Alibaba Cloud
Object Storage Service

API Reference

MORE THAN JUST CLOUD | ⊂–⊃ Alibaba Cloud

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.	📋 **Note:** 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 `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` `[-all|-t]`
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` `{stand | slave}`

Contents

1 OSS API documentation overview

The Object Storage Service (OSS) is a cloud storage service provided by Alibaba Cloud, featuring a massive capacity, security, a low cost, and high reliability. You can upload and download data anytime, anywhere, and on any Internet device through a simple RESTful interface described herein. With the OSS, you can develop a diverse range of massive data-based services such as multimedia sharing websites, online storage, personal data backups, and corporate data backups. This document details the request syntax, request samples and return samples for each interface and precautions for using interfaces to help you with quick secondary development and third-party technology integration.

Before using these interfaces, make sure that you fully understand the OSS product instructions, usage agreements, and billing methods.

2 API overview

The API interfaces provided by OSS are as follows:

Service operations

API	Description
GetService	Obtain all the buckets of a specified account.

Bucket operations

API	Description
Put Bucket	Crate a bucket.
Put Bucket ACL	Set the bucket access permission.
Put Bucket Logging	Enable logging for the bucket.
Put Bucket website	Configure static website hosting for the bucket.
Put Bucket Referer	Configure anti-leech rules for the bucket.
Put Bucket Lifecycle	Configure lifecycle rules for objects in the bucket.
Get Bucket ACL	Get the bucket access permission.
Get Bucket location	Get the location information about the data center to which the bucket belongs.
Get Bucket Logging	View the access log configuration of the bucket.
Get Bucket website	View the static website hosting status of the bucket.
Get Bucket Referer	View anti-leech rules for the bucket.
Get Bucket Lifecycle	View the lifecycle rules of objects in the bucket.
Delete Bucket	Delete the bucket.
Delete Bucket Logging	Disable the access logging feature of the bucket.
Delete Bucket website	Disable the static website hosting mode of the bucket.
Delete Bucket Lifecycle	Delete the lifecycle rules of objects in the bucket.
Get Bucket (list object)	Get information of all the objects in the bucket.

API	Description
Get Bucket info	Get bucket Information

Object operations

API	Description
Put Object	Upload an object.
Copy Object	Copy an object to make it another object.
Get Object	Get an object.
Delete Object	Delete an object.
Delete Multiple Objects	Delete multiple objects.
Head Object	Get the object meta information.
Post Object	Upload an object in the Post mode.
Append Object	Append the upload data at the end of the object .
Put Object ACL	Set the object ACL.
Get Object ACL	Get the object ACL information.
Callback	Upload callback.

Multipart upload operations

API	Description
Initiate Multipart upload	Initialize a multipart upload event.
Upload Part	Upload files in multiple parts.
Upload Part Copy	Copy and upload files in multiple parts.
Complete Multipart upload	Complete the multipart upload of the entire file.
Abort Multipart upload	Cancel a multipart upload event.
List Multipart Uploads	List all the ongoing multipart upload events.
List Parts	List all successfully uploaded parts mapped to a specific upload ID.

Cross-Origin Resource Sharing (CORS)

API	Description
Put Bucket cors	Configure a CORS rule for a specified bucket.

API	Description
Get Bucket cors	Get the current CORS rules of a specified bucket.
Delete Bucket cors	Disable the CORS function for a specified bucket and clear all the rules.
Option Object	Preflight request for cross-region access.

3 Access control

3.1 User signature authentication

OSS verifies the identity of the sender of the request by using the AccessKeyID/AccessKeySecret symmetric encryption method. The AccessKeyID identifies the user. With the help of AccessKeyS ecret, you can encrypt the signature string and OSS can verify the AccessKey of the signature string. You must keep your AccessKeySecret confidential and secured. Based on the account types, the AccessKeys can be categorized as follows:

• Alibaba Cloud account AccessKey: The AccessKey provided by each Alibaba Cloud account has full permissions on its resources.

• RAM account AccessKey: A RAM account is generated under the authorization of an Alibaba Cloud account, and the AccessKey of the RAM account has a limited operation permissions on specified resources.

• STS temporary access credential: A temporary credential generated by an Alibaba Cloud account or an RAM account. The AccessKey of the temporary credential has limited operation permissions on specified resources for a specific period of time. The permissions are withdrawn once this time period expires.

For more information, see *Access control*.

Before sending a request to OSS as an individual, you must first generate a signature string for the request to be sent according to the format specified by OSS. Then encrypt the signature string using the AccessKeySecret to generate a verification code. After receiving the request, OSS finds the corresponding AccessKeySecret based on the AccessKeyID, and extracts the signature string and verification code in the same way. If the calculated verification code is the same as the verification code provided, the request is deemed as valid. Otherwise, OSS rejects the request and return an HTTP 403 error.

3.2 Add a signature to the header

You can add an authorization header to carry signature information in an HTTP request to indicate that the message has been authorized.

Calculation of the Authorization field

```
Authorization = "OSS " + AccessKeyId + ":" + Signature
Signature = base64(hmac-sha1(AccessKeySecret,
          VERB + "\n"
          + Content-MD5 + "\n"
```

```
                              + Content-Type + "\n"
                              + Date + "\n"
                              + CanonicalizedOSSHeaders
                              + CanonicalizedResource))
```

- The `AccessKeySecret` indicates the key required for a signature.

- `VERB` indicates the HTTP request method, including PUT, GET, POST, HEAD, and DELETE.

- `\n` is a line break.

- `Content-MD5` The Content-MD5 is the MD5 value of requested content data. The message content (excluding the header) is calculated to obtain an MD5 value, which is a 128-bit number. This number is encoded with Base64 into a Content-MD5 value. The request header can be used to check the message validity, that is, whether the message content is consistent with the sent content, such as "eB5eJF1ptWaXm4bijSPyxw==". The request header may be empty. For more information, see *RFC2616 Content-MD5*.

- `Content-Type` indicates the requested content type, such as "application/octet-stream". It content type may be empty.

- `Date` indicates the operation time. It must be in GMT format, such as "Sun, 22 Nov 2015 08:16:38 GMT".

- The `CanonicalizedOSSHeaders` indicates an assembly of HTTP headers whose prefixes are "x-oss-".

- The `CanonicalizedResource` indicates the OSS resource that the user wants to access.

Specifically, the values of Date and CanonicalizedResource cannot be empty. If the difference between the value of Date in the request and the time of the OSS server is greater than 15 minutes, the OSS server rejects the request and returns an HTTP 403 error.

Construct CanonicalizedOSSHeaders

All the HTTP headers whose prefixes are x-oss- are called CanonicalizedOSSHeaders. The method to construct CanonicalizedResource is as follows:

1. Convert the names of all HTTP request headers whose prefixes are x-oss- into lowercase letters. For example, convert `X-OSS-Meta-Name:TaoBao` to `x-oss-meta-name: TaoBao`.

2. If the request is sent with the AccessKeyID and AccessKeySecret obtained by the STS, you must also add the obtained security-token value to the signature string in the form of `x-oss-security-token:security-token`.

3. Sort all acquired HTTP request headers in a lexicographically ascending order.

4. Delete any space on either side of a separator between the request header and content. For example, convert `x-oss-meta-name: TaoBao` to `x-oss-meta-name:TaoBao`.

5. Separate all the content and headers with the `\n` separator to form the final CanonicalizedOSSHeaders.

> **Note:**
>
> - CanonicalizedOSSHeaders can be empty, and the `\n` at the end can be removed.
> - If only one header must be constructed, it must be `x-oss-meta-a\n`. Note the `\n` at the end.
> - If multiple headers must be constructed, it must be `x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n`. Note the `\n` at the end.

Construct CanonicalizedResource

The target OSS resource specified in the request sent by the user is called a Canonicali zedResource. The method for constructing CanonicalizedResource is as follows:

1. Set CanonicalizedResource into a null character string ("");

2. Add the OSS resource to be accessed in the following format: `/BucketName/ObjectName`. (If ObjectName does not exist, CanonicalizedResource is "/BucketName/". If BucketName does not exist either, CanonicalizedResource is "/".)

3. If the requested resource includes sub-resources (SubResource), sort all the sub-resources in a lexicographically ascending order and separate the sub-resources using the separator `&` to generate a sub-resource string. Add "?" and the sub-resource string to the end of the CanonicalizedResource string. In this case, CanonicalizedResource is like: `/BucketName/ObjectName?acl&uploadId=UploadId`

4. If the user request specifies the query string (QueryString, also called HTTP Request Parameters), sort these query strings and request values in a lexicographically ascending order, separate the query strings and request values using the separator `&`, and add them to CanonicalizedResource based on the parameters. In this case, CanonicalizedResource is like: `/BucketName/ObjectName?acl&response-content-type=ContentType&uploadId=UploadId`.

> **Note:**
>
> - The sub-resources supported by OSS currently include: acl, uploads, location, cors, logging, website, referer, lifecycle, delete, append, tagging, objectMeta, uploadId, partNumber, security -token, position, img, style, styleName, replication, replicationProgress, replicationLocation, cname, bucketInfo, comp, qos, live, status, vod, startTime, endTime, symlink, x-oss-process

, response-content-type, response-content-language, response-expires, response-cache-control, response-content-disposition, and response-content-encoding.

- Three types of sub-resources are available:

  — Resource identifiers, such as acl, append, uploadId, and symlink sub-resources. For more information, see *Bucket-related operations* and *Object-related operations*.

  — Specify response header fields such as `response-***`. For more information, see the `Request Parameters section` of *GetObject* .

  — Object handling methods, such as `x-oss-process`. It is used as the object handling method, such as *Image Processing*.

Rules to calculate a signature header

- A signature string must be in the UTF-8 format. Encode a signature string containing Chinese characters with UTF-8 first, and then use it with the AccessKeySecret to calculate the final signature.

- The signing method adopted is the HMAC-SHA1 method defined in *RFC 2104*, where Key is `AccessKeySecret` .

- Content-Type and Content-MD5 are not required in a request. If the request requires signature verification, the null value can be replaced with the line break "\n".

- Among all non-HTTP-standard headers, only the headers starting with "x-oss-" require signature strings, and other non-HTTP-standard headers are ignored by OSS. (For example, the "x-oss-magic" header in the preceding example must be added with a signature string.)

- Headers starting with "x-oss-" must comply with the following specifications before being used for signature verification:

  — The header name is changed to lower-case letters.

  — The headers are sorted in a lexicographically ascending order.

  — No space exists before and after the colon, which separates the header name and value.

  — Each header is followed by the line break "\n". If no header is used, CanonicalizedOSSHeaders is set to null.

Example signature

Assume that AccessKeyID is 44CF9590006BF252F707 and AccessKeySecret is OtxrzxIsfp FjA7SwPzILwy8Bw21TLhquhboDYROV.

Request	Signature string calculation formula	Signature string
PUT /nelson HTTP/1.0 Content-MD5: eB5eJF1ptW aXm4bijSPyxw== Content-Type: text/html Date: Thu, 17 Nov 2005 18:49:58 GMT Host: oss-example.oss-cn-hangzhou .aliyuncs.com X-OSS-Meta-Author: foo@bar.com X-OSS-Magic: abracadabra	Signature = base64(hmac-sha1(AccessKeySecret,VERB + "\n" + Content-MD5 + "\n"+ Content-Type + "\n" + Date + "\n" + CanonicalizedOSSHead ers+ CanonicalizedResource))	"PUT\n eB5eJF1ptW aXm4bijSPyxw==\n text/ html\n Thu, 17 Nov 2005 18 :49:58 GMT\n x-oss-magic :abracadabra\nx-oss-meta-author:foo@bar.com\n/oss-example/nels

The signature calculation method is as follows:

Python sample code:

```
import base64
import hmac
import sha
h = hmac.new("OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
             "PUT\nODBGOERFMDMzQTczRUY3NUE3NzA5QzdFNUYzMDQxNEM=\ntext
/html\nThu, 17 Nov 2005 18:49:58 GMT\nx-oss-magic:abracadabra\nx-oss-
meta-author:foo@bar.com\n/oss-example/nelson", sha)
Signature = base64.b64encode(h.digest())
print("Signature: %s" % Signature)
```

The signature calculation result is 26NBxoKdsyly4EDv6inkoDft/yA=. According to the formula

 Authorization = "OSS " + AccessKeyID + ":" + Signature, the value of Authorization is OSS

 44CF9590006BF252F707:26NBxoKdsyly4EDv6inkoDft/yA=. The value is added with the

authorization header to form the message to be sent:

```
PUT /nelson HTTP/1.0
Authorization:OSS 44CF9590006BF252F707:26NBxoKdsyly4EDv6inkoDft/yA=
Content-Md5: eB5eJF1ptWaXm4bijSPyxw==
Content-Type: text/html
Date: Thu, 17 Nov 2005 18:49:58 GMT
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
X-OSS-Meta-Author: foo@bar.com
X-OSS-Magic: abracadabra
```

Detail analysis

- If the input AccessKeyID does not exist or is inactive, the error 403 Forbidden is returned. Error code: InvalidAccessKeyId.

- If the authorization value format in the user request header is incorrect, the error 400 Bad Request is returned. Error code: InvalidArgument.

- All the requests of OSS must use the GMT time format stipulated by the HTTP 1.1 protocol. Specifically, the date format is:`date1 = 2DIGIT SP month SP 4DIGIT; day month year(for example, 02 Jun 1982)`. In the aforesaid date format, "day" occupies "2 digits". Therefore, "Jun 2", "2 Jun 1982", and "2-Jun-82" are all invalid date formats.

- If Date is not input into the header or the format is incorrect during signature verification, the error 403 Forbidden is returned. Error code: AccessDenied.

- The request must be entered within 15 minutes based on the current time of the OSS server; otherwise, the error 403 Forbidden is returned. Error code: RequestTimeTooSkewed.

- If the AccessKeyID is active but OSS determines that the signature of the user request is incorrect, the error 403 Forbidden is returned, and the correct signature string for verification and encryption is returned to the user in the response message. The user can check whether or not the signature string is correct based on the response of OSS. Return example:

```
<? xml version="1.0" ? >
<Error>
 <Code>
     SignatureDoesNotMatch
 </Code>
 <Message>
     The request signature we calculated does not match the
signature you provided. Check your key and signing method.
 </Message>
 <StringToSignBytes>
     47 45 54 0a 0a 0a 57 65 64 2c 20 31 31 20 4d 61 79 20 32 30 31
31 20 30 37 3a 35 39 3a 32 35 20 47 4d 54 0a 2f 75 73 72 65 61 6c 74
65 73 74 3f 61 63 6c
 </StringToSignBytes>
 <RequestId>
     1E446260FF9B10C2
 </RequestId>
 <HostId>
     oss-cn-hangzhou.aliyuncs.com
 </HostId>
 <SignatureProvided>
     y5H7yzPsA/tP4+0tH1HHvPEwUv8=
 </SignatureProvided>
 <StringToSign>
     GET
Wed, 11 May 2011 07:59:25 GMT
/oss-example? acl
 </StringToSign>
 <OSSAccessKeyId>
     AKIAIVAKMSMOY7VOMRWQ
 </OSSAccessKeyId>
</Error>
```

📋 **Note:**

- OSS SDK has implemented the signature. You do not need to worry about the signature issue when you use the OSS SDK. To learn more about the signature implementations of specific languages, see the OSS SDK code. The files for implementing OSS SDK signature are shown in the following table:

SDK	Signature implementation
Java SDK	*OSSRequestSigner.java*
Python SDK	*auth.py*
Net SDK	*OssRequestSigner.cs*
PHP SDK	*OssClient.php*
C SDK	*oss_auth.c*
JavaScript SDK	*client.js*
Go SDK	*auth.go*
Ruby SDK	*util.rb*
iOS SDK	*OSSModel.m*
Android SDK	*OSSUtils.java*

- If try to implement the signature by yourself, and the access to OSS receives an error of SignatureDoesNotMatch, you can use *Visualized Signature Tool* to confirm the signature and eliminate the error.

Content-MD5 calculation method

```
Content-MD5 calculation
The message content "123456789" is used as an example. The Content-MD5
 value of the string
is calculated as follows:
The algorithm defined in related standards can be simplified to the
following:
Calculate the MD5-encrypted 128-bit binary array.
Encode the binary array (instead of the 32-bit string code) with
Base64.
Python is used as an example.
The correct calculation code is:
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789")
>>> base64.b64encode(hash.digest())
'eB5eJF1ptWaXm4bijSPyxw=='
Note:
The correct code is: hash.digest(), used to calculate a 128-bit binary
 array
>>> hash.digest()
'x\x1e^$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'
The common error is to base 64 the computed 32-Bit String encoding
directly.
```

```
An incorrect example: hash.hexdigest(), and a visible 32-bit string is
 calculated.
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
Result of encoding the incorrect MD5 value with Base64:
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

3.3 Add a signature to a URL

In addition to using an authorization header, you can add signature information to a URL. It
enables you to forward a URL to the third party for an authorized access.

Implementation

URL signature example:

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?OSSAccessK
eyId=nz2pc56s936**9l&Expires=1141889120&Signature=vjbyPxybdZaNmGa%
2ByT272YEAiv4%3D
```

The URL signature must include at least the following three parameters: **Signature**, **Expires**,
and **OSSAccessKeyId**.

- The `Expires` parameter indicates the time-out period of a URL. The value of this parameter

  is UNIX time (which is the number of seconds that have elapsed since 00:00:00 UTC, January

  1, 1970. For more information, see *Wikipedia*). If the time when OSS receives the URL request

  is later than the value of the Expires parameter and is included in the signature, an error code

  request timed-out is returned. For example, if the current time is 1141889060, to create a URL

  that is scheduled to expire in 60 seconds, you can set the value of Expires to 1141889120.

- `OSSAccessKeyId` refers to the AccessKeyID in the key.

- `Signature` indicates the signature information. For all requests and header parameters that

  OSS supports, the algorithm for adding a signature to a URL is basically the same as that of

  Adding a signature to a header.

```
Signature = urlencode(base64(hmac-sha1(AccessKeySecret,
          VERB + "\n"
          + CONTENT-MD5 + "\n"
          + CONTENT-TYPE + "\n"
          + EXPIRES + "\n"
          + CanonicalizedOSSHeaders
          + CanonicalizedResource)))
```

The difference is listed as follows:

— When a signature is added to a URL, the Expires parameter replaces the Date parameter.

— Signatures cannot be included in a URL and the Header at the same time.

- If more than one incoming Signature, Expires, or AccessKeyId value is available, the first of each incoming value is used.

- Whether the request time is later than the Expires time, is verified first before verifying the signature.

- When you put the signature string into a URL, remember to perform the UrlEncode for a URL.

