

The preceding modification displays the body content of a webpage in a centered layout. The new style makes the page layout more elegant.

So far, a simple demo has been completed. For a complete demo, see [Alibaba Cloud OSS Rails Demo](#).

14 Media-C

14.1 Preface

Introduction

You can store the video captured by the camera to the on-cloud OSS. However, following are a few limitations:

- Owing to the high risk of the AccessKey ID and AccessKeySecret leak, both cannot be stored permanently on the device.
- The device only allows file uploads and downloads and does not allow administrative permissions such as deletion or configuration changes.
- Webpages are available to manage your own videos.
- Precise control over the device permission is provided.
- The permission for the device has a validity period and thus, the device cannot be held permanently.
- The video files output by the camera is expected to be viewed directly through the HLS protocol.

To provide solutions to the preceding problems, we have launched the OSS MEDIA C SDK built, based on the OSS C SDK version that offers better and convenient solutions for the audio and video industry.

- This article describes how to install and use the OSS MEDIA C SDK and is applicable to the OSS MEDIA C SDK V2.0.1.
- This article assumes that you have activated Alibaba Cloud OSS and created an AccessKey ID and AccessKeySecret.
- If you have not activated or do not know about Alibaba Cloud OSS, log on to the [OSS Product Homepage](#) for more information.
- If you have not created an AccessKey ID and AccessKeySecret, go to [Alibaba Cloud Access Key Management](#) to create the AccessKey.

Download the SDK

- Linux (02/03/2016) V2.0.2
 - SDK package: [aliyun-media-c-sdk-2.0.2.tar.gz](#)
 - Source code: [GitHub](#)
 - Version iterations: [Releases](#)



Note:

The OSS MEDIA C SDK does not support Windows.

- Compatibility

SDKs of 1.x.x versions are not compatible with the list-related interfaces after the interfaces change, but are compatible with the other interfaces.

14.2 Installation

This topic describes how to install Media-C SDK.

Version dependency

- Linux

OSS C SDK = 3.x.x

- Windows

Not supported

Installation in Linux

- Firstly, install the OSS C SDK. For installation steps, see [Install the C SDK](#).
- You can [Download SDK package](#) or download [Source Code](#). The downloaded package includes src, sample, and test directories and the CMakeList.txt file.
- Install to the system directory
 - If the OSS C SDK and its dependencies are all installed under the system directory (/usr/local/ or /usr/) and you want to install the OSS MEDIA C SDK to the system directory too, you can run the following command to compile and install the SDK:

```
cmake .
make
```

```
make install
```

- After the preceding command is successfully run, the OSS MEDIA C SDK is installed to the /usr/local/ directory.
- Install to the custom directory (install the dependency package to the system directory)
 - If the OSS C SDK and its dependencies are all installed under the system directory (/usr/local/ or /usr/), but you want to install the OSS MEDIA C SDK to a custom directory such as /home/user/aliyun/oss/install/, you can run the following command to compile and install the SDK:

```
cmake . - DCMAKE_INSTALL_PREFIX =/ home / user / aliyun /
oss / install / usr / local /
make
make install
```

- Once you run the preceding command, the OSS MEDIA C SDK is installed to the /home/user/aliyun/oss/install/usr/local/ directory.
- Install to the custom directory (install the dependency package to the custom directory)
 - If the OSS C SDK or some dependency packages are installed to a custom directory, you cannot find the header files and library files of these packages by default, when you compile the OSS MEDIA C SDK. You must specify the path when you run the cmake command. For example, if you have installed the OSS C SDK to the /home/user/aliyun/oss/install/ directory, run the following command to compile and install the SDK:

```
cmake . - DCMAKE_INSTALL_PREFIX =/ home / user / aliyun
/ oss / install / usr / local / - DOSS_C_SDK_INCLUDE_DIR =/
home / user / aliyun / oss / install / usr / local / include / -
DOSS_C_SDK_LIBRARY =/ home / user / aliyun / oss / install / usr
/ local / lib / liboss_c_sdk . so
make
make install
```

- Once you run the preceding command, the OSS MEDIA C SDK is installed to the /home/user/aliyun/oss/install/usr/local/ directory.
- Names of other related parameters for other dependency packages include: APR_UTIL_LIBRARY, APR_LIBRARY, CURL_LIBRARY, APR_INCLUDEDIRS, APU_INCLUDEDIRS, OSS_C_SDK_INCLUDE_DIR, and CURL_INCLUDEDIRS.

- Only compile and install the client SDK
 - The client SDK and the server SDK are installed at the same time by default. If you only need to compile and install the client SDK, run the following command:

```
cmake . -DONLY_BUILD_D_CLIENT = ON
make
make install
```

- If you only need to compile and install the server, modify ONLY_BUILD_CLIENT to ONLY_BUILD_SERVER.
- The testing example are compiled only when the client and server SDKs are compiled at the same time.
- Other compilation and installation methods and issues
 - Compile mode: Currently four types are supported, namely Debug, Release, MinSizeRef, and RelWithDebInfo. You can specify a compilation type using the parameter -DCMAKE_BUILD_TYPE. For example, if you want to use Debug mode for compilation, add the parameter -DCMAKE_BUILD_TYPE=Debug: `cmake . -DCMAKE_BUILD_TYPE=Debug`. The Release mode is adopted by default.
 - Debug: No code optimization is performed. GDB is supported. This mode is generally used for program debugging.
 - Release: More advanced optimization. This mode is generally applicable to the production environment.
 - MinSizeRef: This mode generates the smallest library file and is generally used in the embedded environment.
 - RelWithDebInfo: More advanced optimization approach. This mode carries debugging information and is generally used for the production environment.
 - If the “Targets may link only to libraries. CMake is dropping the item” warning is prompted during cmake execution, the cause is that the specified library path is incorrect. The library path must be specified to *.so, such as /path/to/xxx.so.
 - If you want to use the static library of the OSS C SDK, specify the -DOSS_C_SDK_LIBRARY=/home/user/aliyun/oss/install/usr/local/lib/liboss_c_sdk_static.a during cmake execution. Other libraries are similar.
 - If “CMake Error: The following variables are used in this project, but they are set to NOTFOUND.” is prompted when you run the cmake, this is because the corresponding library cannot be found in the default path and must be specified. See `Install to a custom directory`.

14.3 Initialization

This topic describes how to initialize the Media-C SDK environment.

Determine an endpoint

An endpoint is the domain of Alibaba Cloud OSS in a region. It currently supports two formats.

Endpoint type	Description
OSS region address	Address of the region where an OSS bucket is located. For more information about the endpoints in various regions, see Regions and endpoints .
User-defined domain name	Domain name defined by the user, with the CNAME directing to the OSS domain.

- OSS region address

You can use any of the following methods to search for an endpoint mapped to the address of the region where an OSS bucket is located:

- You can query the mapping relationship between the endpoint and the region. For more information, see [Regions and endpoints](#).
- You can log on to [Alibaba Cloud OSS Console](#), open the Bucket Overview page, and find the suffix of the bucket domain. For example, the suffix `oss - cn - hangzhou . aliyuncs . com` of the bucket domain `bucket - 1 . oss - cn - hangzhou . aliyuncs . com` is the endpoint of the bucket on the Internet.