- When you add a signature to a temporary user URL, the `security-token` must also be entered. The format is as follows:

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?
OSSAccessKeyId=nz2pc56s936**9l&Expires=1141889120&Signature=
vjbyPxybdZaNmGa%2ByT272YEAiv4%3D&security-token=SecurityToken
```

Sample code

Python sample code used to add a signature to a URL:

```
import base64
import hmac
import sha
import urllib
h = hmac.new("OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
             "GET\n\n\n1141889120\n/oss-example/oss-api.pdf",
             sha)
urllib.quote (base64.encodestring(h.digest()).strip())
```

Note:

- The preceding code is the Python sample code.

- OSS SDK provides the method for adding a signature into an URL. For usage, see the " Authorize Access" section in the SDK file.

- To add a signature to the OSS SDK URL, see the following table.

SDK	URL signature method	Implementation file
Java SDK	OSSClient.generatePr esignedUrl	*OSSClient.java*
Python SDK	Bucket.sign_url	*api.py*
Net SDK	OssClient.GeneratePr esignedUri	*OssClient.cs*
PHP SDK	OssClient.signUrl	*OssClient.php*
JavaScript SDK	signatureUrl	*object.js*

SDK	URL signature method	Implementation file
C SDK	oss_gen_signed_url	*oss_object.c*

Detail analysis

- If you adopt the approach of adding a signature to a URL, the authorized data is exposed on the Internet before the authorization period expires. We recommend that you must assess the usage risks in advance.

- The PUT and GET requests both support adding a signature in a URL.

- When a signature is added to a URL, the sequence of Signature, Expires, and AccessKeyId can be swapped. If one or more Signature, Expires, or AccessKeyId parameter is missing, the error 403 Forbidden is returned. Error code: AccessDenied.

- If the current access time is later than the Expires time set in the request, the error 403 Forbidden is returned. Error code: AccessDenied.

- If the format of the Expires time is incorrect, the error 403 Forbidden is returned. Error code: AccessDenied.

- If the URL includes one or more Signature, Expires, or AccessKeyId parameter and the header also includes signature information, the error 400 Bad Request is returned. Error code: InvalidArgument.

- When the signature string is generated, the Date parameter is replaced by the Expires parameter, but the headers such as content-type and content-md5 defined in the preceding section are still included. (Though the Date request header still exists in the request, you can skip adding it to the signature string.)

3.4 Temporary authorized access

Introduction of STS

OSS can temporarily perform authorized access through the Alibaba Cloud STS (Security Token Service). Alibaba Cloud STS is a web service that provides a temporary access token to a cloud computing user. Using STS, you can grant access credentials to a third-party application or federated user (you can manage the user IDs) with customized permissions and validity periods . Third-party applications or federated users can use these access credentials to directly call the Alibaba Cloud product APIs or use the SDKs provided by Alibaba Cloud products to access the cloud product APIs.

- You do not need to expose your long-term key (AccessKey) to a third-party application and only need to generate an access token and send the access token to the third-party application .

- You can customize the access permission and validity of this token.

- You do not need to care about permission revocation issues. The access credential automatically becomes invalid when it expires.

Using an app as an example, the interaction process is shown as follows:
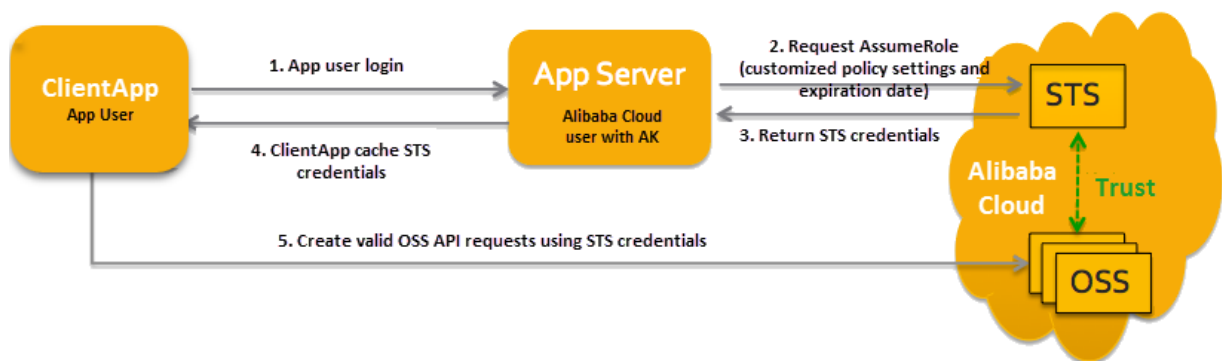
The solution is described in detail as follows:

1. Log on as the app user.

   App user IDs are managed by the customer. Customers 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 calling STS, the AppServer needs to determine the minimum access permission (described in policy syntax) of the app user and the expiration time of the authorization. Then the security token is obtained by calling the STS' AssumeRole interface.

3. The STS returns a valid access credential to the AppServer, including a security token, a temporary AccessKey (AccessKeyID and AccessKeySecret), and the expiry time.

4. The AppServer returns the access credential 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 access credential cached locally to request Alibaba Cloud Service APIs. The cloud services perceives the STS access credential, and relies on STS to verify the credential and correctly respond to the user request.

For more information on the STS Security token, refer to role management in the RAM usage guide.

Call *AssumeRole* of the STS interface to obtain valid access credentials. You can also directly use STS SDK to call the this method.

Use STS credentials to construct signed requests

After obtaining the STS temporary credential, the client of the user creates a signature using the security token (SecurityToken) and temporary AccessKey (AccessKeyId and AccessKeySecret) in the credential. The method for constructing an authorized access signature is basically the same as using the AccessKey of a root account to *add a signature to a header*. Pay attention to the following two points:

- The signature key used by the user is the temporary AccessKey (AccessKeyId and AccessKeySecret) provided by the STS.

- The user needs to carry the security token (security token) in the request header or in the URI as a request parameter. These two manners are alternative. If both manners are selected, OSS returns an InvalidArgument error.

  - The header `x-oss-security-token : SecurityToken` is carried in a request header. When CanonicalizedOSSHeaders of the signature is calculated, x-oss-security-token is taken into consideration.

  - Parameter `security-token=SecurityToken` is carried in the URL. When CanonicalizedResource of the signature is calculated, security-token is taken into consideration and considered as a sub-resource.

3.5 Bucket permission control

OSS provides an Access Control List (ACL) for the bucket-level access control. Currently, three access permissions are available for a bucket: Public Read/Write, Public Read, and Private.

Permission	Name	Access restrictions on visitors
`public-read-write`	Public read and write	• Anyone (including anonymous access) can

Permission	Name	Access restrictions on visitors
		read, write, and delete the objects in the bucket. • The fees incurred by such operations are borne by the owner of the bucket. Therefore, use this permission carefully.
`public-read`	Public read, private write	• Only the owner of the bucket and the authorized users can perform write and delete operations on the objects in the bucket. • Anyone (including anonymous access) can read the objects in the bucket.
`private`	Private read and write	• Only the owner of the bucket and the authorized users can perform read, write, and delete operations on the objects in the bucket. • Other users cannot access objects in the bucket.

📋 **Note:**

- If a new bucket is created without a specified permission, OSS automatically sets a private permission for the bucket.

- For an existing bucket, only the creator of the bucket can change its permissions by using the Put Bucket ACL interface provided by OSS.

4 Definitions of common HTTP headers

Common request headers

Some common request headers are used in the OSS RESTful interfaces. These request headers can be used by all the OSS requests. The following table lists the specific definitions of the request headers:

Name	Type	Description
`Authorization`	string	The verification information used to verify the validity of a request. Default value: none Usage scenario: non-anonymous requests
`Content-Length`	string	Content length of an HTTP request, which is defined in *RFC2616*. Default value: none Usage scenario: requests that need to submit data to OSS
`Content-Type`	string	Content type of an HTTP request, which is defined in *RFC2616*. Default value: none Usage scenario: requests that need to submit data to OSS
`date`	string	The GMT time stipulated in the HTTP 1.1 protocol, for example, Wed, 05 Sep. 2012 23:00:00 GMT Default value: none
`Host`	string	The access host value. Format: `<bucketname>.oss -cn-hangzhou.aliyuncs. com`. Default value: none

Common response headers

Some common response headers are used in the OSS RESTful interfaces. These response
headers can be used by all the OSS requests. The following table lists the specific definitions of
the response headers:

Name	Type	Description
`Content-Length`	string	Content length of an HTTP request, which is defined in *RFC2616*. Default value: none Usage scenario: requests that need to submit data to OSS
`Connection`	enumerative	The connection status between the client and the OSS server. Valid values: `open` or `close` Default value: none
`Date`	string	The GMT time stipulated in the HTTP 1.1 protocol, for example, Wed, 05 Sep. 2012 23:00:00 GMT Default value: none
`Etag`	string	The ETag (entity tag) is created when an object is generated and is used to indicate the content of the object. For an object created for a Put Object request, the value of ETag is the value of MD5 in the content of the object. For an object created in other approaches, the value of ETag is the UUID in the content of the object. The value of ETag can be used to check whether the content of the object is changed. Default value: none
`Server`	string	The server that generates the response. Default value: AliyunOSS

Name	Type	Description
`x-oss-request-id`	string	The UUID of the response. It is created by Alibaba Cloud OSS. In case of any issues when using the OSS service, you can contact OSS support personnel using this field to rapidly locate the issue. Default value: none

5 Service operations

5.1 GetService (ListBuckets)

Sending a Get request to the server can return all buckets owned by the requester, and "/"

represents the root directory.

Request syntax

```
GET / HTTP/1.1
Host: oss.example.com
Date: GMT Date
Authorization: SignatureValue
```

Request parameters

When using GetService(ListBuckets), you can prescribe a limit to the list with a prefix, marker, and
 max-uploads to return partial results.

Table 5-1: Request parameters

Name	Type	Required	Description
prefix	string	No	Indicates that only the buckets whose names match a specified prefix are returned. If this parameter is not specified, prefix information is not used as a filter. Default value: None
marker	string	No	Indicates that the returned results start with the first entry after the marker in an alphabetical order. If this parameter is not specified, all entries are returned from the start. Default value: None
max-keys	string	No	Limits the maximum number of buckets returned for one request. If this parameter is not specified, the default value 100 is used. The value cannot exceed 1000. Default value: 100

Response elements

Name	Type	Description
ListAllMyB ucketsResult	container	Container for saving results of the Get Service request. Subnode: Owner and Buckets Parent node: None
Prefix	string	Prefix of the returned bucket names for one request. This node is available only when not all buckets are returned. Parent node: ListAllMyBucketsResult
Marker	string	Start point of the current GetService(ListBuckets) request. This node is available only when not all buckets are returned. Parent node: ListAllMyBucketsResult
Maxkeys	string	The maximum number of returned results for one request. This node is available only when not all buckets are returned. Parent node: ListAllMyBucketsResult
IsTruncated	Enumerated string	Indicates whether all results have been returned. "true" means that not all results are returned this time; "false" means that all results are returned this time. This node is available only when not all buckets are returned. Valid values: `true` and `false` Parent node: ListAllMyBucketsResult
NextMarker	string	To indicate that this can be counted as a marker for the next GetService(ListBuckets) request to return the unreturned results. This node is available only when not all buckets are returned. Parent node: ListAllMyBucketsResult
Owner	container	Container used for saving the information about the bucket owner. Parent node: ListAllMyBucketsResult
ID	String	User ID of the bucket owner. Parent node: ListAllMyBucketsResult.Owner
DisplayName	string	Name of the bucket owner (the same as the ID currently). Parent node: ListAllMyBucketsResult.Owner

Name	Type	Description
Buckets	container	Container used for saving the information about multiple Buckets. Subnode: Bucket Parent node: ListAllMyBucketsResult
Bucket	container	Container used for saving the bucket information. Subnodes: Name, CreationDate, and Location Parent node: ListAllMyBucketsResult.Buckets
Name	string	Bucket name. Parent node: ListAllMyBucketsResult.Buckets.Bucket
CreateDate	time (format: yyyy-mm-ddThh:mm:ss.timezone, for example, 2011-12-01T12:27:13.000Z)	Bucket creation time. Parent node: ListAllMyBucketsResult.Buckets.Bucket
Location	string	Indicates the data center in which a bucket is located. Parent node: ListAllMyBucketsResult.Buckets.Bucket
ExtranetEndpoint	string	Internet domain name accessed by the bucket. Parent node: ListAllMyBucketsResult.Buckets.Bucket
IntranetEndpoint	string	Intranet domain name accessed by the ECS in the same region. Parent node: ListAllMyBucketsResult.Buckets.Bucket
StorageClass	string	Indicates the bucket storage type. "Standard", "IA", and "Archive" types are available. (The "Archive" type is only available in some regions currently.) Parent node: ListAllMyBucketsResult.Buckets.Bucket

Detail analysis

- The API of GetService is valid only for those users who have been authenticated.

- If no information for user authentication is provided in a request (namely an anonymous access), 403 Forbidden is returned. The error code is "AccessDenied".

- When all buckets are returned, the returned XML does not contain the nodes Prefix, Marker, MaxKeys, IsTruncated, and NextMarker. If some results are not returned yet, the preceding nodes are added, in which NextMarker is used to assign the marker for the successive query.

Example

Request example I

```
GET / HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: oss-cn-hangzhou.aliyuncs.com
Authorization: OSS nxj7dtl1c24jwhcyl5hpvnhi:COS3OQkfQPnKmYZTEHYv
2qUl5jI=
```

Return example I

```
HTTP/1.1 200 OK
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 556
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D74
<? xml version="1.0" encoding="UTF-8"? >
<ListAllMyBucketsResult>
  <Owner>
    <ID>51264</ID>
    <DisplayName>51264</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <CreationDate>2015-12-17T18:12:43.000Z</CreationDate>
      <ExtranetEndpoint>oss-cn-shanghai.aliyuncs.com</ExtranetEndpoint
>
      <IntranetEndpoint>oss-cn-shanghai-internal.aliyuncs.com</
IntranetEndpoint>
      <Location>oss-cn-shanghai</Location>
      <Name>app-base-oss</Name>
      <StorageClass>Standard</StorageClass>
    </Bucket>
    <Bucket>
      <CreationDate>2014-12-25T11:21:04.000Z</CreationDate>
      <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint
>
      <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</
IntranetEndpoint>
      <Location>oss-cn-hangzhou</Location>
      <Name>atestleo23</Name>
      <StorageClass>IA</StorageClass>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

Request example II

```
GET /? prefix=xz02tphky6fjfiuc&max-keys=1 HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
```

```
Host: oss-cn-hangzhou.aliyuncs.com
Authorization: OSS nxj7dtl1c24jwhcyl5hpvnhi:COS3OQkfQPnKmYZTEHYv
2qUl5jI=
```

Return example II

```
HTTP/1.1 200 OK
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 545
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D75
<? xml version="1.0" encoding="UTF-8"? >
<ListAllMyBucketsResult>
  <Prefix>xz02tphky6fjfiuc</Prefix>
  <Marker></Marker>
  <MaxKeys>1</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <NextMarker>xz02tphky6fjfiuc0</NextMarker>
  <Owner>
    <ID>ut_test_put_bucket</ID>
    <DisplayName>ut_test_put_bucket</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <CreationDate>2014-05-15T11:18:32.000Z</CreationDate>
      <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint
>
      <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</
IntranetEndpoint>
      <Location>oss-cn-hangzhou</Location>
      <Name>xz02tphky6fjfiuc0</Name>
      <StorageClass>Standard</StorageClass>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

6 Bucket operations

6.1 PutBucket

The `PutBucket` interface is used to create a bucket (anonymous access is not supported).

The region of the created bucket is consistent with the region of the endpoint from which the request is sent. Once the data center of the bucket is determined, all objects in this bucket are stored in the corresponding region. For more information, see *Regions and endpoints* .

Request syntax

```
PUT / HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
x-oss-acl: Permission
Authorization: SignatureValue
<? xml version="1.0" encoding="UTF-8"? >
<CreateBucketConfiguration>
    <StorageClass>Standard</StorageClass>
</CreateBucketConfiguration>
```

Detail analysis

- You can use the `x-oss-acl` header in a Put request to set access permissions for a bucket. Currently, three access permissions are available for a bucket: public-read-write, public-read, and private.

- If the requested bucket already exists, 409 Conflict is returned. Error code: BucketAlre adyExists.

- If the bucket to be created does not conform to the naming conventions, the message of 400 Bad Request is returned. Error code: InvalidBucketName.

- If the information for user authentication is not introduced when you initiate a Put Bucket request, the message of 403 Forbidden is returned. Error code: AccessDenied.

- You can create a maximum of 30 buckets in a region. If the number is exceeded, the message of 400 Bad Request is returned. Error code: TooManyBuckets.

- If no access permission is specified for the created bucket, the `Private` permission applies by default.

- The storage type of a new bucket can be specified. Standard, IA, and Archive are available.

Example

Request example:

```
PUT / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2017 03:15:40 GMT
x-oss-acl: private
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:77Dvh5wQgIjWjwO/KyRt8dOPfo
8=
<? xml version="1.0" encoding="UTF-8"? >
<CreateBucketConfiguration>
    <StorageClass>Standard</StorageClass>
</CreateBucketConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2017 03:15:40 GMT
Location: /oss-example
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

6.2 Put Bucket ACL

The `PutBucketACL` interface is used to modify the access permissions for a bucket.

Currently, three bucket access permissions are available: public-read-write, public-read, and private. You can use the "x-oss-acl" header in a Put request to set the Put Bucket ACL operation . Only the creator of the bucket has permission to perform this operation. If the operation succeeds , 200 is returned; otherwise, the corresponding error code and prompt message are returned.

Request syntax

```
PUT /? acl HTTP/1.1
x-oss-acl: Permission
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- When a bucket already exists and is owned by the request sender and the permission in the request is different from the existing permission, this request does not change bucket content but updates the permission.

- If the information for user authentication is not introduced when you initiate a Put Bucket request, the message of `403 Forbidden` is returned. Error code: `AccessDenied`.

- If the **`x-oss-acl`** header is unavailable in a request and the bucket already exists and belongs to the request sender, the permissions for the original bucket remain the same.

Example

Request example:

```
PUT /? acl HTTP/1.1
x-oss-acl: public-read
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

If the permission for this setting does not exist, the message of 400 Bad Request is shown:

Returned error example:

```
HTTP/1.1 400 Bad Request
x-oss-request-id: 56594298207FB304438516F9
Date: Fri, 24 Feb 2012 03:55:00 GMT
Content-Length: 309
Content-Type: text/xml; charset=UTF-8
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>InvalidArgument</Code>
  <Message>no such bucket access control exists</Message>
  <RequestId>56594298207FB304438516F9</RequestId>
  <HostId>leo.oss-test.aliyun-inc.com</HostId>
  <ArgumentName>x-oss-acl</ArgumentName>
  <ArgumentValue>error-acl</ArgumentValue>
</Error>
```

6.3 PutBucketLogging

Bucket owners can use **`PutBucketLogging`** to enable the access logging function for their bucket.

When this function is enabled, OSS automatically records the details about the requests to this bucket, and follows the user-specified rules to write the access logs as an object into a user-specified bucket on an hourly basis.

> **Note:**
>
> OSS provides bucket access logs for bucket owners to understand and analyze bucket access behaviors easily. The bucket access logs provided by OSS do not guarantee that every single access record is logged.

Request syntax

```
PUT /? logging HTTP/1.1
Date: GMT Date
Content-Length : ContentLength
Content-Type: application/xml
Authorization: SignatureValue
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus>
    <LoggingEnabled>
        <TargetBucket>TargetBucket</TargetBucket>
        <TargetPrefix>TargetPrefix</TargetPrefix>
    </LoggingEnabled>
</BucketLoggingStatus>
```

Request elements

Name	Type	Required	Description
BucketLogg ingStatus	container	Yes	The container for storing access log status information Child element: LoggingEnabled Parent element: none
LoggingEna bled	container	No	The container for storing access log information. This element is required only when server access logging is enabled. Child element: TargetBucket, TargetPrefix Parent element: BucketLoggingStatus
TargetBuck et	character	This element is required when server access logging is enabled	The bucket for storing access logs. Child element: none Parent element: BucketLoggingStatus. LoggingEnabled
TargetPref ix	character	No	The prefix of the names of saved access log files. Child element: none Parent element: BucketLoggingStatus. LoggingEnabled

Naming rules for the objects storing access logs

```
<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString
```

In the naming rules, the TargetPrefix is specified by the user; YYYY, mm, DD, HH, MM, and SS give the year, month, day, hour, minutes, and seconds of the creation time in Arabic numerals (note the digits); and UniqueString is the string generated by OSS system. An example for the name of an object actually used to store OSS access logs is given as follows:

```
MyLog-oss-example-2012-09-10-04-00-00-0000
```

In the preceding example, "MyLog-" is the Object prefix specified by the user; "oss-example" is the name of the origin bucket; "2012-09-10-04-00-00" is the Object creation time (Beijing time); and " 0000" is the string generated by OSS system.