Configure an AccessKey

To access Alibaba Cloud OSS, you must have a valid pair of AccessKeys (including an AccessKeyID and AccessKeySecret) for signature authentication. Do the steps in the following links to obtain the AccessKeys:

- [Register an Alibaba Cloud account](#)
- [Apply for an AccessKey](#)

Use the RAM and STS services

Only temporary authorization is required for uploading files to or downloading files from the device terminal (such as a mobile phone, tablet, or camera), instead of permanent authorization. In this scenario, the RAM and STS services are used.

1. Log on to the [RAM console](#) and activate the RAM service.
2. Create a role. You can grant the AliyunOSSFullAccess and AliyunSTSAssumeRoleAccess permissions to this role.
3. After successful creation, an ARN in the role details in a format similar to acs:ram:xxxx:role/yyyy. This is the role_arn. It is used to get a temporary token in subsequent steps.

Initialization

- For initialization of the client, see [Client operations](#).
- For initialization of the server, see [Server operations](#).

14.4 Client operations

OSS MEDIA C SDK is divided into three parts: the client, the server, and the HLS. We mainly cover operations related to the client. For other operations, see the following documents:

Interface

Operation interfaces related with the client are all placed in oss_media_file.h. The following interfaces are currently provided:

- oss_media_file_open
- oss_media_file_stat
- oss_media_file_tell
- oss_media_file_seek
- oss_media_file_read
- oss_media_file_write
- oss_media_file_close

Next, the functions and use notes for various interfaces are provided in detail.

Basic structs

```
/**
 * Metadata of the OSS MEDIA FILE , including file
 * length , location , and type
 */
typedef struct {
    int64_t length ;
    int64_t pos ;
    char * type ;
} oss_media_ file_stat_ t ;
```

```

/**
 * Attributes of the OSS MEDIA FILE
 */
typedef struct oss_media_ file_s {
    void * ipc ;
    char * endpoint ;
    int8_t is_cname ;
    char * bucket_name ;
    char * object_key ;
    char * access_key_id ;
    char * access_key_secret ;
    char * token ;
    char * mode ;
    oss_media_ file_stat_t _stat ;

    time_t expiration ;
    auth_fn_t auth_func ;
} oss_media_ file_t ;

```

**Note:**

- **type:** The file type: normal or appendable.
- **ipc:** Set the unique identifier of the device. This attribute can be used in auth func as per requirement, and ignored if it is not used.
- **endpoint:** Such as oss-cn-hangzhou.aliyuncs.com.
- **is_cname:** Whether CNAME has been enabled.
- **bucket_name:** Name of the OSS bucket.
- **object_key:** Name of the OSS object.
- **access_key_id:** The AccessKey ID provided by Alibaba Cloud for access control . Here two modes are available: 1. The AccessKey ID of the master account or the subaccount. In this case, the token that follows must be set to NULL. 2. The temporary AccessKey ID obtained through get_token.
- **access_key_secret:** The AccessKey Secret provided by Alibaba Cloud for access control. The two available modes are: 1. The permanent AccessKey Secret of the master account or the subaccount. In this case, the token that follows must be set to NULL. 2. The temporary AccessKey Secret obtained through get_token. The access_key_id and access_key_secret must use the same mode at the same time.
- **token:** If the permanent AccessKey ID and AccessKey Secret of the master account or the subaccount are used at the same time, this attribute must be set to NULL. If you use a terminal device to upload or download resources, the app server needs to grant the temporary access permission to the terminal device. You can get the temporary access_key_id, access_key_secret and token through the get_token on the server. If you set token to a non-NULL value, the system expects you to use a

temporary AccessKey ID, a temporary AccessKey Secret and a temporary token. If you do not want to use a temporary token, please set this attribute to NULL.

- **expiration:** The time when the authorization fails. After the first authorization, the next authorization is performed only after the expiration time has passed.
- **auth:** The authorization function. You must implement a function and grant values to the host, bucket, and token values within the function.
- **mode:** The read-write mode.

Initialization

```
/**
 * @brief Initialize the OSS media
 * @note This interface should be called first when
 * the program starts to initialize the OSS MEDIA C
 * SDK .
 * @return :
 * If 0 is returned , it indicates the operation
 * was successful .
 * Otherwise , it indicates an error has occurred .
 * Possible causes include : insufficie nt memory , or the
 * APR or CURL versions are too low .
 */
int oss_media_ init ( aos_log_level_t log_level );
```



Note:

For the complete code, see [GitHub](#).

Destruction

```
/**
 * @brief Destroy the OSS meida
 * @note This interface should be called at last
 * when the program ends to destroy the OSS MEDIA C
 * SDK .
 */
void oss_media_ destroy ();
```



Note:

For the complete code, see [GitHub](#).

Open an object

```
/**
 * @brief Open an OSS object
 * @param [ in ] Name of the bucket that stores
 * objects in bucket_name oss
 * @param [ in ] Name of the object in object_key
 * @param [ in ] mode :
 * ' r ' : Read mode
 * ' w ' : Overwrite mode
```



```

*      'a': Append write mode
*      notes: Not allowed to be used in combination
* @param [in] auth_func The authorization function
to set access_key_id / access_key_secret
* @return:
*      If NULL is returned, it indicates success.
Otherwise, it indicates failure.
*/
oss_media_file_t * oss_media_file_open ( char * bucket_name ,
                                         char * object_key ,
                                         char * mode ,
                                         auth_fn_t auth_func );

```

**Note:**

- **mode:** Read-only, overwrite, and append write modes are supported. Combination mode is not supported.
 - **Read mode:** Once you open the object, part of the object content is read-only once. Multiple reads are generally required to read the full content.
 - **Overwrite mode:** After you open the object, the object content is written at a time, in full. Multiple writes are supported, but the latest content overwrites the earlier content.
 - **Append write:** After the object is opened, you can write content to the object in the append mode multiple times.
- For the complete code, see [GitHub](#).

Close an object

```

/**
* @brief Close the OSS object
*/
void oss_media_file_close ( oss_media_file_t * file );

```

**Note:**

For the complete code, see [GitHub](#).

Write an object

```

/**
* @brief Write an object to the OSS.
* @return:
*      If the request is successful, the size of the
written data will be returned. If "-1" is returned
, it indicates the data write failed.
*/

```

```
int64_t  oss_media_ file_write ( oss_media_ file_t * file ,
const void * buf , int64_t  nbyte );
```

Example project:

```
/* Authorizat ion function */
static void auth_func ( oss_media_ file_t * file ) {
    file -> endpoint = " your endpoint ";
    file -> is_cname = 0 ;
    file -> access_key _id = " Access key ID or temporary
access key ID provided by Alibaba Cloud ";
    file -> access_key _secret = " Access key secret or
temporary access key secret provided by Alibaba Cloud
";
    file -> token = " Get the temporary token through the
get_token interface ";

    /* Effective time of this authorizat ion */
    file -> expiration = time ( NULL ) + 300 ;
}

static void write_file () {
    oss_clean ( g_filename );

    int64_t write_size ;
    oss_media_ file_t * file ;
    char * bucket_nam e = "< your bucket name >";
    char * key = "< your object key >";
    char * content = " aliyun oss media c sdk ";

    /* Open an object */
    file = oss_media_ file_open ( bucket_nam e , key , " w ",
auth_func );
    if (! file ) {
        printf ( " open media file failed \ n " );
        return ;
    }

    /* Write an object */
    write_size = oss_media_ file_write ( file , content , strlen
( content ));
    if ( - 1 != write_size ) {
        printf ( " write %" PRIu64 " bytes succeeded \ n ",
write_size );
    } else {
        oss_media_ file_close ( file );
        printf ( " write failed \ n " );
        return ;
    }

    /* Close the object and release the resource */
    oss_media_ file_close ( file );
}
```



Note:

- In the example, the `overwrite (" w ")` mode is used to open an object. If multiple writes are performed, the later write overwrites the previous write. If

you want to activate append write, specify the `append write (" a ")` mode for opening the object.

- For the complete code, see [GitHub](#).

Read an object

```
/**
 * @brief Read the data of a fixed number of
 * nbytes .
 * @note The size of buf should be greater than
 * or equal to nbyte + 1 .
 * @return :
 * If 0 is returned , it indicates the operation
 * is successful .
 * Otherwise , if "- 1 " is returned , it indicates
 * an error may have occurred . Possible causes of the
 * failure include : the object was not opened in the "
 * read - only " mode , the object cannot connect to the
 * OSS , or the object has no permission to read the
 * OSS .
 */
int64_t oss_media_ file_read ( oss_media_ file_t * file , void
* buf , int64_t nbyte );
```

Example project:

```
int ntotal , nread , nbuf = 16 ;
char buf [ nbuf ];
char * content ;
char * bucket_name = "< your bucket name >";
char * key = "< your object key >";

oss_media_ file_t * file ;

/* Open the object */
file = oss_media_ file_open ( bucket_name , key , " r ",
auth_func );
if (! file ) {
    printf (" open media file failed \ n ");
    return ;
}

/* Read the object */
content = malloc ( stat . length + 1 );
ntotal = 0 ;
while (( nread = oss_media_ file_read ( file , buf , nbuf
)) > 0 ) {
    memcpy ( content + ntotal , buf , nread );
    ntotal += nread ;
}
content [ ntotal ] = '\ 0 ' ;

/* Close the object */
oss_media_ file_close ( file );
free ( content );

printf (" oss media c sdk read object succeeded \ n
");
```

}

**Note:**

For the complete code, see [GitHub](#).

Manage objects

OSS MEDIA FILE also supports tell, seek, and stat operations.

```
/**
 * @brief Get the location of the OSS MEDIA FILE
 * @return :
 *         If 0 is returned, it indicates the operation
 *         is successful.
 *         Otherwise, if "- 1" is returned, it indicates
 *         an error may have occurred. Possible causes of the
 *         failure include: the object was not opened in the "
 *         read-only" mode, the object cannot connect to the
 *         OSS, or the object has no permission to read the
 *         OSS.
 */
int64_t oss_media_ file_tell ( oss_media_ file_t * file );

/**
 * @brief Set the OSS MEDIA FILE pointer to the
 *         specified location.
 * @return :
 *         If 0 is returned, it indicates the operation
 *         is successful.
 *         Otherwise, if "- 1" is returned, it indicates
 *         an error may have occurred. Possible causes of the
 *         failure include: the object was not opened in the "
 *         read-only" mode, the object cannot connect to the
 *         OSS, or the object has no permission to read the
 *         OSS.
 */
int64_t oss_media_ file_seek ( oss_media_ file_t * file ,
int64_t offset );

/**
 * @brief Get the metadata of the OSS MEDIA FILE.
 * @return :
 *         If 0 is returned, it indicates the operation
 *         is successful.
 *         Otherwise, if "- 1" is returned, it indicates
 *         an error may have occurred. Possible causes of the
 *         failure include: the object cannot connect to the OSS
 *         , or the object has no permission to read the OSS
 *         .
 */
int oss_media_ file_stat ( oss_media_ file_t * file ,
oss_media_ file_stat_ t * stat );
```

Example project:

```
static void seek_tell_ stat_file () {
    int ntotal, nread, nbuf = 16 ;
    char buf [ nbuf ];
```

```

char * bucket_name = "< your bucket name >";
char * key = "< your object key >";

oss_media_file_t * file ;

/* Open the object */
file = oss_media_file_open ( bucket_name , key , " r ",
auth_func );
if (! file ) {
    printf ( " open media file failed \n " );
    return ;
}

/* Get the meta information of the object */
oss_media_file_stat_t stat ;
if ( 0 != oss_media_file_stat ( file , & stat ) ) {
    oss_media_file_close ( file );
    oss_media_file_free ( file );

    printf ( " stat media file [%s] failed \n ", file ->
object_key );
    return ;
}

printf ( " file [ name = %s , length = %ld , type = %s ]",
file -> object_key , stat . length , stat . type );

/* tell */
printf ( " file [ position = %ld PRId64 ]", oss_media_
file_tell ( file ) );

/* seek */
oss_media_file_seek ( file , stat . length / 2 );

/* Close the object */
oss_media_file_close ( file );

printf ( " oss media c sdk seek object succeeded \n
");
}

```

**Note:**

For the complete code, see [GitHub](#).

14.5 Server operations

OSS MEDIA C SDK has three parts: the client, the server, and the HLS. Next we introduce the operations related to the server. For other operations, see the following chapters.

Interface

Operation interfaces related to the server are all placed in `oss_media.h`. Currently the following interfaces are provided:

- `oss_media_create_bucket`
- `oss_media_delete_bucket`
- `oss_media_create_bucket_lifecycle`
- `oss_media_get_bucket_lifecycle`
- `oss_media_delete_bucket_lifecycle`
- `oss_media_delete_file`
- `oss_media_list_files`
- `oss_media_get_token`
- `oss_media_get_token_from_policy`

The Functions and notes of usage of various interfaces are as follows:

Basic structs

```
typedef struct oss_media_ config_s {
    char * endpoint ;
    int  is_cname ;
    char * access_key _id ;
    char * access_key _secret ;
    char * role_arn ;
} oss_media_ config_t ;

typedef struct oss_media_ files_s {
    char * path ;
    char * marker ;
    int   max_size ;

    char * next_marke r ;
    int   size ;
    char ** file_names ;
} oss_media_ files_t ;
```



Note:

- **endpoint:** For example, `oss-cn-hangzhou.aliyuncs.com`.
- **is_cname:** Whether CNAME has been enabled.
- **access_key_id:** The AccessKey ID for access control provided by Alibaba Cloud.
- **access_key_secret:** The AccessKey Secret for access control provided by Alibaba Cloud.
- **role_arn:** The ARN of the role created in Alibaba Cloud RAM. The value is available in RAM console > Manage Roles > click the name of a role > Basic Information > ARN. The value format is similar to: `acs:ram::xxxxxxx:role/yyyy`. If no role is available, you must create a new role and grant the `AliyunOSSFullAccess` and `AliyunSTSAssumeRoleAccess` permissions to the role.

- **marker:** Set the returned results to begin from the first entry after the Marker in alphabetical order.
- **max_size:** Set the maximum number returned. The value cannot exceed 1,000.
- **next_marker:** The start position of the next execution.

Initialization

```
/**
 * @brief Initialize the OSS media
 * @note This interface must be called first when
 * the program starts to initialize the OSS MEDIA C
 * SDK .
 * @return :
 * If 0 is returned , it indicates the operation
 * is successful .
 * Otherwise , it indicates that an error has
 * occurred . The possible causes of the failure include :
 * insufficient memory , or the APR or CURL versions
 * are too low .
 */
int oss_media_ init ();
```



Note:

For complete code, see [GitHub](#).

Destruction

```
/**
 * @brief Destroy the OSS media
 * @note This interface must be called at last
 * when the program ends to destroy the OSS MEDIA C
 * SDK .
 */
void oss_media_ destroy ();
```



Note:

For the complete code, see [GitHub](#).