Log file format

Name	Example	Description
Remote IP	119.140.142.11	IP address from which the request is initiated (the proxy or user firewall may block this field)
Reserved	-	Reserved field
Reserved	-	Reserved field
Time	[02/May/2012:00:00:04 +0800]	Time when OSS receives the request
Request-URI	"GET /aliyun-logo.png HTTP/1. 1"	User-Requested URI (including query-string)
HTTP Status	200	HTTP status code returned by OSS
SentBytes	5576	Traffic that the user downloads from OSS
RequestTime (ms)	71	Time utilized in completing this request (in ms)
Referer	`http://www.aliyun.com /product/oss`	HTTP Referer in the request
User-Agent	curl/7.15.5	HTTP User-Agent header
HostName	oss-example.regionid.example .com	Domain name for access request
Request ID	505B01695037C2AF032593A4	UUID used to uniquely identify this request
LoggingFlag	true	Whether the access logging function is enabled

Name	Example	Description
Requester Aliyun ID	1657136103983691	Alibaba Cloud ID of the requester, "-" for an anonymous access
Operation	GetObject	Request type
Bucket	oss-example	Name of the bucket requested for access
Key	/aliyun-logo.png	Key of user request
ObjectSize	5576	Object size
Server Cost Time (ms)	17	Time utilized by OSS server to process this request (in ms)
Error Code	NoSuchBucket	Error code returned by OSS
Request Length	302	Length of user request (byte)
UserID	1657136103983691	ID of the bucket owner
Delta DataSize	280	Bucket size variation, "-" for no change
Sync Request	-	Whether this is an origin retrieval request from CND, "-" for no
Reserved	-	Reserved field

Detail analysis

- The source bucket and target bucket must belong to the same user.

- In the preceding request syntax, "BucketName" refers to the bucket for which access logging is enabled; "TargetBucket" refers to the bucket into which access logs are saved; "TargetPrefix" refers to the name prefix of the object storing access logs and can be null.

- The source bucket and target bucket can be the same or different buckets. You can save logs from multiple source buckets to the same target bucket (in this case, we recommend that you assign different values to TargetPrefix).

- To disable the access logging function for a bucket, you only must send an empty BucketLogg ingStatus request. For a detailed method, see the following request example.

- All PUT Bucket Logging requests must be provided with signatures, because the anonymous access is not supported.

- If the initiator of a PUT Bucket Logging request is not the owner of the source bucket (BucketName in the request example), OSS returns error code 403.

- If the source bucket does not exist, OSS returns the error code: NoSuchBucket.

- If the initiator of a PUT Bucket Logging request is not the owner of the target bucket (indicated by TargetBucket in the request example), OSS returns Error 403. If the target bucket does not exist, OSS returns the error code: InvalidTargetBucketForLogging.

- The source bucket and target bucket must belong to the same data center. Otherwise, Error 400 with the error code: InvalidTargetBucketForLogging is returned.

- If a PUT Bucket Logging request has an invalid XML, the error code: MalformedXML is returned.

- The source bucket and target bucket can be the same bucket. You can save the logs of different source buckets into the same target bucket (note that you must set TargetPrefix to different values).

- When the source bucket is deleted, the corresponding logging rules are also deleted.

- OSS generates a bucket access log file every hour. However, all requests during the hour may not be recorded in the log file, but may get recorded in the previous or next log file.

- In the naming rules for log files generated by OSS, "UniqueString" is only a UUID that OSS generates for a file to uniquely identify the file.

- Each time OSS generates a bucket access log file, this is considered a PUT operation and the occupied space is recorded, but the generated traffic is not recorded. After log files are generated, you can operate these log files as common objects.

- OSS ignores all query-string parameters prefixed by "x-" but such query-string parameters are recorded in access logs. If you want to mark a special request from massive access logs, you can add a query-string parameter prefixed by "x-" to the URL. For example:

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/aliyun-logo.png
```

```
http://oss-example.regionid.example.com/aliyun-logo.png?x-user=admin
```

- When OSS processes the preceding two requests, the results are the same. However, you can search access logs with "x-user=admin" to quickly locate the marked request.

- You may see "-" in any field of OSS logs. It indicates that data is unknown or the field is invalid for the current request.

- Certain fields are added to the end of OSS log files in future based on the requirements. We recommend that developers to consider compatibility issues when developing log processing tools.

- If you have uploaded the Content-MD5 request header, OSS calculates the body's Content-MD5 and checks if the two are the same. If the two are different, the error code: InvalidDigest is returned.

Example

Example of a request for enabling bucket access logging:

```
PUT /? logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus>
<LoggingEnabled>
<TargetBucket>doc-log</TargetBucket>
<TargetPrefix>MyLog-</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Example of a request for disabling bucket access logging:

```
PUT /? logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Type: application/xml
Content-Length: 86
Date: Fri, 04 May 2012 04:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus>
</BucketLoggingStatus>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 04:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

6.4 Putbucketwebsite

The `PutBucketWebsite` interface is used to set a bucket to the static website hosting mode.

Request syntax

```
PUT /? website HTTP/1.1
```

```
Date: GMT Date
Content-Length : ContentLength
Content-Type: application/xml
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue

<? xml version="1.0" encoding="UTF-8"? >
<WebsiteConfiguration>
    <IndexDocument>
        <Suffix>index.html</Suffix>
    </IndexDocument>
    <ErrorDocument>
        <Key>errorDocument.html</Key>
    </ErrorDocument>
</WebsiteConfiguration>
```

Request elements

Name	Type	Description	Required
ErrorDocument	container	The parent element of the child element key Parent element: WebsiteConfiguration	No
IndexDocument	container	The parent element of the child element, suffix Parent element: WebsiteConfiguration	Yes
Key	string	The file name used to return Error 404 Parent element: WebsiteConfiguration. ErrorDocument Conditional: This element is required only when ErrorDocument is set.	Conditional
Suffix	string	The index file name added when a directory URL is returned. This element cannot be empty or contain a slash (/). For example, if the index file index.html is configured, oss-cn-hangzhou.aliyuncs.com/	Yes.

Name	Type	Description	Required
		mybucket/mydir/ contained in an access request is converted into oss-cn-hangzhou.aliyuncs.com/ mybucket/index.html by default. Parent element: WebsiteConfiguration. IndexDocument	
WebsiteConfiguration	container	Requested container Parent element: none	Yes

Detail analysis

- Static websites are the websites where all web pages are composed of static content, including scripts such as JavaScript executed on the client. OSS does not support content that needs to be processed by the server, such as PHP, JSP, and APS.NET.

- If you want to use your own domain name to access bucket-based static websites, the CNAME domain name applies. For more information about the configuration method, see *Bind custom domain names (CNAME)*

- When you set a bucket to the static website hosting mode, you must specify the index page wherein the error page is optional.

- When you set a bucket to the static website hosting mode, the specified index page and error page are an object in the bucket.

- After a bucket is set to a static website hosting mode, OSS returns the index page for anonymous access to the root domain name of the static website, and return Get Bucket results for a signed access to the root domain name of the static website.

- If you have uploaded the Content-MD5 request header, OSS calculates the body's Content-MD5 and checks if the two are the same. If the two are different, the error code: InvalidDigest is returned.

Examples

Request example:

```
PUT /? website HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 209
```

```
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=

<? xml version="1.0" encoding="UTF-8"? >
<WebsiteConfiguration>
<IndexDocument>
<Suffix> indexhtml </suffix>
</IndexDocument>
<ErrorDocument>
<Key>error.html</Key>
</ErrorDocument>
</WebsiteConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

6.5 PutBucketReferer

With the `PutBucketReferer` operation, you can set the referer access whitelist of a bucket and
whether the access request with the referer field being null is allowed.

Request syntax

```
PUT /? referer HTTP/1.1
Date: GMT Date
Content-Length :ContentLength
Content-Type: application/xml
Host: BucketName.oss.aliyuncs.com
Authorization: SignatureValue

<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
    <RefererList>
        <Referer> http://www.aliyun.com</Referer>
        <Referer> https://www.aliyun.com</Referer>
        <Referer> http://www. *.com</Referer>
        <Referer> https://www.?.aliyuncs.com</Referer>
    </RefererList>
</RefererConfiguration>
```

Request elements

Name	Type	Required	Description
`RefererCon figuration`	container	Yes	The container that saves the Referer configuration content Sub-nodes: AllowEmptyReferer node and RefererList node

Name	Type	Required	Description
			Parent node: none
AllowEmpty Referer	enumerativ e string	Yes	Specify whether the access request with the referer field being null is allowed. Valid value: `true` or `false` Default value:`true` Parent node: RefererConfiguration
RefererLis t	container	Yes	The container that saves the referer access whitelist. Parent node: RefererConfiguration Sub-node: Referer
Referer	string	No	Specify a referer access whitelist. Parent node: RefererList

Detail analysis

- Only the bucket owner can initiate a Put Bucket Referer request. Otherwise, the message of 403 Forbidden is returned. Error code: AccessDenied.

- The configuration specified in AllowEmptyReferer replaces the previous AllowEmptyReferer configuration. This field is required. By default, AllowEmptyReferer in the system is configured as true.

- This operation overwrites the previously configured whitelist with the whitelist in the RefererLis t. When the user-uploaded RefererList is empty (containing no referer request element), this operation overwrites the configured whitelist, that is, the previously configured RefererList is deleted.

- If you have uploaded the Content-MD5 request header, OSS calculates the body's Content- MD5 and checks if the two are the same. If the two are different, the error code: InvalidDigest is returned.

Example

Example of a request with no referer contained:

```
PUT /? referer HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=

<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList />
```

```
</RefererConfiguration>
```

Example of a request with referer contained:

```
PUT /? referer HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=

<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList>
<Referer> http://www.aliyun.com</Referer>
<Referer> https://www.aliyun.com</Referer>
<Referer> http://www. *.com</Referer>
<Referer> https://www.?.aliyuncs.com</Referer>
</ RefererList>
</RefererConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

6.6 PutBucketLifecycle

The bucket owner can set the lifecycle of a bucket with the `PutBucketLifecycle`

request. After Lifecycle is enabled, OSS automatically deletes the objects or transitions the objects

(to another storage class) corresponding the lifecycle rules on a regular basis.

Request syntax

```
PUT /?lifecycle HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Authorization: SignatureValue
Host: BucketName.oss.aliyuncs.com
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>RuleID</ID>
    <Prefix>Prefix</Prefix>
    <Status>Status</Status>
    <Expiration>
      <Days>Days</Days>
    </Expiration>
    <AbortMultipartUpload>
      <Days>Days</Days>
    </AbortMultipartUpload>
```

```
   </Rule>
</LifecycleConfiguration>
```

Request elements

Name	Type	Required?	Description
CreatedBef oreDate	string	One from the two: Days and CreatedBef oreDate	Specify the time before which the rules go into effect. The date must conform to the ISO8601 format and always be UTC 00:00. For example: 2002-10-11T00:00:00.000Z, which means the objects with a last update time before 2002-10-11T00:00:00.000Z are deleted or transitioned to another storage class, and the objects updated after this time (including this time) are not deleted or transitioned. Parent node: Expiration or AbortMulti partUpload
Days	positive integer	One from the two: Days and CreatedBef oreDate	Specify how many days after the last object update until the rules take effect. Parent node: Expiration
Expiration	container	No	Specify the expiration attribute of the object. Sub-node: Days or CreatedBeforeDate Parent node: Rule
AbortMulti partUpload	container	No	Specify the expiration attribute of the unfulfilled Part rules. Sub-node: Days or CreatedBeforeDate Parent node: Rule
ID	string	No	The unique ID of a rule. An ID is composed of 255 bytes at most. When you fail to specify this value or this value is null, OSS generates a unique value for you. Sub-node: none Parent node: Rule
LifecycleC onfiguration	container	Yes	Container used for storing lifecycle configurations, which can hold a maximum of 1000 rules. Sub-node: Rule Parent node: none

Name	Type	Required?	Description
Prefix	string	Yes	Specify the prefix applicable to a rule. Only those objects with a matching prefix can be affected by the rule. It cannot be overlapped. Sub-node: none Parent node: Rule
Rule	container	Yes	Express a rule Sub-nodes: ID, Prefix, Status, Expiration Parent node: LifecycleConfiguration
Status	string	Yes	If this value is Enabled, OSS runs this rule regularly. If this value is Disabled, then OSS ignores this rule. Parent node: Rule Valid value: `Enabled`, `Disabled`
StorageClass	string	Required if parent node transition is set	Specifies the type of target storage that the object is transition to the OSS. Value: `IA`, `Archive` Parent node: Transition
Transition	Container	No	Specifies when the object is transition to the IA or archive storage type during a valid life cycle .

Detail analysis

- Only the bucket owner can initiate a Put Bucket Lifecycle request. Otherwise, the message of 403 Forbidden is returned. Error code: AccessDenied.

- If no lifecycle has been set previously, this operation creates a new lifecycle configuration or overwrites the previous configuration.

- You can also set an expiration time for an object, or for the Part. Here, the Part refers to the unsubmitted parts for multipart upload.

Notes for storage types transition:

- Supports objects in Standard bucket transition to IA and Archive storage type. Standard bucket can simultaneously configure both transition to IA and archive storage type rules for one object . In this case, the time set to transition to archive must be longer than the time to transition to IA . For example, the days set for transition to IA is 30, then it must be greater than 30 days set for transition to archive. Otherwise, the invalidargument error is returned.

- The object setting must have an expiration time greater than the time converted to IA or archive
  . Otherwise, the invalidArgument error is returned.

- Supports objects transition to archive storage type in IA bucket.

- Archvie bucket creation is not supported.

- IA object convertion is not supported as standard.

- The archive object convertion is not supported for IA or standard.

Examples

Request example:

```
PUT /?lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 443
Date: Mon, 14 Apr 2014 01:08:38 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
<?xml version="1.0" encoding="UTF-8"?>
</LifecycleConfiguration>
  <Rule>
    <ID>delete objects and parts after one day</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>1</Days>
    </Expiration>
    <AbortMultipartUpload>
      <Days>1</Days>
    </AbortMultipartUpload>
  </Rule>
  <Rule>
    <ID>delete created before date</ID>
    <Prefix>backup/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
    </Expiration>
    <AbortMultipartUpload>
      <CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
    </AbortMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu , 8 Jun 2017 13:08:38 GMT
Content-Length: 0
Connection: keep-alive
```

```
Server: AliyunOSS
```

6.7 GetBucket(List Object)

The `GetBucket` operation can be used to list all of the object information in a bucket.

Request syntax

```
GET / HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request parameters

When you initiate a GetBucket (ListObject) request, you can use prefix, marker, delimiter, and max
-keys to prescribe a limit to the list to return partial results. Besides, encoding-type can be used to
 encode the following elements in the returned results: delimiter, marker, prefix, NextMarker, and
key.

Name	Data type	Required	Description
`delimiter`	string	No	A character used to group object names. All the names of the objects that contain a specified prefix and after which the delimiter occurs for the first time, act as a group of elements - CommonPrefixes. Default value: None
`marker`	string	No	Sets the returned results to begin from the first entry after the marker in alphabetical order. Default value: None
`max-keys`	string	No	Limits the maximum number of objects returned for one request. If not specified, the default value is 100. The max-keys value cannot exceed 1000. Default value: 100
`prefix`	string	No	Limits that the returned object key must be prefixed accordingly. Note that the keys returned from queries using a prefix still contain the prefix. Default value: None
`encoding-type`	string	No	Specifies the encoding of the returned content and the encoding type. Parameters delimiter, marker, prefix, NextMarker, and key use UTF-8 characters, but the XML 1.0 Standard does not support parsing certain control characters, such as characters with ASCII values ranging from 0 to 10. If some elements in the returned

Name	Data type	Required	Description
			results contain characters that are not supported by the XML 1.0 Standard, encoding-type can be specified to encode these elements, such as delimiter, marker, prefix, NextMarker, and key. Default value: None; Optional value: URL

Response elements

Name	Type	Description
Contents	container	Container used for saving every returned object meta. Parent node: ListBucketResult
CommonPrefixes	string	If the delimiter parameter is specified in the request, the response returned by OSS contains the CommonPrefixes element. This element indicates the set of objects which ends with a delimiter and have a common prefix. Parent node: ListBucketResult
Delimiter	string	A character used to group object names. 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 - CommonPrefixes. Parent node: ListBucketResult
EncodingType	string	Encoding type for the returned results. If encoding-type is specified in a request, the following elements in the returned results are encoded: delimiter, marker, prefix, NextMarker, and key. Parent node: ListBucketResult
DisplayName	string	Name of the object owner. Parent node: ListBucketResult.Contents.Owner
ETag	string	The ETag (entity tag) is created when an object is generated and is used to indicate the content of the object. For an object created by a Put Object request, the value of ETag is the value of MD5 in the content of the object. For an object created in other way, the value of ETag is the UUID in the content of the object. The value of ETag can be used to check whether the content of the object is changed. We recommend that the ETag be used as the MD5 value of the object content to verify data integrity. Parent node: ListBucketResult.Contents

Name	Type	Description
ID	string	User ID of the bucket owner. Parent node: ListBucketResult.Contents.Owner
IsTruncated	enumerated string	Indicates whether all results have been returned; "true" means that not all results are returned this time; "false" means that all results are returned this time. Valid values: `true` and `false` Parent node: ListBucketResult
Key	string	Key of an object Parent node: ListBucketResult.Contents
LastModified	time	The latest modification time of an object. Parent node: ListBucketResult.Contents
ListBucket Result	container	Container for storing the results of the "Get Bucket" request subnodes: Name, Prefix, Marker, MaxKeys, Delimiter, IsTruncated, Nextmarker, and Contents Parent node: None
Marker	string	Marks the origin of the current Get Bucket (List Object) request. Parent node: ListBucketResult
MaxKeys	string	The maximum number of returned results in response to the request. Parent node: ListBucketResult
Name	string	Name of a bucket Parent node: ListBucketResult
Owner	container	Container used for saving the information about the bucket owner. subnodes: DisplayName and ID Parent node: ListBucketResult
Prefix	string	Starting prefix for the current results of query. Parent node: ListBucketResult
Size	string	Number of bytes of the object. Parent node: ListBucketResult.Contents
StorageClass	string	Indicates Object storage type. "Standard", "IA", and "Archive" types are available. (Currently, the "Archive" type is only available in some regions.) Parent node: ListBucketResult.Contents

Detail analysis

- The custom meta in the object is not returned during the GetBucket request.

- If the bucket to be accessed does not exist, or if you attempt to access a bucket which cannot be created because of standard naming rules are not followed when naming a bucket, Error 404 Not Found with the error code "NoSuchBucket" is returned.

- If you have no permission to access the bucket, the system returns Error 403 Forbidden with the error code "AccessDenied".

- If listing cannot be completed at one time because of the max-keys setting, a `<NextMarker>` is appended to the returned result, prompting that this can be taken as a marker for continued listing. The value in NextMarker is still in the list result.

- During a condition query, even if the marker does not exist in the list actually, what is returned is printed starting from the next to what conforms to the marker letter sorting. If the max-keys value is less than 0 or greater than 1000, error 400 Bad Request is returned. The error code is "InvalidArgument".

- If the prefix, marker, or delimiter parameters do not meet the length requirement, 400 Bad Request is returned. The error code is "InvalidArgument".

- The prefix and marker parameters are used to achieve display by pages, and the parameter length must be less than 1024 bytes.

- Setting a prefix as the name of a folder lists the files starting with this prefix, recursively returning all files and subfolders in this folder. Additionally, if we set the Delimiter as "/", the returned values lists the files in the folder and the subfolders are returned in the CommonPrefixes section. Recursive files and folders in the subfolders are not displayed. For example, a bucket has the following three objects: fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi. If the prefix is set to "fun/", three objects are returned. If the delimiter is set to "/" additionally, file "fun/test.jpg" and prefix "fun/movie/" are returned. That is, the folder logic is achieved.