Create a bucket

```
/**
 * @brief Create a new bucket
 * @param [ in ] oss_media_ acl_t
 * OSS_ACL_PU BLIC_READ Public read and private
 * write
 * OSS_ACL_PU BLIC_READ Public read and private
 * write
 * OSS_ACL_PU BLIC_READ_ WRITE Public read - write
 * @ Return :
 * If 0 is returned , it indicates the operation
 * is successful .
 * Otherwise , if "- 1 " is returned , it indicates
 * an error may occur . Possible causes of the failure
```

```

include : the object cannot connect to the OSS , or
the object has no permission .
*/
int oss_media_create_bucket ( oss_media_config_t * config ,
const char * bucket_name , oss_media_acl_t acl );

```

Example project:

```

static void init_media_config ( oss_media_config_t * config
) {
    config->endpoint = "your endpoint";
    config->access_key_id = "The access key ID provided
    by Alibaba Cloud";
    config->access_key_secret = "The access key secret
    provided by Alibaba Cloud";
    config->role_arn = "The role ARN provided by
    Alibaba Cloud RAM";
    config->is_cname = 0 ;
}

void create_bucket () {
    int ret ;
    char * bucket_name ;
    oss_media_config_t config ;

    /* Initialize the variable */
    bucket_name = "< your bucket name >";
    init_media_config (& config );

    /* Create the bucket */
    ret = oss_media_create_bucket (& config , bucket_name ,
    OSS_ACL_PRIVATE );

    if ( 0 == ret ) {
        printf (" create bucket [%s] succeeded .\n ",
        bucket_name );
    } else {
        printf (" create bucket [%s] failed .\n ", bucket_name
        );
    }
}

```



Note:

For the complete code, see [GitHub](#).

Delete a bucket

```

/*
 * @brief Delete the bucket
 * @return
 * If 0 is returned , it indicates the operation
is successful .
 * Otherwise , if "- 1 " is returned , it indicates
an error may occur . Possible causes of the failure
include : the object cannot connect to the OSS , or
the object has no permission .
*/

```



```
int  oss_media_ delete_buc ket ( oss_media_ config_t * config ,
const  char * bucket_name );
```

Example project:

```
void  delete_buc ket () {
    int  ret ;
    char * bucket_name ;
    oss_media_ config_t  config ;

    /* Initialize the variable */
    bucket_name = "< your bucket name >";
    init_media _config (& config );

    /* Delete the bucket */
    ret = oss_media_ delete_buc ket (& config , bucket_name );

    if ( 0 == ret ) {
        printf (" delete bucket [%s] succeeded .\n ",
bucket_name );
    } else {
        printf (" delete bucket [%s] failed .\n ", bucket_name
e );
    }
}
```



Note:

For the complete code, see [GitHub](#).

Create lifecycle rules for a bucket

```
/**
 * @brief Create lifecycle rules for the bucket .
 * @note These rules can control the time for
automatica lly deleting objects .
 * @return
 * If 0 is returned , it indicates the operation
is successful .
 * Otherwise , if "- 1 " is returned , it indicates
an error may occur . Possible causes of the failure
include : the object cannot connect to the OSS , or
the object has no permission .
 */
int  oss_media_ create_buc ket_lifecy cle ( oss_media_ config_t
* config , const  char * bucket_name , oss_media_ lifecycle_
rules_t * rules );
```

Example project:

```
void  create_buc ket_lifecy cle () {
    int  ret ;
    char * bucket_name ;
    oss_media_ lifecycle_ rules_t * rules ;
    oss_media_ config_t  config ;

    /* Initialize the variable */
    bucket_name = "< your bucket name >";
    init_media _config (& config );
```

```

/* Create lifecycle rules */
rules = oss_media_create_lifecycle_rules ( 2 );
oss_media_lifecycle_rule_t rule1 ;
rule1 . name = " example - 1 ";
rule1 . path = "/ example / 1 ";
rule1 . status = " Enabled ";
rule1 . days = 1 ;
oss_media_lifecycle_rule_t rule2 ;
rule2 . name = " example - 2 ";
rule2 . path = "/ example / 2 ";
rule2 . status = " Disabled ";
rule2 . days = 2 ;

rules -> rules [ 0 ] = & rule1 ;
rules -> rules [ 1 ] = & rule2 ;

/* Set lifecycle rules of the bucket */
ret = oss_media_create_bucket_lifecycle (& config ,
    bucket_name , rules );

if ( 0 == ret ) {
    printf ( " create bucket [% s ] lifecycle succeeded .\n " ,
        bucket_name );
} else {
    printf ( " create bucket [% s ] lifecycle failed .\n " ,
        bucket_name );
}

/* Release the resources . */
oss_media_free_lifecycle_rules ( rules );
}

```

**Note:**

Example code can be found at [GitHub](#).

Get lifecycle rules for the bucket

```

/**
 * @brief Get lifecycle rules for the bucket
 * @return
 * If 0 is returned , it indicates the operation
 * is successful .
 * Otherwise , if "- 1 " is returned , it indicates
 * an error may have occurred . Possible causes of the
 * failure include : the object cannot connect to the OSS
 * .
 */
int oss_media_get_bucket_lifecycle ( oss_media_config_t *
    config , const char * bucket_name , oss_media_lifecycle_
    rules_t * rules );

```

Example project:

```

void get_bucket_lifecycle () {
    int ret , i ;
    char * bucket_name ;
    oss_media_lifecycle_rules_t * rules ;
    oss_media_config_t config ;

    /* Initialize the variable */

```

```

    bucket_name = "< your bucket name >";
    init_media_config (& config );

    /* Get lifecycle rules */
    rules = oss_media_create_lifecycle_rules ( 0 );
    ret = oss_media_get_bucket_lifecycle (& config ,
    bucket_name , rules );

    if ( 0 == ret ) {
        printf (" get bucket [%s] lifecycle succeeded .\n ",
    bucket_name );
    } else {
        printf (" get bucket [%s] lifecycle failed .\n ",
    bucket_name );
    }

    for ( i = 0 ; i < rules -> size ; i ++ ) {
        printf (">>>> rule : [ name :%s , path :%s , status =%s
    , days =%d ]\n ",
            rules -> rules [ i ]-> name , rules -> rules [ i ]->
    path ,
            rules -> rules [ i ]-> status , rules -> rules [ i ]->
    > days );
    }

    /* Release the resources . */
    oss_media_free_lifecycle_rules ( rules );
}

```

**Note:**

For the complete code, see [GitHub](#).

Delete lifecycle rules for a bucket

```

/**
 * @brief Delete the bucket lifecycle rules
 * @return :
 *     If 0 is returned , it indicates the operation
 *     is successful .
 *     Otherwise , if "- 1 " is returned , it indicates
 *     an error may occur . Possible causes of the failure
 *     include : the object cannot connect to the OSS , or
 *     the object has no permission .
 */
int oss_media_delete_bucket_lifecycle ( oss_media_config_t
* config , const char * bucket_name );

```

Example project:

```

void delete_bucket_lifecycle ()
{
    int ret ;
    char * bucket_name ;
    oss_media_config_t config ;

    /* Initialize the variable */
    bucket_name = "< your bucket name >";
    init_media_config (& config );

    /* Delete lifecycle rules */
}

```

```

    ret = oss_media_delete_bucket_lifecycle (& config ,
    bucket_name );

    if ( 0 == ret ) {
        printf (" delete bucket [%s] lifecycle succeeded .\n ",
    bucket_name );
    } else {
        printf (" delete bucket [%s] lifecycle failed .\n ",
    bucket_name );
    }
}

```

**Note:**

For the complete code, see [GitHub](#).

Delete an object

```

/**
 * @brief Delete a specific object in the bucket
 * @return :
 *         If 0 is returned, it indicates the operation
 *         is successful .
 *         Otherwise, if "- 1" is returned, it indicates
 *         an error may occur . Possible causes of the failure
 *         include: the object cannot connect to the OSS, or
 *         the object has no permission .
 */
int oss_media_delete_file ( oss_media_config_t * config ,
    const char * bucket_name , const char * key );

```

Example project:

```

void delete_file () {
    int ret ;
    oss_media_config_t config ;
    char * file ;
    char * bucket_name ;

    /* Initialize the variable */
    file = " oss_media_file ";
    bucket_name = "<your bucket name >";
    init_media_config (& config );

    /* Delete an object */
    ret = oss_media_delete_file (& config , bucket_name ,
    file );

    if ( 0 == ret ) {
        printf (" delete file [%s] succeeded .\n ", file );
    } else {
        printf (" delete file [%s] lifecycle failed .\n ",
    file );
    }
}

```

**Note:**

For the complete code, see [GitHub](#).

List objects

```
/**
 * @brief List objects in a specific bucket
 * @return :
 * If 0 is returned, it indicates the operation
is successful.
 * Otherwise, if "-1" is returned, it indicates
an error may occur. Possible causes of the failure
include: the object cannot connect to the OSS, or
the object has no permission.
 */
int oss_media_list_files ( oss_media_config_t * config ,
const char * bucket_name , oss_media_files_t * files );
```

Example project:

```
void list_files () {
    int ret, i;
    char * bucket_name;
    oss_media_files_t * files;
    oss_media_config_t config;

    /* Initialize the variable */
    bucket_name = "< your bucket name >";
    init_media_config (& config);

    files = oss_media_create_files ();
    files->max_size = 50;

    /* List objects */
    ret = oss_media_list_files (& config, bucket_name,
files);

    if (0 == ret) {
        printf ("list files succeeded.\n");
    } else {
        printf ("list files lifecycle failed.\n");
    }

    for (i = 0; i < files->size; i++) {
        printf (">>> file name: %s\n", files->file_names [
i]);
    }

    /* Release the resources */
    oss_media_free_files (files);
}
```



Note:

For the complete code, see [GitHub](#).

Get the temporary token

```
/**
 * @brief Get the temporary token
 * @param [in]:
 * mode :
 * 'r': Read mode
```

```

*      'w': Overwrite mode
*      'a': Append write mode
*      expiration: The valid period of the temporary
token. The value ranges from 15 minutes to one hour
. The unit is seconds.
* @return:
*      If 0 is returned, it indicates the operation
is successful.
*      Otherwise, if "-1" is returned, it indicates
an error may occur. Possible causes of the failure
include: the object cannot connect to the Alibaba
Cloud STS, the RAM or STS is not activated, no
role is created and the parameter is invalid.
*/
int oss_media_get_token ( oss_media_config_t * config ,
                        const char * bucket_name ,
                        const char * path ,
                        const char * mode ,
                        int64_t expiration ,
                        oss_media_token_t * token );

```

Example project:

```

void get_token () {
    int ret;
    char * bucket_name;
    oss_media_token_t token;
    oss_media_config_t config;

    /* Initialize the variable */
    bucket_name = "< your bucket name >";
    init_media_config (& config);

    /* Get the temporary token */
    ret = oss_media_get_token (& config, bucket_name, "/*",
                                "rwa",
                                3600, & token);

    if (0 == ret) {
        printf ("get token succeeded, access_key_id=%s,
access_key_secret=%s, token=%s\n",
                token.tmpAccessKeyId, token.tmpAccessKey
                eySecret, token.securityToken);
    } else {
        printf ("get token failed.\n");
    }
}

```



Note:

For the complete code, see [GitHub](#).

Get the token through a custom policy

```

/**
* @brief Get the token through the specific custom
policy
* @param [in]:
*      expiration: The valid period of the temporary
token. The value ranges from 15 minutes to one hour
. The unit is second.

```

```

* @ return :
*   If 0 is returned , it indicates the operation
is successful .
*   Otherwise , if "- 1 " is returned , it indicates
an error may occur . Possible causes of the failure
include : the object cannot connect to the Alibaba
Cloud STS , the RAM or STS is not activated , no
role is created and the parameter is invalid .
*/
int oss_media_ get_token_ from_polic y ( oss_media_ config_t *
config ,
                                     const char * policy ,
                                     int64_t expiration ,
                                     oss_media_ token_t * token );

```

Example project:

```

void get_token_ from_polic y () {
    int ret ;
    oss_media_ token_t token ;
    char * policy ;
    oss_media_ config_t config ;

    /* Initialize the variable */
    init_media _config (& config );

    /* Set the custom policy */
    policy = "{\" Version \": \" 1 \", \" Statement \": [{ \" Effect \":
    \" Allow \", \"
        \" Action \": \" * \", \" Resource \": \" * \" } ] }";

    /* Get the token */
    ret = oss_media_ get_token_ from_polic y (& config , policy
    , 3600 , & token );

    if ( 0 == ret ) {
        printf ( " get token succeeded , access_key _id =% s ,
        access_key _secret =% s , token =% s \ n " ,
            token . tmpAccessK eyId , token . tmpAccessK
            eySecret , token . securityTo ken );
    } else {
        printf ( " get token failed .\ n " );
    }
}

```



Note:

For the complete code, see [GitHub](#).

14.6 HLS basic interfaces

The OSS MEDIA C SDK client supports packing the received H.264 or AAC format data into TS or M3U8 format and then write the data to OSS. You can view the video and listen to the audio through the corresponding M3U8 address.

Interface

HLS-related basic interfaces are all located in the `oss_media_hls.h`. Currently the provided interfaces include:

- `oss_media_hls_open`
- `oss_media_hls_write_frame`
- `oss_media_hls_begin_m3u8`
- `oss_media_hls_write_m3u8`
- `oss_media_hls_end_m3u8`
- `oss_media_hls_flush`
- `oss_media_hls_close`

The functions and notes of usage of various interfaces are as follows:

- Basic structs

```
/**
 * OSS MEDIA HLS FRAME metadata
 */
typedef struct oss_media_hls_frame_s {
    stream_type_t stream_type;
    frame_type_t frame_type;
    uint64_t pts;
    uint64_t dts;
    uint32_t continuity_counter;
    uint8_t key : 1;
    uint8_t * pos;
    uint8_t * end;
} oss_media_hls_frame_t;

/**
 * OSS MEDIA HLS descriptive information
 */
typedef struct oss_media_hls_option_s {
    uint16_t video_pid;
    uint16_t audio_pid;
    uint32_t hls_delay_ms;
    uint8_t encrypt : 1;
    char key [ OSS_MEDIA_HLS_ENCRYPT_KEY_SIZE ];
    file_handler_t handler_func;
    uint16_t pat_interval_frame_count;
} oss_media_hls_option_t;

/**
 * OSS MEDIA HLS FILE descriptive information
```



```

*/
typedef struct oss_media_hls_file_s {
    oss_media_file_t * file ;
    oss_media_hls_buf_t * buffer ;
    oss_media_hls_option_s_t options ;
    int64_t frame_count ;
} oss_media_hls_file_t ;

```

**Note:**

- **stream_type:** Stream type. Currently the st_h264 and st_aac types are supported.
- **frame_type:** Frame type. Currently the ft_non_idr, ft_idr, ft_sei, ft_sps, ft_pps and ft_aud are supported.
- **pts:** The display timestamp.
- **dts:** The decoding timestamp.
- **continuity_counter:** The incremental counter. The value ranges from 0 to 15, but the starting value is not necessarily 0. The values must be consecutive.
- **key:** Whether or not it is the key frame.
- **pos:** The starting position (inclusive) of the current frame data.
- **end:** The ending position (exclusive) of the current frame data.
- **video_pid:** The video PID.
- **audio_pid:** The audio PID.
- **hls_delay_ms:** The display latency (milliseconds).
- **encrypt:** Whether to use AES-128 encryption. Currently AES-128 encryption is not supported.
- **key:** The key when encryption is enabled. Currently this feature is not supported.
- **handler_func:** The callback function of the object operation.
- **pat_interval_frame_count:** The number of frames between every two inserted PAT or every two inserted MAT tables.