Scenario example

Four objects are available in the bucket "my_oss" and are named as:

- oss.jpg

- fun/test.jpg

- fun/movie/001.avi

- fun/movie/007.avi

Example

Request example:

```
GET / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:BC+oQIXVR2/ZghT7cGa0y
kboO4M=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1866
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
<Name>oss-example</Name>
<Prefix></Prefix>
<Marker></Marker>
<MaxKeys>100</MaxKeys>
<Delimiter></Delimiter>
    <IsTruncated>false</IsTruncated>
    <Contents>
        <Key>fun/movie/001.avi</Key>
        <LastModified>2012-02-24T08:43:07.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user-example</DisplayName>
        </Owner>
    </Contents>
    <Contents>
        <Key>fun/movie/007.avi</Key>
        <LastModified>2012-02-24T08:43:27.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user-example</DisplayName>
        </Owner>
    </Contents>
<Contents>
        <Key>fun/test.jpg</Key>
        <LastModified>2012-02-24T08:42:32.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
```

```
            <DisplayName>user-example</DisplayName>
        </Owner>
    </Contents>
    <Contents>
        <Key>oss.jpg</Key>
        <LastModified>2012-02-24T06:07:48.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user-example</DisplayName>
        </Owner>
    </Contents>
 </ListBucketResult>
```

Example of a request containing the prefix parameter:

```
GET /? prefix=fun HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:BC+oQIXVR2/ZghT7cGa0y
kboO4M=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1464
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
<Name>oss-example</Name>
<Prefix>fun</Prefix>
<Marker></Marker>
<MaxKeys>100</MaxKeys>
<Delimiter></Delimiter>
    <IsTruncated>false</IsTruncated>
    <Contents>
        <Key>fun/movie/001.avi</Key>
        <LastModified>2012-02-24T08:43:07.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user_example</DisplayName>
        </Owner>
    </Contents>
    <Contents>
        <Key>fun/movie/007.avi</Key>
        <LastModified>2012-02-24T08:43:27.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
```

```
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user_example</DisplayName>
        </Owner>
    </Contents>
    <Contents>
        <Key>fun/test.jpg</Key>
        <LastModified>2012-02-24T08:42:32.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user_example</DisplayName>
        </Owner>
    </Contents>
</ListBucketResult>
```

Example of a request containing parameters prefix and delimiter:

```
GET /? prefix=fun/&delimiter=/ HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:DNrnx7xHk3sgysx7I8U9
I9IY1vY=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 712
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<ListBucketResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
<Name>oss-example</Name>
<Prefix>fun/</Prefix>
<Marker></Marker>
<MaxKeys>100</MaxKeys>
<Delimiter>/</Delimiter>
    <IsTruncated>false</IsTruncated>
    <Contents>
        <Key>fun/test.jpg</Key>
        <LastModified>2012-02-24T08:42:32.000Z</LastModified>
        <ETag>&quot;5B3C1A2E053D763E1B002CC607C5A0FE&quot;</ETag>
        <Type>Normal</Type>
        <Size>344606</Size>
        <StorageClass>Standard</StorageClass>
        <Owner>
            <ID>00220120222</ID>
            <DisplayName>user_example</DisplayName>
        </Owner>
    </Contents>
    <CommonPrefixes>
        <Prefix>fun/movie/</Prefix>
    </CommonPrefixes>
```

```
</ListBucketResult>
```

6.8 GetBucketAcl

`GetBucketAcl` is used to obtain the access permissions for a bucket.

Request syntax

```
GET /? acl HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
Accesscont rollist	container	Container used for storing the ACL information Parent node: AccessControlPolicy
AccessCont rolPolicy	container	Specify the container that stores the Get Bucket ACL result Parent node: none
Displayname	string	Name of the bucket owner. (Currently it is consistent with the ID) Parent node: AccessControlPolicy.Owner
Grant	enumerative string	ACL permissions of the bucket The acl permission for the bucket. Valid values: `private`, `public-read`, and `public-read-write` Parent node: AccessControlPolicy.AccessControlList
ID	string	User ID of the bucket owner Parent node: AccessControlPolicy.Owner
Owner	container	Container used for saving the information about the bucket owner Parent node: AccessControlPolicy

Detail analysis

Only the bucket owner can use the `GetBucketAcl` interface.

Example

Request example:

```
GET /? acl HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 04:11:23 GMT
```

```
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0Fmg
brQ0=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 04:11:23 GMT
Content-Length: 253
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS

<? xml version="1.0" ? >
<AccessControlPolicy>
    <Owner>
        <ID>00220120222</ID>
        <DisplayName>user_example</DisplayName>
    </Owner>
    <AccessControlList>
        <Grant>public-read</Grant>
    </AccessControlList>
</AccessControlPolicy>
```

6.9 Getbucketlocation

`GetBucketLocation` is used to view the location information about the data center to which a

bucket belongs.

Request syntax

```
GET /? Location HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
`Locationconstraint`	String	Region where a bucket is located. Values: `oss-cn-hangzhou, oss-cn-qingdao, oss-cn-beijing, oss-cn-hongkong, oss-cn-shenzhen`and `oss-cn-shanghai`

Detail analysis

• Only the owner of a bucket can view the location information of the bucket. If other users

attempt to access the location information, the error 403 Forbidden with the error code:

AccessDenied is returned.

- LocationConstraint has the following valid values: oss-cn-hangzhou, oss-cn-qingdao, oss-cn
  -beijing, oss-cn-hongkong, oss-cn-shenzhen, oss-cn-shanghai, oss-us-west-1, oss-us-east-1
  , and oss-ap-southeast-1, which separately indicate the Hangzhou data center, Qingdao data
   center, Beijing data center, Hong Kong data center, Shenzhen data center, Shanghai data
  center, US Silicon Valley data center, US, Virginia data center, and Asia-Pacific (Singapore)
  data center.

Examples

Request example:

```
Get /? location HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSp
YbJ3naA=
```

Response example with logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 15 Mar 2013 05:31:04 GMT
Connection: keep-alive
Content-Length: 90
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<LocationConstraint xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
oss-cn-hangzhou</LocationConstraint >
```

6.10 GetBucketInfo

`GetBucketInfo` operation is used to view the bucket information.

The information includes the following:

- Create time

- Internet access endpoint

- Intranet access endpoint

- Bucket owner information

- Bucket ACL (AccessControlList)

Request syntax

```
GET /? bucketInfo HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: GMT Date
```

```
Authorization: SignatureValue
```

Response elements

Name	Type	Description
BucketInfo	Container	The container that saves the bucket information content Sub-node: Bucket node Parent node: none
Bucket	Container	The container that saves the bucket specific information Parent node: BucketInfo node
CreationDate	time	The creation time of the bucket. Time format: 2013-07-31T10:56:21.000Z Parent node: BucketInfo.Bucket
ExtranetEn dpoint	string	The Internet domain name that the bucket accesses Parent node: BucketInfo.Bucket
IntranetEn dpoint	string	The intranet domain name for accessing the bucket from ECS in the same region Parent node: BucketInfo.Bucket
Location	string	The region of the data center that the bucket is located in Parent node: BucketInfo.Bucket
Name	string	The bucket name Parent node: BucketInfo.Bucket
Owner	container	Container used for saving the information about the bucket owner. Parent node: BucketInfo.Bucket
ID	string	User ID of the bucket owner. Parent node: BucketInfo.Bucket.Owner
DisplayName	string	Name of the bucket owner (the same as the ID currently). Parent node: BucketInfo.Bucket.Owner
AccessCont rolList	container	Container used for storing the ACL information Parent node: BucketInfo.Bucket
Grant	enumerative string	ACL permissions of the bucket. Valid values: `private`, `public-read`, and `public-read-write` Parent node: BucketInfo.Bucket.AccessControlList

Detail analysis

- If the bucket does not exist, error 404 is returned. Error code: NoSuchBucket.

- Only the owner of a bucket can view the information of the bucket. If other users attempt to access the location information, the error 403 Forbidden with the error code: AccessDenied is returned.

- The request can be initiated from any OSS endpoint.

Example

Request example:

```
Get /? bucketInfo HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Sat, 12 Sep 2015 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51
NNHD39zXw=
```

Return example after the bucket information is obtained successfully:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Sat, 12 Sep 2015 07:51:28 GMT
Connection: keep-alive
Content-Length: 531
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<BucketInfo>
  <Bucket>
    <CreationDate>2013-07-31T10:56:21.000Z</CreationDate>
    <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
    <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</
IntranetEndpoint>
    <Location>oss-cn-hangzhou</Location>
    <Name>oss-example</Name>
    <Owner>
      <DisplayName>username</DisplayName>
      <ID>271834739143143</ID>
    </Owner>
    <AccessControlList>
      <Grant>private</Grant>
    </AccessControlList>
  </Bucket>
</BucketInfo>
```

Return example if the requested bucket information does not exist:

```
HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906009B
Date: Sat, 12 Sep 2015 07:51:28 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS
```

```
<? xml version="1.0" encoding="UTF-8"? >
<Error>
   <Code>NoSuchBucket</Code>
   <Message>The specified bucket does not exist.</Message>
   <RequestId>568D547F31243C673BA14274</RequestId>
   <HostId>nosuchbucket.oss.aliyuncs.com</HostId>
   <BucketName>nosuchbucket</BucketName>
</Error>
```

Return example if the requester has no access permission to the bucket information:

```
HTTP/1.1 403
x-oss-request-id: 534B371674E88A4D8906008C
Date: Sat, 12 Sep 2015 07:51:28 GMT
Connection: keep-alive
Content-Length: 209
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<Error>
   <Code>AccessDenied</Code>
   <Message>AccessDenied</Message>
   <RequestId>568D5566F2D0F89F5C0EB66E</RequestId>
   <Hostid> test.oss.aliyuncs.com </hostid>
</Error>
```

6.11 GetBucketLogging

`GetBucketLogging` is used to view the access log configurations of a bucket.

Request syntax

```
GET /? logging HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
`BucketLogg ingStatus`	Container	The container for storing access log status information Child element: LoggingEnabled Parent element: none
`LoggingEnabled`	Container	The container for storing access log information. This element is required only when server access logging is enabled. Child element: TargetBucket, TargetPrefix Parent element: BucketLoggingStatus
`TargetBucket`	Character	The bucket for storing access logs. Child element: none

Name	Type	Description
		Parent element: BucketLoggingStatus. LoggingEnabled
`TargetPrefix`	Character	The prefix of the names of saved access log files. Child element: none Parent element: BucketLoggingStatus. LoggingEnabled

Detail analysis

- If a bucket does not exist, the error "404 no content" is returned. Error code: NoSuchBucket.

- Only the owner of a bucket can view the access logging configuration of the bucket. If other users attempt to access the configuration, the error 403 Forbidden with the error code: AccessDenied is returned.

- If no logging rules are set for the source bucket, OSS still returns an XML message body with the element BucketLoggingStatus being null.

Example

Request example:

```
Get /? logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSp
YbJ3naA=
```

Response example with logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 210
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
    <LoggingEnabled>
        <TargetBucket>mybucketlogs</TargetBucket>
        <TargetPrefix>mybucket-access_log/</TargetPrefix>
    </LoggingEnabled>
</BucketLoggingStatus>
```

Response example with no logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
```

```
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 110
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<BucketLoggingStatus xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
</BucketLoggingStatus>
```

6.12 GetBucketWebsite

`GetBucketWebsite` operation is used to view the static website hosting status of a bucket.

Request syntax

```
GET /? website HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
ErrorDocument	Container	The parent element of the child element key Parent element: WebsiteConfiguration
IndexDocument	Container	The parent element of the child element suffix Parent element: WebsiteConfiguration
Key	String	The file name used to return Error 404 Parent element: WebsiteConfiguration.ErrorDocumen This element is required when ErrorDocument is set
Suffix	String	The index file name added when a directory URL is returned. This element cannot be empty or contain a slash (/). For example, if the index file index.html is configured, oss-cn-hangzhou.aliyuncs.com/mybucket/ mydir/ contained in an access request is converted into oss-cn-hangzhou.aliyuncs.com/mybucket/index.html by default. Parent element: WebsiteConfiguration.IndexDocument
WebsiteCon figuration	Container	Requested container Parent element: none

Detail analysis

• If a bucket does not exist, the error "404 no content" is returned. Error code: NoSuchBucket.

- Only the owner of a bucket can view the static website hosting status of the bucket. If other users attempt to access the status information, the error 403 Forbidden with the error code: AccessDenied is returned.

- If the source bucket is not configured with static website hosting, OSS returns Error 404 with the error code: NoSuchWebsiteConfiguration.

Example

Request example:

```
    Get /? website HTTP/1.1
    Host: oss-example.oss-cn-hangzhou.aliyuncs.com
    Date: Thu, 13 Sep 2012 07:51:28 GMT
    Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51
NNHD39zXw=
```

Response example with logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<Websiteconfiguration xmlns = "http://doc.oss-cn-hangzhou.aliyuncs.com
">
<IndexDocument>
<Suffix>index.html</Suffix>
    </IndexDocument>
    <ErrorDocument>
        <Key>error.html</Key>
    </ErrorDocument>
</WebsiteConfiguration>
```

Return example with logging rules not set:

```
HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<Error xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
    <Code>NoSuchWebsiteConfiguration</Code>
    <Message>The specified bucket does not have a website configurat
ion.</Message>
    <Bucketname> OSS-example </bucketname>
    <RequestId>505191BEC4689A033D00236F</RequestId>
    <HostId>oss-example.oss-cn-hangzhou.aliyuncs.com</HostId>
```

```
</Error>
```

6.13 GetBucketReferer

GetBucketReferer operation is used to view the referer configuration of a bucket.

Request syntax

```
GET /? referer HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
RefererConfiguration	Container	The container that saves the Referer configuration content. Sub-nodes: AllowEmptyReferer node and RefererList node Parent node: none
AllowEmptyReferer	enumerative string	Specify whether the access request with the referer field being null is allowed. Valid value: `true` or`false` Default value: `true` Parent node: RefererConfiguration
RefererList	container	The container that saves the referer access whitelist. Parent node: RefererConfiguration Sub-node: Referer
Referer	String	Specify a referer access whitelist. Parent node: RefererList

Detail analysis

- If the bucket does not exist, error 404 is returned. Error code: NoSuchBucket.

- Only the owner of a bucket can view the referer configuration of the bucket. If other users attempt to access the configuration, the error 403 Forbidden with the error code: AccessDenied is returned.

- If no referer configuration has been conducted for the bucket, OSS returns the default AllowEmptyReferer value and an empty RefererList.

Example

Request example:

```
Get /? referer HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51
NNHD39zXw=
```

Response example with a referer rule configured for the bucket:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<Allowemptyreferer> true </allowemptyreferer>
    <RefererList>
        <Referer> http://www.aliyun.com</Referer>
        <Referer> https://www.aliyun.com</Referer>
        <Referer> http://www. *.com</Referer>
        <Referer> https://www.?.aliyuncs.com</Referer>
    </RefererList>
</RefererConfiguration>
```

Response example with no referer rule configured for the bucket:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList />
</RefererConfiguration>
```

6.14 GetBucketLifecycle

`GetBucketLifecycle` is used to view the lifecycle configuration of a bucket.

Request syntax

```
GET /? lifecycle HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: GMT Date
```

```
Authorization: SignatureValue
```

Detail analysis

- Only the owner of a bucket can view the lifecycle configuration of the bucket. If other users
  attempt to access the configuration, the error 403 Forbidden with the error code: AccessDenied
  is returned.

- If the bucket or lifecycle does not exist, the error 404 Not Found with the error code:
  NoSuchBucket or NoSuchLifecycle is returned.

Example

Request example:

```
Get /? lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Mon, 14 Apr 2014 01:17:29 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSp
YbJ3naA=
```

Response example with bucket lifecycle configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: keep-alive
Content-Length: 255
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<LifecycleConfiguration>
  <Rule>
    <ID>delete after one day</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>1</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Response example with no bucket lifecycle configured:

```
HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: keep-alive
Content-Length: 278
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <BucketName>oss-example</BucketName>
  <Code>NoSuchLifecycle</Code>
  <Message>No Row found in Lifecycle Table.</Message>
```

```
    <RequestId>534B372974E88A4D89060099</RequestId>
    <HostId> BucketName.oss.example.com</HostId>
</Error>
```

6.15 DeleteBucket

DeleteBucket is used to delete a bucket.

Request syntax

```
DELETE / HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- If a bucket does not exist, the error "404 no content" is returned. Error code: NoSuchBucket.

- To prevent the trouble caused by accidental deletion, OSS does not allow the bucket owner to delete a non-empty bucket.

- If you try to delete a non-empty bucket, the error 409 Conflict with the error code: BucketNotEmpty is returned.

- Only the bucket owner has the permission to delete the bucket. If you try to delete a bucket you have no permission for, the error 403 Forbidden is returned. Error code: AccessDenied.

Example

Request example:

```
DELETE / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSp
YbJ3naA=
```

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 0
```

```
Server: AliyunOSS
```

6.16 DeleteBucketLogging

DeleteBucketLogging interface is used to disable the access logging function of a bucket.

Request syntax

```
DELETE /? logging HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- If the bucket does not exist, Error 404 No Content with the error code "NoSuchBucket" is returned.

- Only the bucket owner can disable the access logging function for the bucket. If you try to operate a bucket which does not belong to you, OSS returns the error 403 Forbidden with the error code: AccessDenied.

- If the access logging function is not enabled for the target bucket, HTTP status code 204 is returned.

Example

Request example:

```
DELETE /? logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:6ZVHOehYzxoC1yxRydPQs/
CnMZU=
```

Return example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

6.17 DeleteBucketWebsite

The **DeleteBucketWebsite** is used to disable the static website hosting mode of a bucket.

Request syntax

```
DELETE /? website HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
```

```
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- If the bucket does not exist, the error 404 No Content with the error code: NoSuchBucket is
  returned.

- Only the bucket owner can disable the bucket's static website hosting mode. If you try to
  operate a bucket which does not belong to you, OSS returns the error 403 Forbidden with the
  error code: AccessDenied.

Example

Request example:

```
DELETE /? website HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:LnM4AZlOeIduZF5vGFWi
cOMEkVg=
```

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

6.18 DeleteBucketLifecycle

The `DeleteBucketLifecycle` interface is used to delete the lifecycle configuration of a
specified bucket.

Request syntax

```
DELETE /? lifecycle HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- This operation deletes all lifecycle rules of a specified bucket. After that, no objects are
  automatically deleted in this bucket.

- If the bucket or lifecycle does not exist, a 404 not found error is returned. The error code is
  NoSuchBucket or NoSuchLifecycle.

- Only the bucket owner can delete the lifecycle configuration of a bucket. If you try to operate a bucket which does not belong to you, OSS returns the 403 Forbidden error with the error code : AccessDenied.Forbidden error, error code: AccessDenied.

Examples

Request example:

```
DELETE /? lifecycle HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: Mon, 14 Apr 2014 01:17:35 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:6ZVHOehYzxoC1yxRydPQs/
CnMZU=
```

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:35 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

7 Object operations

7.1 PutObject

The `PutObject` operation is used to upload files.

Request syntax

```
PUT /ObjectName HTTP/1.1
Content-Length : ContentLength
Content-Type: ContentType
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request header

Name	Type	Description
`Cache-Control`	String	Specifies the web page caching behavior when the object is downloaded. For more information, see *RFC2616*. Default: None
`Content-Disposition`	String	Specifies the name of the object when the object is downloaded. For more information, see *RFC2616*. Default: None
`Content-Encoding`	String	Specifies the content encoding format when the object is downloaded. For more information, see *RFC2616*. Default: None
`Content-Md5`	String	As defined in RFC 1864, the message content (excluding the header) is calculated to obtain an MD5 value, which is a 128-bit number. This number is encoded using Base64 into a Content-MD5 value. This request header can be used to check the validity of a message, that is, whether the message content is consistent with the sent content. Although this request header is optional, OSS recommends that you use this request header for an end-to-end check. Default: None Restrictions: None
`Expires`	String	Specifies the expiration time. For more information, see *RFC2616*. Default: None **Note:** OSS imposes no limits or verification on this value.