· **Open an HLS file**

```

/**
 * @brief Open an OSS HLS file
 * @param [ in ] Name of the bucket that stores
objects in bucket_name oss
 * @param [ in ] Name of the object in object_key
oss
 * @param [ in ] auth_func The authorization function
to set access_key_id / access_key_secret
 * @return

```

```

*      If NULL is returned , it indicates success .
Otherwise , it indicates failure .
*/
oss_media_hls_file_t * oss_media_hls_open ( char *
bucket_name , char * object_key , auth_fn_t auth_func );

```



Note:

For the complete code, see [GitHub](#).

- Close an HLS file

```

/**
 * @brief Close the OSS HLS file
 */
int oss_media_hls_close ( oss_media_hls_file_t * file );

```



Note:

For the complete code, see [GitHub](#).

- Write an HLS file

```

/**
 * @brief Write a frame of data in H.264 or
AAC format to the OSS
 * @param [ in ] frame A frame of data in H
.264 or AAC format
 * @param [ out ] file hls file
 * @return :
 *      If 0 is returned , it indicates the operation
is successful .
 *      Otherwise , it indicates an error has occurred .
 */
int oss_media_hls_write_frame ( oss_media_hls_frame_t *
frame , oss_media_hls_file_t * file );

```

Example project:

```

static void write_frame ( oss_media_hls_file_t * file ) {
    oss_media_hls_frame_t frame ;
    FILE * file_h264 ;
    uint8_t * buf_h264 ;
    int len_h264 , i ;
    int cur_pos = - 1 ;
    int last_pos = - 1 ;
    int video_frame_rate = 30 ;
    int max_size = 10 * 1024 * 1024 ;
    char * h264_file_name = "/ path / to / example . h264 " ;

    /* Read the H.264 file */
    buf_h264 = calloc ( max_size , 1 );
    file_h264 = fopen ( h264_file_name , " r " );
    len_h264 = fread ( buf_h264 , 1 , max_size , file_h264 );

    /* Initialize the frame struct */
    frame . stream_type = st_h264 ;
    frame . pts = 0 ;
    frame . continuity_counter = 1 ;
}

```

```

        frame . key = 1 ;

        /* Tranverse the H . 264 data and extract the
        data of every frame , and then write the data into
        the OSS */
        for ( i = 0 ; i < len_h264 ; i ++ ) {
            /* Determine whether the current location is
            the starting point of the data of the next frame
            , that is , the end of the current frame */
            if ( ( buf_h264 [ i ] & 0x0F ) == 0x00 && buf_h264 [ i +
            1 ] == 0x00
                && buf_h264 [ i + 2 ] == 0x00 && buf_h264 [ i + 3 ] ==
            0x01 )
            {
                cur_pos = i ;
            }

            /* If the data of a complete frame can be
            obtained , call the interface to convert the data
            into the HLS format and write the data into the
            OSS */
            if ( last_pos != - 1 && cur_pos > last_pos ) {
                frame . pts += 90000 / video_frame_e_rate ;
                frame . dts = frame . pts ;

                frame . pos = buf_h264 + last_pos ;
                frame . end = buf_h264 + cur_pos ;

                oss_media_hls_write_frame ( & frame , file ) ;
            }

            last_pos = cur_pos ;
        }

        /* Close the object and release the resource */
        fclose ( file_h264 ) ;
        free ( buf_h264 ) ;
    }

```

**Note:**

- For the complete code, see [GitHub](#).
- If Access Unit Delimiter NALs (00 00 00 01 09 xx) is missing in the H.264 data, this NAL is required. Otherwise the audio or video cannot be played on iPad, iPhone, or Safari.
- H.264 frames are separated through 0xX0, 0x00, 0x00, and 0x01. AAC frames are separated through 0xFF and 0x0X.
- When the current frame is a key frame, frame. The key needs to be set to 1.

• Write an M3U8 object

```

/**
 * @ brief    Write the header data of the M3U8
 object
 * @ param [ in ] max_durati on    The maximum duration of
 the TS object

```

```

* @ param [ in ]   sequence      The starting number of
the TS object
* @ param [ out ]  file          M3U8 file
* @ return :
*   If 0 is returned , it indicates the operation
is successful .
*   Otherwise , if "- 1 " is returned , it indicates
an error has occurred .
*/
void oss_media_ hls_begin_ m3u8 ( int32_t max_durati on ,
                                int32_t sequence ,
                                oss_media_ hls_file_t * file );

/**
* @ brief   Write the data of the M3U8 object
* @ param [ in ]   size          Number of the M3U8
items
* @ param [ in ]   m3u8          Details of the M3U8
item
* @ param [ out ]  file          M3U8 file
* @ return :
*   If 0 is returned , it indicates the operation
is successful .
*   Otherwise , if "- 1 " is returned , it indicates
an error has occurred .
*/
int oss_media_ hls_write_ m3u8 ( int size ,
                                oss_media_ hls_m3u8_i nfo_t m3u8
[],
                                oss_media_ hls_file_t * file );

/**
* @ brief   Write data such as the M3U8 object
terminator
* @ param [ out ]  file          M3U8 file
*/
void oss_media_ hls_end_m3 u8 ( oss_media_ hls_file_t * file
);

```

Example project:

```

static void write_m3u8 () {
    char * bucket_name ;
    char * key ;
    oss_media_ hls_file_t * file ;

    bucket_name = "< your bucket name >";
    key = "< your m3u8 file name >";

    /* Open an HLS object to write the M3U8 - format
    data . The object name must end with ". m3u8 " */
    file = oss_media_ hls_open ( bucket_name , key ,
auth_func );
    if ( file == NULL ) {
        printf ( " open m3u8 file [% s ] failed .", key );
        return ;
    }

    /* Construct the informatio n of three TS objects
    */
    oss_media_ hls_m3u8_i nfo_t m3u8 [ 3 ];
    m3u8 [ 0 ]. duration = 9 ;

```

```

    memcpy ( m3u8 [ 0 ]. url , " video - 0 . ts ", strlen (" video
- 0 . ts "));
    m3u8 [ 1 ]. duration = 10 ;
    memcpy ( m3u8 [ 1 ]. url , " video - 1 . ts ", strlen (" video
- 1 . ts "));

    /* Write to the M3U8 object
    oss_media_hls_begin_ m3u8 ( 10 , 0 , file );
    oss_media_hls_write_ m3u8 ( 2 , m3u8 , file );
    oss_media_hls_end_m3 u8 ( file );

    /* Close the HLS object */
    oss_media_hls_close ( file );

    printf (" write m3u8 to oss file succeeded \ n ");
}

```

**Note:**

- M3U8 V3 is currently used.
- In a recording scenario, the `oss_media_hls_end_m3u8(file)` interface is called at the end of the recording to write a terminator. Otherwise the video cannot be played. In a live video scenario, this interface cannot be called.
- For the complete example code, see [GitHub](#).
- You can view the effect in the example project.
- You can watch the video through [VLC Player](#), and through Safari directly on iPhone, iPad, and Mac devices.

14.7 HLS encapsulated interfaces

The OSS MEDIA C SDK client can pack the received H.264 or AAC format data into the TS or M3U8 format and then write the data into the OSS. Apart from basic interfaces, the SDK also provides encapsulated recording and live video interfaces.

Interface

HLS-related encapsulated interfaces are all located in the `oss_media_hls_stream.h`. Currently the following interfaces are provided:

- `oss_media_hls_stream_open`
- `oss_media_hls_stream_write`
- `oss_media_hls_stream_close`

The functions and notes of usage of various interfaces are mentioned as follows:

• Basic structure

```
/**
 * Information about OSS MEDIA HLS STREAM OPTIONS
 */
typedef struct oss_media_hls_stream_options_s {
    int8_t is_live;
    char * bucket_name;
    char * ts_name_prefix;
    char * m3u8_name;
    int32_t video_frame_rate;
    int32_t audio_sample_rate;
    int32_t hls_time;
    int32_t hls_list_size;
} oss_media_hls_stream_options_t;

/**
 * Information about OSS MEDIA HLS STREAM
 */
typedef struct oss_media_hls_stream_s {
    const oss_media_hls_stream_options_t * options;
    oss_media_hls_file_t * ts_file;
    oss_media_hls_file_t * m3u8_file;
    oss_media_hls_frame_t * video_frame;
    oss_media_hls_frame_t * audio_frame;
    oss_media_hls_m3u8_info_t * m3u8_infos;
    int32_t ts_file_index;
    int64_t current_file_begin_pts;
    int32_t has_audio;
    aos_pool_t * pool;
} oss_media_hls_stream_t;
```



Note:

- **is_live:** Indicates whether the video is in live mode. When the video is in live mode, the M3U8 object only contains the information of the last few TS objects.
- **bucket_name:** Indicates the names of the buckets that store the HLS files.
- **ts_name_prefix:** Indicates the prefix of the TS object name.
- **m3u8_name:** Indicates the name of the M3U8 object.
- **video_frame_rate:** Indicates the frame rate of the video data.
- **audio_sample_rate:** Indicates the sample rate of the audio data.
- **hls_time:** Indicates the maximum duration of each TS object.
- **hls_list_size:** Indicates the number of TS objects retained in the M3U8 object in live video mode.

• Open an HLS stream object

```
/**
 * @brief Open an OSS HLS file
 * @param [in] auth_func The authorization function
 * to set access_key_id / access_key_secret
```

```

* @param [ in ] options      Configuratio n informatio n
* @return :
*       If NULL is returned , it indicates success .
*       Otherwise , it indicates failure .
*/
oss_media_hls_stream_t * oss_media_hls_stream_open (
auth_fn_t auth_func ,
const oss_media_hls_stream_options_t
* options );

```

**Note:**

- For the complete example code, see [GitHub](#)
- .

- Close an HLS stream object

```

/**
* @brief Close the HLS stream object
*/
int oss_media_hls_stream_close ( oss_media_hls_stream_t
* stream );

```

**Note:**

For the complete example code, see [GitHub](#).

- Write data into an HLS stream object

```

/**
* @brief Write the video and audio data
* @param [ in ] video_buf Video data
* @param [ in ] video_len Video data length . It can
be 0
* @param [ in ] audio_buf Audio data
* @param [ in ] audio_len Audio data length . It can
be 0
* @param [ in ] stream HLS stream
* @return :
*       If 0 is returned , it indicates the operation
is successful .
*       Otherwise , it indicates an error has occurred
*/
int oss_media_hls_stream_write ( uint8_t * video_buf ,
uint64_t video_len ,
uint8_t * audio_buf ,
uint64_t audio_len ,
oss_media_hls_stream_t * stream
);

```

Example project:

```

static void write_video_audio(void) {
int ret;

```

```

int    max_size = 10 * 1024 * 1024 ;
FILE * h264_file , * aac_file ;
uint8_t * h264_buf , * aac_buf ;
int    h264_len , aac_len ;

oss_media_hls_stream_options_t options ;
oss_media_hls_stream_t * stream ;

/* Set the HLS stream parameter value */
options . is_live = 0 ;
options . bucket_name = "< your bucket name >";
options . ts_name_prefix = "vod / video_audio / test ";
options . m3u8_name = "vod / video_audio / vod . m3u8 ";
options . video_frame_rate = 30 ;
options . audio_sample_rate = 24000 ;
options . hls_time = 5 ;

/* Open the HLS stream */
stream = oss_media_hls_stream_open ( auth_func , &
options );
if ( stream == NULL ) {
    printf ( " open hls stream failed .\n " );
    return ;
}

/* Create two buffers to store the audio and
video data */
h264_buf = malloc ( max_size );
aac_buf = malloc ( max_size );

/* Read a piece of video data and audio data
and call the interface to write data to the OSS
*/
{
    h264_file = fopen ( "/ path / to / video / 1 . h264 " , " r
");
    h264_len = fread ( h264_buf , 1 , max_size , h264_file
);
    fclose ( h264_file );

    aac_file = fopen ( "/ path / to / audio / 1 . aac " , " r
");
    aac_len = fread ( aac_buf , 1 , max_size , aac_file );
    fclose ( aac_file );

    ret = oss_media_hls_stream_write ( h264_buf ,
h264_len ,
        aac_buf , aac_len , stream );
    if ( ret != 0 ) {
        printf ( " write vod stream failed .\n " );
        return ;
    }
}

/* Read a piece of video data and audio data
again and call the interface to write data to the
OSS */
{
    h264_file = fopen ( "/ path / to / video / 2 . h264 " , " r
");
    h264_len = fread ( h264_buf , 1 , max_size , h264_file
);
    fclose ( h264_file );

```



```

");
    aac_len = fread ( aac_buf , 1 , max_size , aac_file );
    fclose ( aac_file );

    ret = oss_media_hls_stream_write ( h264_buf ,
    h264_len ,
        aac_buf , aac_len , stream );
    if ( ret != 0 ) {
        printf ( " write vod stream failed .\n " );
        return ;
    }
}

/* After the data write , close the HLS stream */
ret = oss_media_hls_stream_close ( stream );
if ( ret != 0 ) {
    printf ( " close vod stream failed .\n " );
    return ;
}

/* Release the resources */
free ( h264_buf );
free ( aac_buf );

printf ( " convert H . 264 and aac to HLS vod
succeeded \n " );
}

```



Note:

- Currently the recording and live video interfaces support video-only, audio-only and audio-video modes.
- For the complete example code, see [GitHub](#).
- Recording and live video interfaces are still relatively basic. If you need advanced features, use the basic interfaces to simulate the two interfaces to implement advanced custom features by yourself.
- You can view the effect in the example project.