Name	Type	Description
`x-oss-server-side -encryption`	String	Specifies the server-side encryption algorithm when OSS creates an object. Valid values: `AES256` and `KMS` 📋 **Note:** You must enable the KMS (Key Management Service) on the console to use the KMS encryption algorithm. Otherwise, a KmsServiceNotenabled error code is reported.
`x-oss-object-acl`	String	Specifies the access permission when OSS creates an object. Valid values: `public-read`, `private`, and `public-read-write`

Detail analysis

- If you have uploaded the Content-MD5 request header, OSS calculates Content-MD5 of the body and checks if the two are consistent. If the two are different, the error code InvalidDigest is returned.

- If the Content-Length value in the request header is smaller than the length of data transmitted in the actual request body, OSS still creates a file, but the object size is equal to the size defined by Content-Length, and the remaining data is dropped.

- If a file of the same name with the object to be added already exists, and you are authorized to access this object, the newly-added file overwrites the existing file, and the system returns the 200 OK message.

- If the PutObject request carries a parameter prefixed with x-oss-meta-, the parameter is treated as user meta, for example, x-oss-meta-location. A single object can have multiple similar parameters, but the total size of all user meta cannot exceed 8 KB.

- If the header is not chunked eoncoding and the Content length parameter is not added, the system returns the 411 Length Required error. Error code: MissingContentLength.

- If the length is set, but the message body is not sent, or the size of the sent body is smaller than the specified size, the server waits until time-out, and then returns the 400 Bad Request message. Error code: RequestTimeout.

- If the bucket of the object to be added does not exist, the system returns the 404 Not Found error. Error code: NoSuchBucket.

- If you have no permission to access the bucket of the object to be added, the system returns the 403 Forbidden error. Error code: AccessDenied.

- If the length of the added file exceeds 5 GB, the system returns the 400 Bad Request message . Error code: InvalidArgument.

- If the length of the input object key exceeds 1023 bytes, the system returns the 400 Bad Request message. Error code: InvalidObjectName.

- When you put an object, OSS supports the following five header fields defined in *RFC2616*: Cache-Control, Expires, Content-Encoding, Content-Disposition, and Content-Type. If these headers are set when you upload an object, the corresponding header values are automatically set to the uploaded values next time when this object is downloaded.

- If the x-oss-server-side-encryption header is specified when you upload an object, the value of this header must be set to AES256. Otherwise, the system returns the 400 error and the error code: InvalidEncryptionAlgorithmError. After this header is specified, the response header also contains this header, and OSS stores the encryption algorithm of the uploaded object. When this object is downloaded, the response header contains x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this object.

Common problems

Content-MD5 calculation method error

The upload content "0123456789" is used as an example. The Content-MD5 value of the string is calculated.

- Correct calculation method

  The algorithm defined in related standards is used to calculate Content-MD5 in the following process:

  1. Calculate the upload Content-MD5, whose value is a 128-bit binary array.

  2. Encode the binary array with Base64.

  Python is used as an example. The correct calculation code is as follows:

```
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789")
>>> base64.b64encode(hash.digest())
```

```
  'eB5eJF1ptWaXm4bijSPyxw=='
```

The result of `hash.digest()` is used as the input of Base64 encoding.

```
>>> hash.digest()
'x\x1e^$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'
```

- Incorrect calculation method

   A common mistake is to encode the calculated 32-byte `MD5 string` with Base64. Incorrect

   example: Encoding `hash.hexdigest()` with Base64:

```
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
>>>
>>> # The following is the incorrect calculation result.
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

- Investigation method

   You can perform investigation using the following method:

   1. The final calculated Content-MD5 is a 24-byte visible string.

   2. Use "0123456789" as the content to check whether Content-MD5 is `eB5eJF1ptW`
      `aXm4bijSPyxw==`.

Example

Request example:

```
PUT /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Cache-control: no-cache
Expires: Fri, 28 Feb 2012 05:38:42 GMT
Content-Encoding: utf-8
Content-Disposition: attachment;filename=oss_download.jpg
Date: Fri, 24 Feb 2012 06:03:28 GMT
Content-Type: image/jpg
Content-Length: 344606
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2PR
rk=

[344606 bytes of object data]
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Sat, 21 Nov 2015 18:52:34 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5650BD72207FB30443962F9A
x-oss-bucket-version: 1418321259
```

```
ETag: "A797938C31D59EDD08D86188F6D5B872"
```

7.2 CopyObject

`CopyObject` is used to copy an existing object in OSS into another object.

You can send a PUT request to OSS, and add the element "x-oss-copy-source" to the PUT request header to specify the copy source. OSS automatically determines that this is a Copy Object operation, and directly performs this operation on the server side. If the Copy Object operation is successful, the system returns new object information.

This operation is applicable to a file smaller than 1 GB. To copy a file greater than 1 GB, you must use the Multipart Upload operation. For more information about this operation, see *UploadPartCopy*.

> **Note:**
>
> For the Copy Object operation, the source bucket and the target bucket must be in the same region.

Request syntax

```
PUT /DestObjectName HTTP/1.1
Host: DestBucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
x-oss-copy-source: /SourceBucketName/SourceObjectName
```

Request header

Name	Type	Description
`x-oss-copy-source`	String	Specifies the copy source address (the requester must have the permission to read the source object). Default: None
`x-oss-copy-source-if-match`	String	If the source object's ETag value is same as the ETag value provided by the user, a COPY operation is executed, and the code 200 is returned. Otherwise, the system returns the HTTP error code 412 (preprocessing failed). Default: None
`x-oss-copy-source-if-none-match`	String	If the source object's ETag value is not the same as the ETag value provided by the user, a COPY operation is executed, and the code 200 is returned. Otherwise, the system returns the HTTP error code 304 (preprocessing failed).

Name	Type	Description
		Default: None
x-oss-copy-source-if-unmodified-since	String	If the time specified by the received parameter is same as or later than the modification time of the file, the system transfers the file normally, and returns 200 OK; otherwise, the system returns 412 Precondition Failed. Default: None
x-oss-copy-source-if-modified-since	String	If the source object has been modified after the time specified by the user, the system performs a COPY operation. Otherwise, the system returns the 304 HTTP error code (preprocessing failed). Default: None
x-oss-metadata-directive	String	Valid values include COPY and REPLACE. If this parameter is set to COPY, the system copies meta for the new object from the source object. If this parameter is set to REPLACE, the system ignores all meta values of the source object, and uses the meta value specified in this request. If this parameter is set to a value other than COPY and REPLACE, the system returns the 400 Bad Request message. Note that when the value is COPY, the source object's x-oss-server-side-encryption meta value cannot be copied. Default value: COPY Valid values: COPY and REPLACE
x-oss-server-side-encryption	String	Specifies the server-side entropy encryption algorithm when OSS creates the target object. Valid values: AES256 or KMS 📋 **Note:** You must enable the KMS (Key Management Service) on the console to use the KMS encryption algorithm. Otherwise, a KmsServiceNotenabled error code is reported.
x-oss-object-acl	String	Specifies the access permission when OSS creates an object. Valid values: public-read, private, public-read-write

Response elements

Name	Type	Description
CopyObjectResult	String	Object copying result

Name	Type	Description
		Default: None
ETag	String	ETag value of the new object. Parent element: CopyObjectResult
LastModified	String	Last update time of the new object. Parent element: CopyObjectResult

Detail analysis

- You can use the Copy Object operation to modify the meta information of an existing object.

- If the source object address is the same as the target object address in the Copy Object operation, the system directly replaces the meta information in the source object regardless of the value of x-oss-metadata-directive.

- OSS allows the Copy Object request to contain any number of the four pre-judgment headers. For more information about the related logic, see Detail Analysis of Get Object.

- To complete a Copy Object operation, the requester must have the permission to read the source object.

- The source object and the target object must belong to the same data center. Otherwise, the system returns the error code 403 AccessDenied. The error message is Target object does not reside in the same data center as source object.

- In the billing statistics of the Copy Object operation, the number of Get requests increases by 1 in the bucket of the source object, the number of Put requests increases by 1 in the bucket of the target object, and a storage space is added accordingly.

- In the Copy Object operation, all relevant request headers start from x-oss-, and therefore must be added to the signature string.

- If the x-oss-server-side-encryption header is specified in the Copy Object request, and its value (AES256) is valid, the target object is encrypted on the server side after the Copy Object operation is performed no matter whether the source object has been encrypted on the server side. In addition, the Copy Object response header contains x-oss-server-side-encryption, the value of which is set to the encryption algorithm of the target object. When this target object is downloaded, the response header also contains x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this target object. If the x-oss-server-side-encryption request header is not specified in the Copy Object operation, the target object is the data that is not encrypted on the server side even if the source object has been encrypted on the server side.

- When the x-oss-metadata-directive header in the Copy Object request is set to COPY (default value), the system does not copy the x-oss-server-side-encryption value of the source object. That is, the target object is encrypted on the server side only when x-oss-server-side-encryption is specified accordingly in the Copy Object request.

- When the x-oss-server-side-encryption request header is specified in the COPY operation, and the request value is not AES256, the system returns Error 400 with the error code "InvalidEncryptionAlgorithmError".

- If the size of the file to be copied is greater than 1 GB, the system returns Error 400 with the error code "EntityTooLarge".

- This operation cannot be used to copy objects created by Append Object.

- If the file type is symbolic link, copy the symbolic link only.

Example

Request example:

```
PUT /copy_oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 07:18:48 GMT
x-oss-copy-source: /oss-example/oss.jpg
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:gmnwPKuu20LQEjd+iPkL259A+
n0=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Content-Type: application/xml
Content-Length: 193
Connection: keep-alive
Date: Fri, 24 Feb 2012 07:18:48 GMT
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<CopyObjectResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
 <LastModified>Fri, 24 Feb 2012 07:18:48 GMT</LastModified>
 <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE"</ETag>
</CopyObjectResult>
```

7.3 GetObject

`GetObject` is used to obtain an object which you must have the permission to read.

Request syntax

```
GET /ObjectName HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

```
Range: bytes=ByteRange (Optional)
```

Request parameters

When sending a `GET` request, you can customize some headers in OSS response. If you send an anonymous user request, you can only customize `content-type` headers. Other headers require that you send`GET` requests with signatures. These headers include:

Name	Type	Description
`response-content-type`	string	Specifies the content-type header in the request returned by OSS. If you send an anonymous user request, you can also specify the content-type header. Default value: None
`response-content-language`	string	Specifies the content-language header in the request returned by OSS. Default value: None
`response-expires`	string	Specifies the expires header in the request returned by OSS. Default value: None
`response-cache-control`	string	Specifies the cache-control header in the request returned by OSS. Default value: None
`Response-content-Disposition`	string	Specifies the content-disposition header in the request returned by OSS. Default value: None
`response-content-encoding`	string	Specifies the content-encoding header in the request returned by OSS. Default value: None

Request header

Name	Type	Description
`Range`	string	Specifies the range of file transfer. For example, if the range is set to bytes = 0-9, the system transfers byte 0 to byte 9. Default value: None
`If-Modified-Since`	string	If the specified time is earlier than the actual modification time, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 304 Not Modified message. Default value: None

Name	Type	Description
		Time format: GMT, for example: Fri, 13 Nov 2015 14:47:53 GMT
If-Unmodified-Since	string	If the specified time is same as or later than the actual file modification time, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 412 Precondition Failed message. Default value: None Time format: GMT, for example: Fri, 13 Nov 2015 14:47:53 GMT
If-Match	string	If the expected ETag that is introduced matches the ETag of the object, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 412 Precondition Failed message. Default value: None
If-None-Match	string	If the introduced ETag does not match the ETag of the object, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 304 Not Modified message. Default value: None

Detail analysis

- The Range parameter in the Get Object request can be set to support resumable data transfer from breakpoints. This function is recommended if the object size is large.

- If the Range parameter is used in the request header, the returned message includes the length of the entire file and the range returned for the request. For example, if the returned message is Content-Range: bytes 0-9/44, it means that the length of the entire file is 44, and the range returned is 0 to 9. If the range requirement is not met, the system transfers the entire file and does not include Content-Range in the result.

- If the time specified by If-Modified-Since does not match the actual modification time, the system directly returns the file and the 200 OK message.

- If-Modified-Since can coexist with If-Unmodified-Since. If-Match can also coexist with If-None-Match.

- If the request contains If-Unmodified-Since and If-Unmodified-Since does not match the actual modification time, or the request contains If-Match and If-Match does not match the Etag of the object, the system returns the 412 Precondition Failed message.

- If the request contains If-Modified-Since and If-Modified-Since does not match the actual modification time, or the request contains If-None-Match and If-None-Match does not match the ETag of the object, the system returns Error 304 Not Modified.

- If the file does not exist, the system returns Error 404 Not Found. The error code is NoSuchKey.

- OSS does not allow you to customize the headers in OSS returned request by using request parameters in the GET request during an anonymous access.

- When you customize some headers in OSS returned request, OSS sets these headers to the values specified by parameters in the GET Object Request only when the request is successfully processed, that is, when the system returns the 200 OK message.

- If this object is encrypted on the server side, the system automatically returns the decrypted object on receiving the GET Object request, and returns x-oss-server-side-encryption in the response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm of the object.

- If you want to compress and transfer the returned content using GZIP, add Accept-Encoding:gzip to the display mode in the request header. OSS determines whether to return the data compressed by GZIP to you based on the Content-Type and size of the file. If the content is compressed using GZIP, the content does not contain the Etag. Currently, OSS supports GZIP compression for the following Content-Types: HTML, Javascript, CSS, XML, RSS, and JSON, and the file size must be at least 1 KB.

- If the file type is symbolic link, the content of the target file is returned. In the response header, `Content-Length`, `ETag`, and `Content-Md5` are metadata of the target file, `Last-Modified` is the maximum value of the target file and symbolic link, and others are metadata of symbolic links.

- If the file type is symbolic link and the target file does not exist, the system returns Error 404 Not Found. The error code is SymlinkTargetNotExist.

- If the file type is symbolic link and the target file type is symbolic link, the system returns Error 400 Bad request. The error code is InvalidTargetType.

- For the Archive type, submit the Restore request and complete Restore before downloading an object. The object can be downloaded only when the Restore operation is completed and not timed-out.

  — If the Restore request is not submitted or the last Restore operation is time-out, the system returns Error 403. The error code is "InvalidObjectState".

— If the Restore request has been submitted but the Restore operation is not completed, the system returns Error 403. The error code is "InvalidObjectState".

— Data can be directly downloaded only when the Restore operation is completed and not timed-out.

Example

Request example:

```
GET /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:38:30 GMT
Authorization:OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJCZkcde6OhZ9Jfe8
=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 3a89276f-2e2d-7965-3ff9-51c875b99c41
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 06:38:30 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Content-Type: image/jpg
Content-Length: 344606
Server: AliyunOSS
[344606 bytes of object data]
```

Request example with range specified:

```
GET //oss.jpg HTTP/1.1
Host:oss-example. oss-cn-hangzhou.aliyuncs.com
Date: Fri, 28 Feb 2012 05:38:42 GMT
Range: bytes=100-900
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:qZzjF3DUtd+yK16BdhGtF
cCVknM=
```

Response example:

```
HTTP/1.1 206 Partial Content
x-oss-request-id: 28f6508f-15ea-8224-234e-c0ce40734b89
x-oss-object-type: Normal
Date: Fri, 28 Feb 2012 05:38:42 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Accept-Ranges: bytes
Content-Range: bytes 100-900/344606
Content-Type: image/jpg
Content-Length: 801
Server: AliyunOSS
```

```
[801 bytes of object data]
```

Request example with the returned message header customized:

```
GET /oss.jpg? response-expires=Thu%2C%2001%20Feb%202012%2017%3A00%3A00
%20GMT& response-content-type=text&response-cache-control=No-cache&
response-content-disposition=attachment%253B%2520filename%253Dtesting
.txt&response-content-encoding=utf-8&response-content-language=%E4%B8%
AD%E6%96%87 HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com:
Date: Fri, 24 Feb 2012 06:09:48 GMT
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 06:09:48 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Content-Length: 344606
Connection: keep-alive
Content-disposition: attachment; filename:testing.txt
Content-language: Chinese
Content-encoding: utf-8
Content-type: text
Cache-control: no-cache
Expires: Fri, 24 Feb 2012 17:00:00 GMT
Server: AliyunOSS
[344606 bytes of object data]
```

Request example of symbolic link:

```
GET /link-to-oss.jpg HTTP/1.1
Accept-Encoding: identity
Date: Tue, 08 Nov 2016 03:17:58 GMT
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Authorization:OSS qn6qrrqxo2oawuk53otfjbyc:qZzjF3DUtd+yK16BdhGtFcCVknM
=
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 08 Nov 2016 03:17:58 GMT
Content-Type: application/octet-stream
Content-Length: 20
Connection: keep-alive
x-oss-request-id: 582143E6D3436A212ADCC87D
Accept-Ranges: bytes
ETag: "8086265EFC0211ED1F9A2F09BF462227"
Last-Modified: Tue, 08 Nov 2016 03:17:58 GMT
x-oss-object-type: Symlink
```

```
Content-MD5: gIYmXvwCEe0fmi8Jv0YiJw==
```

Example of a request when the Restore operation of an object of the Archive type is completed:

```
GET /oss.jpg HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Sat, 15 Apr 2017 09:38:30 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:zUglwRPGkbByZxm1+y4eyu+
NIUs=
```

Response example

```
HTTP/1.1 200 OK
x-oss-request-id: 58F723894529F18D7F000053
x-oss-object-type: Normal
x-oss-restore: ongoing-request="false", expiry-date="Sun, 16 Apr 2017
08:12:33 GMT"
Date: Sat, 15 Apr 2017 09:38:30 GMT
Last-Modified: Sat, 15 Apr 2017 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Content-Type: image/jpg
Content-Length: 344606
Server: AliyunOSS
[354606 bytes of object data]
```

7.4 AppendObject

`AppendObject` is used to upload files in appending mode.

The type of the objects created with the Append Object operation is Appendable Object, and the type of the objects uploaded with the Put Object operation is Normal Object.