14.8 Scenarios

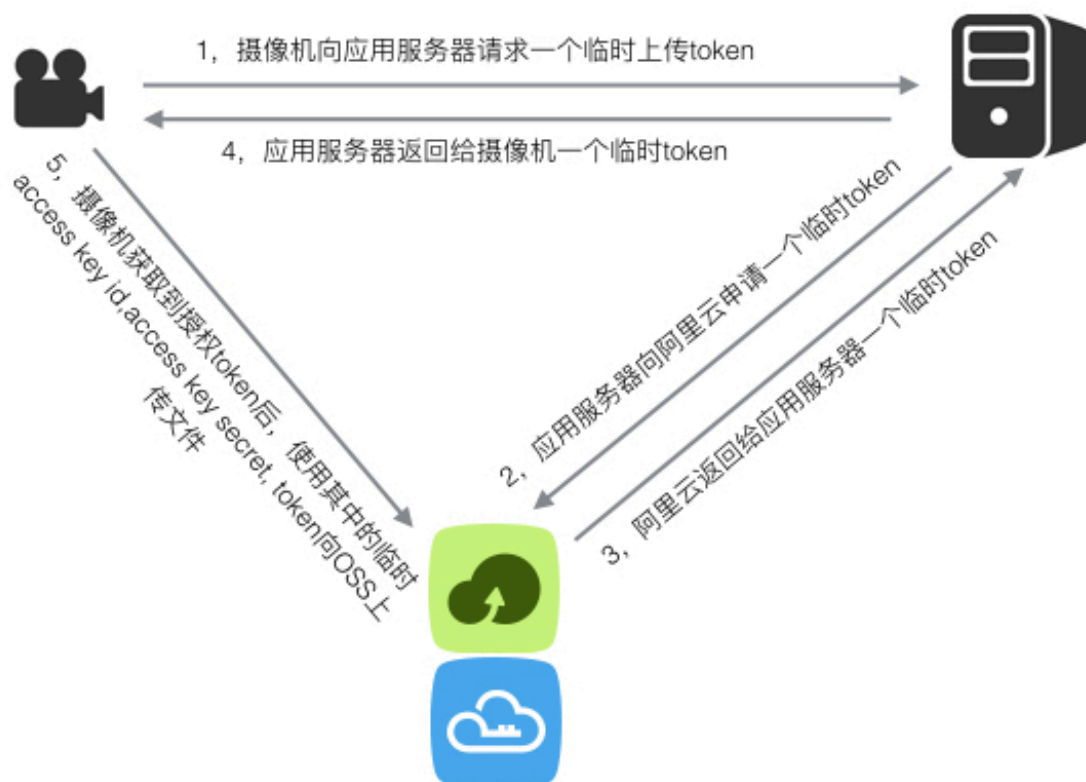
The previous two sections described relative operations on the [client](#) and on the [server](#). This section explains how to connect the client and server for use. We

recommend, that you to read previous two sections before you proceed with this section.

Video monitoring

In the preface, we have introduced the devices like network cameras and its functions and usage are described as follows:

- Procedure



This function involves the three roles such as the network camera, the app server, and Alibaba Cloud Object Storage Service (OSS). The network camera uses the client part of the OSS MEDIA C SDK and the app server uses the server part of the OSS C MEDIA SDK.

Following is the procedure to upload a video clip captured by a network camera:

1. To upload a video clip on the OSS, the network camera sends a network request to the app server to get authorization.
2. After the app server receives the request, it performs a check to confirm that the upload permission can be granted to the network camera, and then requests a

token with a specific validity period and with only the upload permission from Alibaba Cloud through the `get_token` interface of the OSS MEDIA C SDK.

3. After Alibaba Cloud receives the request from the app server to get a token, it checks the user configuration. If the check verifies that the request can be allowed, it issues a temporary token (including the temporary AccessKey ID, the temporary AccessKey Secret, and a temporary STS token) with only the upload permission to the OSS and only validity for a specified duration and sends the token to the app server.
4. The app server forwards the token to the network camera once the temporary token is received.
5. After the network camera gets the token, you can upload video files to the OSS through the `oss_media_write` interface of the OSS MEDIA C SDK client.
6. You can also use the server part of the C SDK or use Java, C#, Go, Python, PHP, Ruby SDKs on the app server to implement an HTTP service. This helps the other users to view and manage various video files on the webpage.

• Example code

A simple example project simulating operations on the client and on the server is as follows:

```
char * global_tem p_access_k ey_id = NULL ;
char * global_tem p_access_k ey_secret = NULL ;
char * global_tem p_token = NULL ;

/* Authorizat ion function */
static void auth_func ( oss_media_ file_t * file ) {
    file -> endpoint = " your endpoint ";
    file -> is_cname = 0 ;
    file -> access_key _id = global_tem p_access_k ey_id ;
    file -> access_key _secret = global_tem p_access_k
ey_secret ;
    file -> token = global_tem p_token ;

    /* Effective time of this authorizat ion */
    file -> expiration = time ( NULL ) + 300 ;
}

/* Simulate sending a token to the client from the
server */
static void send_token _to_client ( oss_media_ token_t
token ) {
    global_tem p_access_k ey_id = token . tmpAccessK eyId ;
    global_tem p_access_k ey_secret = token . tmpAccessK
eySecret ;
    global_tem p_token = token . securityTo ken ;
}

void get_and_us e_token () {
    oss_media_ token_t token ;
```

```

/* Server logic : Get a temporary token from
Alibaba Cloud and send it to the client */
{
    int ret ;
    char * policy = NULL ;
    oss_media_config_t config ;

    policy = "{\n "
        "\" Statement \": [\n "
        "{\n "
        "\" Action \": \" oss :*\",\n "
        "\" Effect \": \" Allow \",\n "
        "\" Resource \": \"*\",\n "
        "}\n "
        "],\n "
        "\" Version \": \" 1 \"\n "
        "}\n " ;

    init_media_config (& config );

/* Request a temporary authorization token from
Alibaba Cloud */
    ret = oss_media_get_token_from_policy (& config ,
        policy ,
        17 * 60 , & token );

    if ( ret != 0 ) {
        printf (" Get token failed .");
        return ;
    }

/* Simulate sending the temporary token to the
client */
    send_token_to_client ( token );
}

/* Client logic : Use the temporary token to
operate on files after the temporary token is
obtained from the server */
{
    int ret ;
    int64_t write_size = 0 ;
    oss_media_file_t * file = NULL ;
    char * content = NULL ;
    char * bucket_name ;
    char * object_key ;
    oss_media_file_stat_t stat ;

    content = " hello oss media file \n " ;
    bucket_name = "< your bucket name >";
    object_key = " key " ;

    /* Open the object */
    file = oss_media_file_open ( bucket_name ,
object_key , " w " , auth_func );
    if ( file != NULL ) {
        printf (" open file failed .");
        return ;
    }

    /* Write an object */
    write_size = oss_media_file_write ( file , content ,
strlen ( content ));

```

```
        if ( write_size != strlen ( content )) {  
            printf ( " write   file   failed .");  
            return ;  
        }  
  
        /* Close the object and release the resource  
*/  
        oss_media_ file_close ( file );  
    }  
}
```

**Note:**

- Log on to the [RAM console](#). Click Roles, from the left-side navigation pane. Locate the role and click Manage. Go to the Basic Information page, the policy is available.
- If precise control of permissions is not required, you can use the simpler `oss_media_get_token` interface. The path parameter can be `/*`, and the mode parameter can be `rwa`.
- For more information, see the [RAM and STS Guide](#).