Request syntax

```
POST /ObjectName? append&position=Position HTTP/1.1
Content-Length :ContentLength
Content-Type: ContentType
Host: BucketName.oss.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request header

Name	Type	Description
Cache-Control	String	Specifies the cache action of the web page when the object is downloaded. For more information, see *RFC2616*. Default: None
Content-Disposition	String	Specifies the name of the object downloaded. For more information, see *RFC2616*. Default: None

Name	Type	Description
`Content-Encoding`	String	Specifies the content encoding format when the object is downloaded. For more information, see *RFC2616*. Default: None
`Content-MD5`	String	As defined in RFC1864, the message content (excluding the header) is computed to obtain an MD5 value, which is a 128-bit number. Then, this number is Base64-encoded into a Content-MD5 value. This request header can be used for checking the validity of a message, that is, whether the message content is consistent with the sent content. Although this request header is optional, we recommend that you use this request header for end-to-end check. Default: None Restrictions: None
`Expires`	Integer	Indicates the expiration time. For more information, see *RFC2616*. Default: None
`x-oss-server-side-encryption`	String	Specifies the server-side encryption algorithm when OSS creates an object. Valid value: `AES256` or `KMS` **Note:** You must enable the KMS (Key Management Service) on the console to use the KMS encryption algorithm. Otherwise, a KmsServiceNotenabled error code is reported.
`x-oss-object-acl`	String	Specifies the access permission when OSS creates an object. Valid values: `public-read`, `private`, and `public-read-write`

Response header

Name	Type	Description
`x-oss-next-append-position`	64-bit integer	Specifies the position that must be provided in the next request. It is in fact the current object length. This header is contained when a successful message is returned for Append Object, or when a 409 error occurs because of a position mismatch and the object length.

Name	Type	Description
`x-oss-hash-crc64ecma`	64-bit integer	Specifies the 64-bit CRC value of the object. The 64-bit CRC value is calculated according to *ECMA-182 Standard* .

Association with other operations

- Append Object is not applicable to a non-appendable object. For example, if a normal object with the same name already exists and the Append Object operation is still performed, the system returns the 409 message and the error code ObjectNotAppendable.

- If you perform the Put Object operation on an existing appendable object, this appendable object is overwritten by the new object, and the type of this object 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. If the object is an appendable object, the value of x-oss-object-type is Appendable. For an appendable object, after the Head Object operation is performed, the system also returns x-oss-next-append-position and x-oss-hash-crc64ecma.

- In the response XML of the Get Bucket (List Objects) request, the type of an appendable object is set to Appendable.

- You can neither use Copy Object to copy an appendable object, nor change the server-side encryption attribute of this object. You can use Copy Object to modify the custom metadata.

Detail analysis

- The two URL parameters, append, and position, are both CanonicalizedResource, and must be contained in the signature.

- URL parameters must also contain append, which specifies that the operation is an Append Object operation.

- URL query parameters must contain position, which specifies the position from where appending starts. The value of position in the first Append Object operation must be 0, and the value of position in the subsequent operation is the current object length. For example, if the value of position specified in the first Append Object request is 0, and the value of content-length is 65536, the value of position specified in the second Append Object request must be set to 65536. After each operation succeeds, x-oss-next-append-position in the response header also specifies the position of the next Appendix Object request.

- If the position value is different from the current object length, OSS returns the 409 message and the error code PositionNotEqualToLength. If such an error occurs, you can obtain the

position for the next Append Object request from x-oss-next-append-position in the response header, and send an Append Object request again.

- If the position value is 0 and an appendable object with the same name does not exist, or if the length of an appendable object with the same name is 0, the Append Object operation is successful; otherwise, the system regards that the position and object length are mismatched.

- If the position value is 0 and an object with the same name does not exist, headers (such as x-oss-server-side-encryption) can be set in the Append Object request like the Put Object request. This is the same as the case of Initiate Multipart Upload. If the position value is 0, and the correct x-oss-server-side-encryption header is added to the request, the header of the response to the subsequent Append Object request also contains x-oss-server-side-encryption , which indicates the encryption algorithm. Later, if meta must be modified, you can use the Copy Object request.

- Because of the concurrency, even if you set the value of position to x-oss-next-append-position , this request can still fails owing to the PositionNotEqualToLength.

- The length limit of an object generated by Append Object is the same as that of an object generated by Put Object.

- After each Append Object operation, the last modification time of this object is updated.

- If the position value is correct and the content with the length of 0 is appended to an existing appendable object, this operation does not change the status of the object.

- If the bucket type is Archive, you cannot call this interface; otherwise, the system returns Error 400 with the error code "OperationNotSupported".

CRC64 computing method

The CRC value of an appendable object is computed according to *ECMA-182 Standard*. Its computing method is the same as that of XZ. CRC64 can be computed as follows using the boost CRC module:

```
typedef boost::crc_optimal<64, 0x42F0E1EBA9EA3693ULL, 0xffffffff
ffffffffULL, 0xffffffffffffffffULL, true, true> boost_ecma;
uint64_t do_boost_crc(const char* buffer, int length)
{
    boost_ecma crc;
    crc.process_bytes(buffer, length);
    return crc.checksum();
}
```

Alternatively, CRC64 can be computed as follows using the Python crcmod:

```
do_crc64 = crcmod.mkCrcFun(0x142F0E1EBA9EA3693L, initCrc=0L, xorOut=
0xffffffffffffffffL, rev=True)
```

```
print do_crc64("123456789")
```

Example

Request example:

```
POST /oss.jpg? append&position=0 HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Cache-control: no-cache
Expires: Wed, 08 Jul 2015 16:57:01 GMT
Content-Encoding: utf-8
Content-Disposition: attachment;filename=oss_download.jpg
Date: Wed, 08 Jul 2015 06:57:01 GMT
Content-Type: image/jpg
Content-Length: 1717
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2PR
rk=

[1717 bytes of object data]
```

Return example:

```
HTTP/1.1 200 OK
Date: Wed, 08 Jul 2015 06:57:01 GMT
ETag: "0F7230CAA4BE94CCBDC99C5500000000"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-hash-crc64ecma: 14741617095266562575
x-oss-next-append-position: 1717
x-oss-request-id: 559CC9BDC755F95A64485981
```

7.5 DeleteObject

The `DeleteObject` operation is used to delete an object.

Request syntax

```
DELETE /ObjectName HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- To delete an object with Delete Object, you must have the write permission to this object.

- If the object to be deleted does not exist, OSS returns the 204 No Content status code.

- If the bucket of the object does not exist, the system returns 404 Not Found.

- If the file type is **symbolic link**, only the symbolic links are deleted.

Example

Request example:

```
DELETE /copy_oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 07:45:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:zUglwRPGkbByZxm1+y4eyu+
NIUs=
```

Response example:

```
HTTP/1.1 204 NoContent
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Fri, 24 Feb 2012 07:45:28 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

7.6 DeleteMultipleObjects

The `DeleteMultipleObjects` operation allows you to delete multiple objects in the same bucket with one HTTP request.

You can perform the `DeleteMultipleObjects` operation to delete up to 1,000 objects with one request. Two response modes are available: the Verbose mode and the Quiet mode.

- Verbose mode: The message body returned by OSS contains the result of each deleted object.

- Quiet mode: The message body returned by OSS only contains the results for objects which encountered an error in the DELETE process. If all objects are successfully deleted, no message body is returned.

Request syntax

```
POST /? delete HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: ContentLength
Content-MD5: MD5Value
Authorization: SignatureValue
<? xml version="1.0" encoding="UTF-8"? >
<Delete>
  <Quiet>true</Quiet>
  <Object>
    <Key>key</Key>
  </Object>
…
```

```
</Delete>
```

Request parameters

During the Delete Multiple Objects operation, you can use encoding-type to encode the Key in the
 returned result.

Name	Description
`encoding-type`	Specify the encoding type of the Key in the returned result. Currently, the URL encoding is supported. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Data type: String Default: None Optional value: `url`

Request elements

Name	Type	Description
`Delete`	Container	Specify the container that saves the Delete Multiple Objects request. Sub-nodes: one or more object elements and the optional quiet element Parent node: None
`Key`	String	Specify the name of the deleted object. Parent node: Object
`Object`	Container	Specify the container that saves the information about the object. Sub-node: key Parent node: Delete
`Quiet`	enumerative string	Enables the "Quiet" response mode.

Name	Type	Description
		Valid values: `true`, `false` Default: `false` Parent node: Delete

Response elements

Name	Type	Description
Deleted	Container	Specify the container that saves the successfully deleted objects. Sub-node: key Parent node: DeleteResult
DeleteResult	Container	Specify the container that saves the result of the Delete Multiple Objects request. Sub-node: Deleted Parent node: None
Key	String	Specify the name of the object on which OSS performs the Delete operation. Parent node: Deleted
EncodingType	String	Specify the encoding type for the returned results. If encoding-type is specified in the request, the Key is encoded in the returned result. Parent node: Container

Detail analysis

- The Content-Length and Content-MD5 fields must be specified in the Delete Multiple Objects request. OSS verifies that the received message body is correct based on the two fields before performing the Delete operation.

- Method to generate the Content-MD5 field: Encrypt the MD5 value of the Delete Multiple Objects request to obtain a 128-byte array, and encode the array using Base64. The final string obtained is the content of the Content-MD5 field.

- The return mode of the Delete Multiple Objects request is Verbose by default.

- If the Delete Multiple Objects request is used to delete a non-existing object, the operation is still regarded as successful.

- The Delete Multiple Objects request can contain a message body of up to 2 MB. If the size of the message body exceeds 2 MB, the system returns the MalformedXML error code.

- The Delete Multiple Objects request can be used to delete up to 1,000 objects at a time. If the number of objects to be deleted at a time exceeds 1,000, the system returns the MalformedX ML error code.

- If you have uploaded the Content-MD5 request header, OSS calculates the body's Content-MD5 and check if the two are consistent. If the two are different, the error code InvalidDigest is returned.

Example

Request example I:

```
POST /? delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length:151
Content-MD5: ohhnqLBJFiKkPSBO1eNaUA==
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:+z3gBfnFAxBcBDgx27Y/
jEfbfu8=
<? xml version="1.0" encoding="UTF-8"? >
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
  </Object>
  <Object>
    <Key>test.jpg</Key>
  </Object>
  <Object>
    <Key>demo.jpg</Key>
  </Object>
</Delete>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 78320852-7eee-b697-75e1-b6db0f4849e7
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length: 244
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<? xml version="1.0" encoding="UTF-8"? >
<DeleteResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
    <Deleted>
        <Key>multipart.data</Key>
    </Deleted>
    <Deleted>
        <Key>test.jpg</Key>
```

```
        </Deleted>
        <Deleted>
            <Key>demo.jpg</Key>
        </Deleted>
</DeleteResult>
```

Request Example II:

```
POST /? delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length:151
Content-MD5: ohhnqLBJFiKkPSBO1eNaUA==
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:WuV0Jks8RyGSNQrBca64
kEExJDs=
<? xml version="1.0" encoding="UTF-8"? >
<Delete>
  <Quiet>true</Quiet>
  <Object>
    <Key>multipart.data</Key>
  </Object>
  <Object>
    <Key>test.jpg</Key>
  </Object>
  <Object>
    <Key>demo.jpg</Key>
  </Object>
</Delete>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

7.7 HeadObject

`HeadObject` is used to return the meta information of a certain object without returning the file content.

Request syntax

```
HEAD /ObjectName HTTP/1.1
Host: BucketName/oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Request header

Name	Type	Description
If-Modified-Since	String	If the specified time is earlier than the actual modification

Name	Type	Description
		time, the system returns the 200 OK message and the object metadata; otherwise, the system returns the 304 Not Modified message. Default: None
`If-Unmodified-Since`	String	If the specified time is same as or later than the actual file modification time, the system returns the 200 OK message and the object metadata; otherwise, the system returns the 412 Precondition Failed message. Default: None
`If-Match`	String	If the expected ETag that is introduced matches the ETag of the object, the system returns the 200 OK message and the object metadata; otherwise, the system returns the 412 Precondition Failed message. Default: None
`If-None-Match`	String	If the introduced ETag does not match the ETag of the object, the system returns the 200 OK message and the object metadata; otherwise, the system returns the 304 Not Modified message. Default: None

Detail analysis

- After the Head Object request is sent, no message body is returned even if the system returns the 200 OK message or an error message.

- The If-Modified-Since, If-Unmodified-Since, If-Match, and If-None-Match query conditions can be set in the header of the Head Object request. For the detailed setting rules, see the related

fields in the Get Object request. If no modification is made, the system returns the 304 Not
Modified message.

- If you upload the user meta prefixed with x-oss-meta- when sending a Put Object request, for
  example, x-oss-meta-location, the user meta is returned.

- If the file does not exist, the system returns Error 404 Not Found.

- If this object is entropy encrypted on the server, the system returns x-oss-server-side-
  encryption in the header of the response to the Head Object request. The value of x-oss-server
  -side-encryption indicates the server-side encryption algorithm of the object.

- If the file type is symbolic link, in the response header, `Content-Length`, `ETag`, and
  `Content-Md5` are metadata of the target file, `Last-Modified` is the maximum value of the
  target file and symbolic link, and others are metadata of symbolic links.

- If the file type is symbolic link and the target file does not exist, the system returns Error 404
  Not Found. The error code is "SymlinkTargetNotExist".

- If the file type is symbolic link and the target file type is symbolic link, the system returns Error
  400 Bad request. The error code is "InvalidTargetType".

- If the bucket type is Archive and the Restore request has been submitted, the Restore state of
  Object is indicated by x-oss-restore in the response header.

  — If the Restore request is not submitted or times out, the field is not returned.

  — If the Restore request has been submitted and does not time out, the value of x-oss-restore
    returned is ongoing-request="true".

  — If the Restore request has been submitted and completed, the value of x-oss-restore
    returned is ongoing-request="false", expiry-date="Sun, 16 Apr 2017 08:12:33 GMT".

Example

Request example:

```
HEAD /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanlJ5dLA092wpDC/
E=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
x-oss-object-type: Normal
x-oss-storage-class: Archive
Date: Fri, 24 Feb 2012 07:32:52 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
```

```
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 344606
Content-Type: image/jpg
Connection: keep-alive
Server: AliyunOSS
```

Example of a request when the Restore request has been submitted but not completed:

```
HEAD /oss.jpg HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Sat, 15 Apr 2017 07:32:52 GMT
Authorization: OSS e1Unnbm1rgdnpI:KKxkdNrUBu2t1kqlDh0MLbDb99I=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 58F71A164529F18D7F000045
x-oss-object-type: Normal
x-oss-storage-class: Archive
x-oss-restore: ongoing-request="true"
Date: Sat, 15 Apr 2017 07:32:52 GMT
Last-Modified: Sat, 15 Apr 2017 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 344606
Content-Type: image/jpg
Connection: keep-alive
Server: AliyunOSS
```

Example of a request when the Restore request has been submitted and completed:

```
HEAD /oss.jpg HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Sat, 15 Apr 2017 09:35:51 GMT
Authorization: OSS e1Unnbm1rgdnpI:21qtGJ+ykDVmdu6O6FMJnn+WuBw=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 58F725344529F18D7F000055
x-oss-object-type: Normal
x-oss-storage-class: Archive
x-oss-restore: ongoing-request="false", expiry-date="Sun, 16 Apr 2017
08:12:33 GMT"
Date: Sat, 15 Apr 2017 09:35:51 GMT
Last-Modified: Sat, 15 Apr 2017 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a39863328"
```

```
Content-Length: 344606
```

7.8 GetObjectMeta

`GetObjectMeta` is used to obtain the basic meta information of an object in a bucket, but it does not return the content. The meta information includes the Etag, Size (the file size), and LastModified.

Request syntax

```
GET /ObjectName? objectMeta HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- After the Get Object Meta request is sent, no message body is returned no matter whether the system returns the OK message or an error message.

- Get Object Meta must contain the parameters of the ObjectMeta request; otherwise, it indicates a Get Object request.

- If the file does not exist, the system returns the 404 Not Found error.

- Get Object Meta is more lightweight than Header Object. Only some basic meta information of an object in a bucket is returned. The meta information includes the Etag, Size (the file size), and LastModified. The Size is measured with the value of the Content-Length header.

- If the file type is symbolic link, only the information of the symbolic link itself is returned.

Example

Request example:

```
GET /oss.jpg? objectMeta HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0Fmg
brQ0=
```

Request example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE"
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 344606
Connection: keep-alive
```

```
Server: AliyunOSS
```

7.9 PutObjectACL

The `PutObjectACL` interface is used to modify the access permission of an object.

Currently, an object may have four types of access permissions: default, private, public-read, and
 public-read-write. You can use the "x-oss-object-acl" header in the Put Object ACL request to
set the access permission. Only the bucket owner has the permission to perform this operation.
If the operation succeeds, 200 is returned; otherwise, the corresponding error code and prompt
message are returned.

Request syntax

```
PUT /ObjectName? acl HTTP/1.1
x-oss-object-acl: Permission
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Definition of Object ACL

Name	Description
`private`	This ACL indicates that an object is a private resource. Only the owner of this object has the permission to read or write this object.
`public-read`	This ACL indicates that an object is a resource that can be read by the public. Only the owner of this object has the permission to read and write this object. Other users only have the permission to read this object.
`public-read-write`	This ACL indicates that an object is a resource that can be read and written by the public. All users have the permission to read and write this object.
`default`	This ACL indicates an object is a resource inheriting the read-write permissions of the bucket. That is, the bucket and the object have the same permissions.

Detail analysis

- Read operations to an object include: the read operations to the source object in GetObject,
  HeadObject, CopyObject, and UploadPartCopy. Write operations to an object include: the write

operations to a new object in PutObject, PostObject, AppendObject, DeleteObject, DeleteMult
ipleObjects, CompleteMultipartUpload, and CopyObject.

- The x-oss-object-acl must be set to one of the preceding four permissions. Otherwise, OSS
  returns the 400 Bad Request message and the error code is: InvalidArgument.

- You can use PutObject ACL to set the ACL of an object. In addition, when writing an object,
  you can include x-oss-object-acl in the request header to set the ACL of the object. The effect
  is equivalent to PutObject ACL. For example, if the header of the PutObject request carries x-
  oss-object-acl, you can set the ACL of an object while uploading the object.

- When a user who has no permission to read an object reads the object, OSS returns the 403
  Forbidden message and the error code is: AccessDenied. The message displayed is: You do
  not have read permission on this object.

- When a user who has no permission to write an object writes the object, OSS returns the 403
  Forbidden message and the error code is: AccessDenied. The message displayed is: You do
  not have write permission on this object.

- Only the owner of a bucket has the permission to call the PutObject ACL to modify the ACL for
  an object in this bucket. When a non-bucket owner calls the PutObject ACL, OSS returns the
  403 Forbidden message and the error code is: AccessDenied. The message displayed is: You
  do not have write acl permission on this object.

- The object ACL takes precedence over the bucket ACL. For example, if the bucket ACL is
  private and the object ACL is public-read-write, the system first checks the ACL of the object
  when a user accesses the object. As a result, all users can access this object even if the
  bucket is a private bucket. If the ACL of an object has never been set, the ACL of this object is
  same as that of the bucket where the object is located.

Example

Request example:

```
Put/test-object? acl HTTP/1.1
x-oss-object-acl: public-read
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 0
```

```
Connection: keep-alive
Server: AliyunOSS
```

7.10 GetObjectACL

The `GetObjectACL` operation is used to obtain the permission to access an object in a bucket.

Request syntax

```
GET /ObjectName? acl HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
`AccessControlList`	Container	Container used for storing the ACL information Parent node: AccessCont rolPolicy
`AccessControlPolicy`	Container	Container that stores the Get Object ACL result Parent node: None
`DisplayName`	String	Name of the bucket owner (Currently is consistent with the ID) Parent node: AccessCont rolPolicy.Owner
`Grant`	Enumerating string	The ACL permission of an object Valid values: `private`, `public-read`, and `public-read-write` Parent node: AccessCont rolPolicy.AccessControlList
ID	String	User ID of the bucket owner Parent node: AccessCont rolPolicy.Owner
Owner	Container	Container used for saving the information about the bucket owner Parent node: AccessCont rolPolicy

Detail analysis

- Only the bucket owner can use Get Object ACL to obtain the ACL of an object in the bucket. If you are not the bucket owner and send a Get Object ACL request, the system returns the 403 Forbidden message. Error code: AccessDenied. The message displayed is: You do not have read acl permission on this object.

- If a Get Object ACL request is sent but the ACL has never been set for the object, ObjectACL returned by OSS is default, indicating that the ACL of this object is the same as the bucket ACL. That is, if the access permission of the bucket is private, the access permission of this object is also private; if the access permission of the bucket is public-read-write, the access permission of this object is also public-read-write.

Example

Request example:

```
GET /test-object? acl HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0Fmg
brQ0=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 253
Content-Tupe: application/xml
Connection: keep-alive
Server: AliyunOSS

<? xml version="1.0" ? >
<AccessControlPolicy>
    <Owner>
        <ID>00220120222</ID>
        <DisplayName>00220120222</DisplayName>
    </Owner>
    <AccessControlList>
        <Grant>public-read </Grant>
    </AccessControlList>
```

```
</AccessControlPolicy>
```

7.11 PostObject

The `PostObject` operation is used to upload a file to a specified bucket using the HTML form.

As a substitute of Put Object, Post Object makes it possible to upload files to a bucket based on the browser. The message body of Post Object is encoded using multipart/form-data. In the Put Object operation, parameters are transferred through the HTTP request header.

Post object

Request syntax

```
POST / HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
User-Agent: browser_data
Content-Length : ContentLength
Content-Type: multipart/form-data; boundary=9431149156168
--9431149156168
Content-Disposition: form-data; name="key"
key
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"
success_redirect
--9431149156168
Content-Disposition: form-data; name="Content-Disposition"
attachment;filename=oss_download.jpg
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-uuid"
myuuid
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-tag"
mytag
--9431149156168
Content-Disposition: form-data; name="OSSAccessKeyId"
access-key-id
--9431149156168
Content-Disposition: form-data; name="policy"
encoded_policy
--9431149156168
Content-Disposition: form-data; name="Signature"
signature
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg
file_content
--9431149156168
Content-Disposition: form-data; name="submit"
Upload to OSS
```

Form fields

Name	Type	Description	Required or Optional
`OSSAccessKeyId`	string	Specify the AccessKey ID of the bucket owner. Default value: none Restriction: This form field is required when the bucket does not allow public-read-write, or when the Policy (or Signature) form field is provided.	Conditional
`policy`	string	Specify validity of the form fields in the request. A request that does not contain the Policy form field is treated as an anonymous request, and can only access buckets that allow public-read-write. For more information, see 5.7.4.1 Post policy. Default value: none Restriction: This form field is required when the bucket does not allow public-read-write, or when the OSSAccessKeyId (or Signature) form field is provided.	Conditional
`Signature`	string	Specify the signature information that is computed based on the Access Key Secret and Policy. The OSS checks the signature information to verify validity of the Post Object request. For	Conditional

Name	Type	Description	Required or Optional
		more information, see 5.7.4.2 Post Signature. Default value: none Restriction: This form field is required when the bucket does not allow public-read-write, or when the OSSAccessKeyId (or Policy) form field is provided.	
`Cache-Control`, `Content-Type`, `Content-Disposition`, `Content-Encoding`, `Expires`	string	REST request headers. For more information, see the related descriptions in Put Object. Default value: none	Optional
`file`	string	Specify the file or text content. It must be the last field in the form. The browser automatically sets Content-Type based on the file type, and overwrites the user setting. The OSS can only upload one file at a time. Default value: none	Required
`key`	string	Specify the object name of the uploaded file. If the object name contains forward slashes (/), such as a/b/c/b.jpg, OSS will create the corresponding directory. Default value: none	Required

Name	Type	Description	Required or Optional
`success_ac tion_redirect`	string	Specify the URL to which the client is redirected after successful upload. If this form field is not specified, the returned result is specified by success_action_status. If upload fails, the OSS returns an error code, and the client is not redirected to any URL. Default value: none	Optional
`success_ac tion_status`	string	Specify the status code returned to the client after the previous successful upload if success_action_redirect is not specified. Valid values include 200, 201, and 204 (default). If this field is set to 200 or 204, the OSS returns an empty file and a corresponding status code. If this field is set to 201, the OSS returns an XML file and the 201 status code. If this field is not specified or set to an invalid value, the OSS returns an empty file and the 204 status code. Default value: none	
`x-oss-meta-*`	string	Specify the user meta value set by the user. The OSS does not	Optional

Name	Type	Description	Required or Optional
		check or use this value. Default value: none	
`x-oss-server-side-encryption`	string	Specify the server-side encryption algorithm when the OSS creates an object. Valid value: `AES256`	Optional
`x-oss-object-acl`	string	Specify the access permission when the OSS creates an object. Valid values: `public-read`, `private`, and `public-read-write`	Optional
`x-oss-security-token`	string	If STS temporary authorization is used for this access, you must specify the item to be the SecurityToken value. At the same time, OSSAccessKeyId must use a paired temporary AccessKeyId. The signature calculation is consistent with the general AccessKeyId signature. Default value: none	Optional

Response header

Name	Type	Description
`x-oss-server-side-encryption`	string	If x-oss-server-side-encryption is specified in the request, the response contains this header, which indicates the encryption algorithm used.

Response elements

Name	Type	Description
PostResponse	container	Specify the container that saves the result of the Post Object request. Sub-nodes: Bucket, ETag, Key, and Location
Bucket	string	Specify the bucket name. Parent node: PostResponse
ETag	string	Specify the entity tag (ETag) that is created when an object is generated. For an object created by Post Object, the ETag value is the UUID of the object, and can be used to check whether the content of the object has changed. Parent node: PostResponse
Location	string	Specify the URL of the newly created object. Parent node: PostResponse

Detail analysis

- To perform the Post Object operation, you must have the permission to write the bucket. If the bucket allows public-read-write, you can choose not to upload the signature information ; otherwise, signature verification must be performed on the Post Object operation. Unlike Put Object, Post Object uses AccessKeySecret to compute the signature for the policy. The computed signature string is used as the value of the Signature form field. The OSS checks this value to verify validity of the signature.

- No matter whether the bucket allows public-read-write, once any one of the OSSAccessKeyId, Policy, and Signature form fields is uploaded, the remaining two form fields are required. If the remaining two form fields are missing, the OSS returns the error code: InvalidArgument.

- Form encoding submitted by the Post Object operation must be "multipart/form-data". That is, Content-Type in the header must be in the `multipart/form-data; boundary=xxxxxx` format, where boundary is the boundary string.

- The URL of the submitted form can be the domain name of the bucket. It is not necessary to specify the object in the URL. That is, the request line is `POST / HTTP/1.1`, and cannot be written as `POST /ObjectName HTTP/1.1`.

- The policy specifies the valid values of form fields in the Post Object request. The OSS checks validity of the request based on the policy. If the request is invalid, the OSS returns the error

code: AccessDenied. When checking validity of the policy, the OSS does not check irrelevant
form fields in the policy.

- The form and policy must be encoded with UTF-8. The policy is a JSON text encoded with UTF
-8 and Base64.

- The Post Object request can contain extra form fields. The OSS checks validity of these form
fields based on the policy.

- If you have uploaded the Content-MD5 request header, the OSS calculates the body's Content
-MD5 and check if the two are consistent. If the two are different, the error code InvalidDigest is
 returned.

- If the Post Object request contains the Header signature or URL signature, the OSS does not
check these signatures.

- If the Put Object request carries a form field prefixed with x-oss-meta-, the form field is treated
 as the user meta, for example, x-oss-meta-location. A single object can have multiple similar
parameters, but the total size of all user meta cannot exceed 8 KB.

- The total length of the body in the Post Object request cannot exceed 5 GB. When the file
length is too large, the system returns the error code: EntityTooLarge.

- If the x-oss-server-side-encryption header is specified when you upload an object, the value of
 this header must be set to AES256 or KMS. Otherwise, the system returns 400 and the error
 code: InvalidEncryptionAlgorithmError. After this header is specified, the response header
also contains this header, and the OSS stores the encryption algorithm of the uploaded object
. When this object is downloaded, the response header contains x-oss-server-side-encryption,
the value of which is set to the encryption algorithm of this object.

- Form fields are not case-sensitive, but their values are case-sensitive.

Examples

- Request example:

```
POST / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 344606
Content-Type: multipart/form-data; boundary=9431149156168
--9431149156168
Content-Disposition: form-data; name="key"
/user/a/objectName.txt
--9431149156168
Content-Disposition: form-data; name="success_action_status"
200
--9431149156168
Content-Disposition: form-data; name="Content-Disposition"
content_disposition
--9431149156168
```

```
Content-Disposition: form-data; name="x-oss-meta-uuid"
uuid
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-tag"
metadata
--9431149156168
Content-Disposition: form-data; name="OSSAccessKeyId"
44CF9590006BF252F707
--9431149156168
Content-Disposition: form-data; name="policy"
eyJleHBpcmF0aW9uIjoiMjAxMy0xMi0wMVQxMjowMDowMFoiLCJjb25kaXRp
b25zIjpbWyJjb250ZW50LWxlbmd0aC1yYW5nZSIsIDAsIDEwNDg1NzYwXSx7
ImJ1Y2tldCI6ImFoYWhIn0sIHsiQSI6ICJhIn0seyJrZXkiOiAiQUJDIn1dfQ==
--9431149156168
Content-Disposition: form-data; name="Signature"
kZoYNv66bsmc10+dcGKw5x2PRrk=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.
txt"
Content-Type: text/plain
abcdefg
--9431149156168
Content-Disposition: form-data; name="submit"
Upload to OSS
--9431149156168--
```

- Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 61d2042d-1b68-6708-5906-33d81921362e
Date: Fri, 24 Feb 2014 06:03:28 GMT
ETag: 5B3C1A2E053D763E1B002CC607C5A0FE
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Post Policy

The policy form field requested by POST is used to verify the validity of the request. The policy is a JSON text encoded with UTF-8 and Base64. It states the conditions that a Post Object request must meet. Although the post form field is optional for uploading public-read-write buckets, we strongly suggest using this field to limit POST requests.

Policy example

```
{ "expiration": "2014-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "johnsmith" },
    ["starts-with", "$key", "user/eric/"]
  ]
}
```

In the Post Object request, the policy must contain expiration and conditions.

Expiration

Expiration specifies the expiration time of the policy, which is expressed in ISO8601 GMT. For example, "2014-12-01T12:00:00.000Z" means that the Post Object request must be sent before 12:00 on December 1, 2014.

Conditions

Conditions is a list that specifies the valid values of form fields in the Post Object request. Note: The value of a form field is extended after the OSS checks the policy. Therefore, the valid value of the form field set in the policy is equivalent to the value of the form field before extension. The following table lists the conditions supported by the policy:

Name	Description
`content-length-range`	Specify the acceptable maximum and minimum sizes of the uploaded file. This condition supports the content-length-range match mode.
`Cache-Control`, `Content-Type`, `Content-Disposition`, `Content-Encoding`, `Expires`	HTTP request headers. This condition supports the exact match and starts-with match modes.
`key`	Specify the object name of the uploaded file. This condition supports the exact match and starts-with match modes.
`success_action_redirect`	Specify the URL to which the client is redirected after successful upload. This condition supports the exact match and starts-with match modes.
`success_action_status`	Specify the status code returned after successful upload if success_action_redirect is not specified. This condition supports the exact match and starts-with match modes.
`x-oss-meta-*`	Specify the user meta set by the user. This condition supports the exact match and starts-with match modes.

If the Post Object request contains other form fields, these extra form fields can be added to Conditions of the policy. The OSS does not check validity of the form fields that are not contained in the conditions.

Condition match modes

Condition match modes	Description
Exact match	The value of a form field must be exactly the same as the value declared in the conditions. For example, if the value of the key form field must be a, the conditions must be: {"key": "a"}, or: ["eq", "$key", "a"]
Starts With	The value of a form field must start with the specified value. For example, if the value of key must start with /user/user1, the conditions must be: ["starts-with", "$key", "/user/user1"]
Specified file size	Specify the maximum and minimum sizes of the files that can be uploaded. For example , if the acceptable file size is 1–10 bytes, the conditions must be: ["content-length-range", 1, 10]

Escape characters

In the policy form field of the Post Object request, $ is used to indicate a variable. Therefore, to describe $, the escape character must be used. In addition, some characters in JSON strings are escaped. The following chart describes characters in the JSON string of the policy form field of a Post Object request.

Escape characters	Description
\/	Slash
\	Backslash
\"	Double quotation marks
\$	Dollar sign
Space	Space
\f	Form feed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\uxxxx	Unicode character

Post Signature

For a verified Post Object request, the HTML form must contain policy and signature. Policy specifies which values are acceptable in the request. The procedure for computing signature is as follows:

1. Create a UTF-8 encoded policy.

2. Encode the policy with Base64. The encoding result is the value of the policy form field, and this value is used as the string to be signed.

3. Use AccessKeySecret to sign the string. The signing method is the same as the computing method of the signature in the Header, that is, replacing the string to be signed with the policy form field.

Demo sample

Demo of passing parameters from the web form field to the OSS: *Click here*.

7.12 Callback

To perform a callback, you only need to attach the relevant callback parameters to the request sent to OSS.

APIs that currently support callbacks are PutObject, PostObject, and CompleteMultipartUpload.

Construct the callback parameter

The callback parameter is composed of a JSON string encoded in Base64. It is critical that you specify the request callback server URL (callbackUrl) and callback content (callbackBody). Detailed JSON fields are as follows:

Field	Meaning	Required?
callbackUrl	• After a file is uploaded successfully, OSS sends a callback request to this URL. The request method is POST and the body is the content specified for callbackBody. Under normal circumstances, if this URL must respond to "HTTP/1.1 200 OK", the response body must be in the JSON format and the	Yes

Field	Meaning	Required?
	response header Content-Length must be a valid value and not exceeding 3 MB. • This function allows users to set up to 5 URLs, separated by ";". OSS sends requests one by one until the first successful response is returned. • If no URL is configured or the value is null, it is regarded that callback is not configured. • HTTPS addresses are supported. • To make sure that Chinese characters are correctly processed, the callbackUrl must be encoded. For example, `http://example.com/Chinese.php?key=value&Chinese Name=Chinese Value` needs to be encoded into `http://example.com/%E4%B8%AD%E6%96%87.php?key=value&%E4%B8%AD%E6%96%87%E5%90%8D%E7%A7%B0=%E4%B8%AD%E6%96%87%E5%80%BC.`	
callbackHost	• The host header value for initiating callback requests. It is valid only when the callbackUrl is set. • If no callbackHost is set, the URL in callbackUrl is resolved and the host	No

Field	Meaning	Required?
	generated after resolving is entered in callbackHost.	
callbackBody	• The value of the request body when a callback is initiated, for example, key =$(key)&etag=$(etag)& my_var=$(x:my_var). • It supports OSS system variables, custom variables , and constants. The supported system variables are described in the following table. Custom variables are supported by transmission through callback-var in PutObject and CompleteMultipart. In Post Object operations, each variable is transmitted through a form field.	Yes
callbackBodyType	• The Content-Type of the callback requests initiated . It supports application/x-www-form-urlencoded and application/json, and the former is the default value. • If the Content-Type is set to application/x-www-form-urlencoded, the variables in callbackBody are replaced by URL encoded values. If the Content-Type is set to application/json, these variables are replaced according to the JSON format.	No

JSON string examples are as follows:

```
{
"callbackUrl":"121.101.166.30/test.php",
```

```
"callbackHost":"oss-cn-hangzhou.aliyuncs.com",
"callbackBody":"{\"mimeType\":${mimeType},\"size\":${size}}",
"callbackBodyType":"application/json"
}
```

```
{
"callbackUrl":"121.43.113.8:23456/index.html",
"callbackBody":"bucket=${bucket}&object=${object}&etag=${etag}&size
=${size}&mimeType=${mimeType}&imageInfo.height=${imageInfo.height}&
imageInfo.width=${imageInfo.width}&imageInfo.format=${imageInfo.format
}&my_var=${x:my_var}"
}
```

Here, the system variables that can be set for callbackBody include the following. In specific, the imageInfo is for the image format. It must be left empty for a non-image format:

System variable	Meaning
bucket	bucket
object	object
etag	The file's etag, that is, the etag field returned to the user.
size	The object size. During the CompleteMu ltipartUpload operation, this is the size of the whole object.
mimeType	The resource type. For jpeg images, the resource type is image/jpeg
imageInfo.height	The image height
imageInfo.width	The image width
imageInfo.format	The image format, such as jpg and png

Custom parameters

You can use the callback-var parameter to configure custom parameters.

Custom parameters are a map of key-values. You can configure the required parameters to the map. When initiating a POST callback request, OSS puts these parameters and the system parameters described in the preceding section in the body of the POST request, so that these parameters can be easily obtained by the callback recipient.

You can construct custom parameters in the same way as constructing the callback parameter. The custom parameters can also be transmitted in the JSON format. The JSON string is a map containing key-values of all custom parameters.

> **Note:**
>
> It must be particularly noted that, the keys of the custom parameters must start with x: and be in the lower case. Otherwise, OSS returns an error.

Assume that you must set two custom parameters x:var1 and x:var2, and the values of the two parameters are value1 and value2 respectively, the JSON format constructed is as follows:

```
{
"x:var1":"value1",
"x:var2":"Value2"
}
```

Construct callback requests

After the callback and callback-var parameters are constructed, you can transmit the parameters to OSS with three methods. The callback parameter is required, and the callback-var parameter is optional. If you configure no custom parameter, the callback-var field does not need to be added. The aforesaid three methods are as follows:

- Including parameters in the URL.

- Including parameters in the header.

- Using form fields to include parameters in the body of a POST request.

  > **Note:**
  >
  > You can only use this method to specify the callback parameter when using POST to upload an object.

The three methods are alternative; otherwise, OSS returns an InvalidArgument error.

To include a parameter in OSS request, first you must use Base64 to encode the preceding constructed JSON string, and include the string in OSS request using the methods described as follows:

- To include parameters in the URL, use 'callback=[CallBack]' or 'callback-var=[CallBackVar]' as a URL parameter to send it with the request. When CanonicalizedResource of the signature is calculated, callback, or callback-var is taken into consideration as a sub-resource.

- To include parameters in the header, use 'x-oss-callback=[CallBack]' or 'x-oss-callback-var=[CallBackVar]' as a head to send it with the request. When CanonicalizedOSSHeaders of

the signature is calculated, x-oss-callback-var and x-oss-callback are taken into consideration.

An example is provided as follows:

```
PUT /test.txt HTTP/1.1
Host: callback-test.oss-test.aliyun-inc.com
Accept-ncoding: identity
Content-Length: 5
x-oss-callback-var: eyJ4Om15X3ZhciI6ImZvci1jYWxsYmFjay10ZXN0In0=
User-Agent: aliyun-sdk-python/0.4.0 (Linux/2.6.32-220.23.2.ali1089.
el5.x86_64/x86_64;2.5.4)
x-oss-callback: eyJjYWxsYmFja1VybCI6IjEyMS40My4xMTMuODoyMzQ1Ni9pbm
RleC5odG1sIiwgICJjYWxsYmFja0JvZHkiOiJidWNrZXQ9JHtidWNrZXR9Jm
9iamVjdD0ke29iamVjdH0mZXRhZz0ke2V0YWd9JnNpemU9JHtzaXplfSZtaW
1lVHlwZT0ke21pbWVUeXBlfSZpbWFnZUluZm8uaGVpZ2h0PSR7aW1hZ2VJbm
ZvLmhlaWdodH0maW1hZ2VJbmZvLndpZHRoPSR7aW1hZ2VJbmZvLndpZHRofS
ZpbWFnZUluZm8uZm9ybWF0PSR7aW1hZ2VJbmZvLmZvcm1hdH0mbXlfdmFyPS
R7eDpteV92YXJ9In0=
Host: callback-test.oss-test.aliyun-inc.com
Expect: 100-Continue
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: text/plain
Authorization: OSS mlepou3zr4u7b14:5a74vhd4UXpmyuudV14Kaen5cY4=
Test
```

- It is slightly complicated to include the callback parameter when POST is used to upload an object, because the callback parameter must be included using an independent form field. See the following example:

```
--9431149156168
Content-Disposition: form-data; name="callback"
eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5w
aHAiLCJjYWxsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tC
b2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkmadGFibGU9JHt4OnRhYmxlfSIs
ImNhbGxiYWNrQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94L3d3dy1mb3JtLXVy
bGVuY29kZWQifQ==
```

If custom parameters are used, you cannot directly include the callback-var parameter in the form field. Each custom parameter must be included using an independent form field. For example, if the JSON of a custom parameter is:

```
{
"x:var1":"value1",
"x:var2":"value2"
}
```

The form field of the POST request are as follows:

```
--9431149156168
Content-Disposition: form-data; name="callback"
eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5w
aHAiLCJjYWxsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tC
b2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkmadGFibGU9JHt4OnRhYmxlfSIs
ImNhbGxiYWNrQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94L3d3dy1mb3JtLXVy
bGVuY29kZWQifQ==
```

```
--9431149156168
Content-Disposition: form-data; name="x:var1"
value1
--9431149156168
Content-Disposition: form-data; name="x:var2"
value2
```

At the same time, you can add callback conditions in the policy (if callback is not added, upload

verification is not performed on this parameter). For example:

```
{ "expiration": "2014-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "johnsmith" },
    {"callback": "eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYW
xsYmFjay5waHAiLCJjYWxsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2
FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkiLCJjYWxsYmFja0
JvZHlUeXBlIjoiYXBwbGljYXRpb24veC13d3ctZm9ybS11cmxlbmNvZGVkIn0="},
    ["starts-with", "$key", "user/eric/"],
  ]
}
```

Initiate callback requests

If the file is uploaded successfully, OSS uses the POST method to send the specific content to the

application server based on the callback parameter and the custom parameters (the callback-var

parameter) in the user's request.

```
POST /index.html HTTP/1.0
Host: 121.43.113.8
Connection: close
Content-Length: 0
Content-Type: application/x-www-form-urlencoded
User-Agent: ehttp-client/0.0.1
bucket=callback-test&object=test.txt&etag=D8E8FCA2DC0F896FD7CB
4CB0031BA249&size=5&mimeType=text%2Fplain&imageInfo.height=&imageInfo.
width=&imageInfo.format=&x:var1=for-callback-test
```

Return callback results

For example, the application server returns the following request for response:

```
HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: application/json
Content-Length: 9
{"a":"b"}
```

Return upload results

The following content is sent to the client:

```
HTTP/1.1 200 OK
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: application/json
```

```
Content-Length: 9
Connection: keep-alive
ETag: "D8E8FCA2DC0F896FD7CB4CB0031BA249"
Server: AliyunOSS
x-oss-bucket-version: 1442231779
x-oss-request-id: 55F6BF87207FB30F2640C548
{"a":"b"}
```

It must be noted that, in the case of requests such as CompleteMultipartUpload, the returned request body includes content (for example, information in XMl format). After using the upload callback function, the original body content is overwritten, such as '"a":"b"'. Take this into consideration for judgment and processing.

Callback signature

When the callback parameter is set, OSS sends the POST callback request to the user's application server based on the callbackUrl set by the user. After receiving the callback request, if you expect the application server to check whether the callback request is initiated by OSS, you can include a signature in the callback request to verify the OSS identity.

• Generate signatures

The signature occurs at the OSS side, and is signed using the RSA Asymmetric Encryption. You can encrypt the signature using a private key as follows:

```
authorization = base64_encode(rsa_sign(private_key, url_decode(path)
 + query_string + '\n' + body, md5))
```

Instructions: The private_key indicates a private key which is only known to OSS. The path indicates the resource path of the callback request. The query_string indicates a query string . The body indicates the message body of the callback. The signature thus consists of the following steps:

— Obtain the string to be signed: The resource path URL is decoded, added by the initial query string, a carriage return and the callback message body.

— RSA signature: Use a private key to sign the expected string. The hashing function for signature is MD5.

— Use Base64 to encode the signed result to get the final signature. Put the signature in the authorization header of the callback request.

An example is provided as follows:

```
POST /index.php? id=1&index=2 HTTP/1.0
Host: 121.43.113.8
Connection: close
Content-Length: 18
```

```
authorization: kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgb
WEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==
Content-Type: application/x-www-form-urlencoded
User-Agent: ehttp-client/0.0.1
x-oss-pub-key-url: aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9j
YWxsYmFja19wdWJfa2V5X3YxLnBlbQ==
bucket=yonghu-test
```

The path is `/index.php`, query_string is `? id=1&index=2`, the body is `bucket=yonghu` `-test`, and the final signature result is `kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/` `kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==`.

- Verify signature

  Signature verification is an inverse process of signature. The signature is verified by the application server, and the process is as follows:

  ```
  Result = rsa_verify(public_key, md5(url_decode(path) + query_string
   + '\n' + body), base64_decode(authorization))
  ```

  The fields have the same meanings as described during the signature process. The public_key indicates a public key. The authorization indicates the signature in the callback header. The signature verification consists of the following steps:

  1. The x-oss-pub-key-url header of the callback request stores the Base64-encoded URL of the public key. The header must be decoded with Base64 to obtain the public key as follows:

     ```
     public_key = urlopen(base64_decode(x-oss-pub-key-url header))
     ```

     It must be noted that, the value of the `x-oss-pub-key-url` header must start with `http` `://gosspublic.alicdn.com/` or `https://gosspublic.alicdn.com/`, so as to make sure that the public key is provided by OSS.

  2. Obtain the Base64-decoded signature

     ```
     signature = base64_decode(Value of the authorization header)
     ```

  3. Obtain the string to be signed the same way as described in the signature process.

     ```
     sign_str = url_decode(path) + query_string + '\n' + body
     ```

  4. Verify the signature

     ```
     result = rsa_verify(public_key, md5(sign_str), signature)
     ```

  The preceding sample is used as an example:

1. Obtain the URL of the public key, that is, with Base64 decoding the `aHR0cDovL2 dvc3NwdWJsaWMuYWxpY2RuLmNvbS9jYWxsYmFja19wdWJfa2V5X3YxLnBlbQ==` to `http://gosspublic.alicdn.com/callback_pub_key_v1.pem`.

2. The signature header `kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgb WEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==` is decoded with Base64 (The decoded result cannot be displayed because it is a nonprintable string).

3. Obtain the string to be singed, that is, url_decode("index.php") + "?id=1&index=2" + "\n" + " bucket=yonghu-test" . Then perform the MD5 check.

4. Verify the signature

- Application server example

    Python is used as an example to demonstrate how an application server verifies a signature. In this example, the M2Crypto library must be installed.

```
import httplib
import base64
import md5
import urllib2
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
from M2Crypto import RSA
from M2Crypto import BIO
def get_local_ip():
    try:
        csock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        csock.connect(('8.8.8.8', 80))
        (addr, port) = csock.getsockname()
        csock.close()
        return addr
    except socket.error:
        return ""
class MyHTTPRequestHandler(BaseHTTPRequestHandler):

    def log_message(self, format, *args):
        return

    def do_POST(self):
        #get public key
        pub_key_url = ''
        try:
            pub_key_url_base64 = self.headers['x-oss-pub-key-url']
            pub_key_url = pub_key_url_base64.decode('base64')
            if not pub_key_url.startswith("http://gosspublic.alicdn.
com/") and not pub_key_url.startswith("https://gosspublic.alicdn.com
/"):
                self.send_response(400)
                self.end_headers()
                return
            url_reader = urllib2.urlopen(pub_key_url)
            #you can cache it
            pub_key = url_reader.read()
        except:
            print 'pub_key_url : ' + pub_key_url
```

```
            print 'Get pub key failed!'
            self.send_response(400)
            self.end_headers()
            return
        #get authorization
        authorization_base64 = self.headers['authorization']
        authorization = authorization_base64.decode('base64')
        #get callback body
        content_length = self.headers['content-length']
        callback_body = self.rfile.read(int(content_length))
        #compose authorization string
        auth_str = ''
        pos = self.path.find('?')
        if -1 == pos:
            auth_str = urllib2.unquote(self.path) + '\n' +
callback_body
        else:
            auth_str = urllib2.unquote(self.path[0:pos]) + self.path
[pos:] + '\n' + callback_body
        print auth_str
        #verify authorization
        auth_md5 = md5.new(auth_str).digest()
        bio = BIO.MemoryBuffer(pub_key)
        rsa_pub = RSA.load_pub_key_bio(bio)
        try:
            result = rsa_pub.verify(auth_md5, authorization, 'md5')
        except:
            result = False
        if not result:
            print 'Authorization verify failed!'
            print 'Public key : %s' % (pub_key)
            print 'Auth string : %s' % (auth_str)
            self.send_response(400)
            self.end_headers()
            return
        #do something accoding to callback_body
        #response to OSS
        resp_body = '{"Status":"OK"}'
        self.send_response(200)
        self.send_header('Content-Type', 'application/json')
        self.send_header('Content-Length', str(len(resp_body)))
        self.end_headers()
        self.wfile.write(resp_body)
class MyHTTPServer(HTTPServer):
    def __init__(self, host, port):
        HTTPServer.__init__(self, (host, port), MyHTTPRequestHandler
)
if '__main__' == __name__:
    server_ip = get_local_ip()
server_port = 23451
server = MyHTTPServer(server_ip, server_port)
server.serve_forever()
```

Application servers implemented in other languages are as follows:

Java version:

— Download address: *click here*.

— Running method: Extract the package and run `java -jar oss-callback-server-demo.jar 9000` (9000 is the port number and can be designated as needed)

PHP version:

— Download address: *click here*

— Running method: Deploy the program to an Apache environment. The characteristics of the PHP language determine that the environment is depended on to retrieve some headers. You may see the example to make modifications to your own environment.

Python version:

— Download address: *click here*

— Running method: Extract the package and directly run `python callback_app_server.py`. You must install RSA dependencies to run this program.

C # version:

— Download address: *click here*

— Running method: Extract the package and see `README.md`.

.NET version:

— Download address: *click here*

— Running method: Extract the package and see `README.md`.

Go version:

— Download address: *click here*

— Running method: Extract the package and see `README.md`.

Ruby version:

— Download address: *click here*

— Running method: ruby aliyun_oss_callback_server.rb

Special instructions

• If the input callback parameter or callback-var parameter is invalid, a 400 error is returned, with the error code of "InvalidArgument". Invalid situations include the following:

— In the PutObject() and CompleteMultipartUpload() interfaces, the callback(x-oss-callback) or callback-var(x-oss-callback-var) parameters are input at the same time to the URL and header fields.

- The callback or callback-var parameter is too long (over 5KB). PostObject() is not subject to this restriction because callback-var parameter is not used, and this is true for the following as well.

- Callback or callback-var is not Base64 encoded.

- After Base64 decoding, the callback or callback-var parameter is not in a valid JSON format.

- After callback parameter resolution, the callbackUrl field contains more than 5 URLs, or the input port in the URL is invalid, such as `{"callbackUrl":"10.101.166.30:test", "callbackBody":"test"}`

- After callback parameter resolution, the callbackBody field is blank.

- After callback parameter resolution, the callbackBodyType field value is not "application/x-www-form-urlencoded" or "application/json".

- After callback parameter resolution, the callbackBody field contains invalid formats of variables. The valid format is `${var}`

- After callback-var parameter resolution, the format is not the expected JSON format. The expected format is:`{"x:var1":"value1","x:var2":"value2"...}`

• If a callback fails, the system returns a 203 error, with the error code "CallbackFailed". A callback failure only indicates that OSS did not receive the expected callback response (for example, the response from the application server was not in the JSON format), not that the application server did not receive the callback request. In addition, by this time, the file has been successfully uploaded to OSS.

• The response returned by the application server to OSS must contain the Content-Length header, and the size of the body cannot exceed 1 MB.

Regions used in Callback

Currently, callback only supports the following regions: China North 2 (Beijing), China East 1 (Hangzhou), China North 1 (Qingdao), China East 2 (Shanghai), Shanghai Financial Cloud, China South 1 (Shenzhen), Hong Kong, China North 5 (huhehaote), China North 3 (zhangjiakou), Middle East 1 (Dubai), Asia Pacific NE 1 (Tokyo), EU Central 1 (Frankfurt), Asia Pacific SE 1 (Singapore), US East 1 (Virginia), US West 1 (Silicon Valley), Asia Pacific SE 2 (Sydney) and Asia Pacific SE 3 (Kuala Lumpur).

7.13 PutSymlink

`PutSymlink` is used to create a symbolic link pointing to the TargetObject on OSS. Users can use the symbolic link to access the TargetObject.

Request syntax

```
PUT /ObjectName? symlink HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
x-oss-symlink-target: TargetObjectName
```

Request header

Name	Type	Description
`x-oss-symlink-target`	String	Indicates the target file that a symbolic link directs to. Valid value: the naming rules are the same as that of objects .

Detail analysis

- As with ObjectName, TargetObjectName must be URL-encoded.

- The target file type of a symbolic link cannot be the symbolic link.

- When creating a symbolic link,

  — The interface does not check whether the target file exists.

  — The interface does not check whether the target file type is valid.

  — The interface does not check access to the target file.

  The foregoing checks are deferred until GetObject must access the API of the target file.

- If a file to be added already exists and you have the file access permission, the newly-added file overwrites the existing file, and the system returns 200 OK.

- If the PutSymlink request carries a parameter prefixed with x-oss-meta-, the parameter is treated as user meta, for example, x-oss-meta-location. A single object can have multiple similar parameters, but the total size of all user meta cannot exceed 8 KB.

- If the bucket type is Archive, you cannot call this interface; otherwise, the system returns Error 400 with the error code "OperationNotSupported".

Example

Request example:

```
PUT /link-to-oss.jpg? symlink HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Cache-control: no-cache
Content-Disposition: attachment;filename=oss_download.jpg
Date: Tue, 08 Nov 2016 02:00:25 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2PR
rk=
x-oss-symlink-target: oss.jpg
```

Return example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 08 Nov 2016 02:00:25 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 582131B9109F4EE66CDE56A5
ETag: "0A477B89B4602AA8DECB8E19BFD447B6"
```

7.14 GetSymlink

The `GetSymlink` operation is used to obtain a symbolic link which you must have the permission

to read.

Request syntax

```
GET /ObjectName? symlink HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response header

Name	Type	Description
`x-oss-symlink-target`	String	The target file that the symbolic link points to.

Detail analysis

If the symbolic link does not exist, the system returns the 404 Not Found error. Error code:

NoSuchKey.

Example

Request example:

```
GET /link-to-oss.jpg? symlink HTTP/1.1
```

```
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:38:30 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJCZkcde6O
hZ9Jfe8=
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 24 Feb 2012 06:38:30 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5650BD72207FB30443962F9A
x-oss-symlink-target: oss.jpg
ETag: "A797938C31D59EDD08D86188F6D5B872"
```

7.15 Restore Object

`RestoreObject` interface is used to have the server restore the object.

To read an object of the Archive type, perform the Restore operation to have the server restore the object. If the type of an object is Standard or IA, do not call the Restore interface.

The status change process of the archived object before and after the Restore operation is shown as follows:

1. An object of the Archive type is archived at first.

2. After a Restore request is initiated, the server starts to restore the object.

3. After the server completes restoration, the object is restored and you can read the object.

4. The restoration status lasts for one day by default and can be prolonged to seven days at most . After the status is expired, the object is archived again.

Performing the Restore operation on an archived object incurs data retrieval costs. Initiating another Restore request for an archived object that is being restored or already restored does not incur any further data retrieval costs.

Request syntax

```
POST /ObjectName? restore HTTP/1.1
Host: archive-bucket.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

• If the Restore interface is called for the object for the first time, the system returns 202.

- If the restore interface has been called and the restoration is already in progress, when the interface is called for the second time, the system returns 409 with the error code "RestoreAlr eadyInProgress", which means that the server is performing the Restore operation and you only have to wait for completion for at most four hours.

- If the Restore interface has been called and the restoration is completed, when the interface is called again, the system returns 200 and the time available for download is prolonged for one day (seven days at most).

- If the object does not exist, the system returns 404.

- If the Restore request is initiated for an object not of the Archive type, the system returns Error 400 with the error code "OperationNotSupported".

Example

Example of the Restore request initiated for the first time

```
POST /oss.jpg? restore HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Sat, 15 Apr 2017 07:45:28 GMT
Authorization: OSS e1Unnbm1rgdnpI:y4eyu+4yje5ioRCr5PB=
```

Return example

```
HTTP/1.1 202 Accepted
Date: Sat, 15 Apr 2017 07:45:28 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D74
```

Example of a request called again when restoration is not completed

```
POST /oss.jpg? restore HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Sat, 15 Apr 2017 07:45:29 GMT
Authorization: OSS e1Unnbm1rgdnpI:21qtGJ+ykDVmdy4eyu+NIUs=
```

Return example

```
HTTP/1.1 409 Conflict
Date: Sat, 15 Apr 2017 07:45:29 GMT
Content-Length: 556
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D74
<? xml version="1.0" encoding="UTF-8"? >
<Error>
  <Code>RestoreAlreadyInProgress</Code>
  <Message>The restore operation is in progress.</Message>
  <RequestId>58EAF141461FB42C2B000008</RequestId>
  <HostId>10.101.200.203</HostId>
```

```
</Error>
```

Example of a request called again when restoration is completed

```
POST /oss.jpg? restore HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Sat, 15 Apr 2017 07:45:29 GMT
Authorization: OSS e1Unnbm1rgdnpI:u6O6FMJnn+WuBwbByZxml+y4eyu+NIUs=
```

Return example

```
HTTP/1.1 200 Ok
Date: Sat, 15 Apr 2017 07:45:30 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D74
```

7.16 SelectObject (in beta phase)

Introduction

Object Storage Service (OSS) built on Alibaba Cloud's Apsara distributed system is a massive, secure, and highly reliable cloud storage solution that offers low cost storage accessible anywhere in the world. OSS possesses excellent scaling abilities for storage capacity and processes, and supports RESTful APIs. Not only can OSS store media files, but it can also be utilized as a data warehouse for massive data file storage. OSS can seamlessly integrate with Hadoop 3.0, and services that are run on EMR (such as Spark/Hive/Presto, MaxCompute, HybridDB and the newly-released Data Lake Analytics) support data processing and retrieval directly from OSS.

However, the current GetObject interface provided by OSS determines that the big data platform can only download all OSS data locally and then for analysis and filter. A lot of bandwidth and client resources are wasted in querying scenarios.

To address this problem, the SelectObject interface is provided. This method allows big data platforms to access OSS to perform basic filtering on data through conditions and Projection, and return useful data only to the big data platform. In this way, the bandwidth and the amount of data processed at the client-side is greatly reduced, making OSS-based data warehousing and data analysis a highly attractive option.

SelectObject is now in beta phase, and provides Java and Python SDKs. SelectObject supports CSV files of RFC 4180 standard to be encoded as UTF-8 (including Class CSV files such as TSV, row and column separators of the file and customizable Quote characters). SelectObject supports

files in standard and low frequency access storage types, and encrypted files, which are fully managed by OSS (or CMK managed by KMS).

The supported SQL syntax is as follows:

- SQL statements: Select From Where

- Data Type: String, Int (64bit), float (64bit), Timestamp, and Boolean

- Operation: Logical condition (AND, OR, NOT), Arithmetic Expression (+-*/%), Comparison operation (>,=, <, >=, <=, ! =), and String operation (LIKE, ||)

The sharding mechanism of SelectObject is similar to the shard download mechanism of GetObject, and includes two sharding methods: sharding by row and sharding by Split. Sharding by row is a common method, but it results in uneven load balancing of sparse data. Sharding by Split is more efficient than sharding by row as a Split contains multiple rows of data, and the data size of each Split is roughly equal, which enables better load balancing performance. Additionally , byte-based sharding (provided by GetObject) may corrupt data. Therefore, sharding by Split is recommended for CSV data.

CSV data in OSS is String type by default. Users can use CAST function to convert data. For example, the following SQL query converts _1 and _2 into Int and compares them.

```
Select * from OSSOBject where cast (_1 as int) > cast(_2 as int)
```

Furthermore, SelectObject supports implicit conversion in WHERE condition, such as the first and the second columns in the following statement will be converted to Int:

```
Select _1 from ossobject where _1 + _2 > 100
```

Description of RESTful API

Execute the SQL statement on the target CSV files and the execution results will be returned. At the same time, the command will automatically save the metadata information of the CSV files, such as the total number of rows and columns.

The API returns a 206 response when the SQL statement is executed correctly. If the SQL statement is incorrect or does not match the CSV files, a 400 error response will be returned.

Request syntax

```
POST /object? x-oss-process=csv/select HTTP/1.1
HOST: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: time GMT
Content-Length: ContentLength
Content-MD5: MD5Value
Authorization: Signature
```

```
<? xml version="1.0" encoding="UTF-8"? >
<SelectRequest>
 Base64 encode (select * From ossobject where)
 <InputSerialization>
  <CompressionType>None</CompressionType>
  <CSV>
   <FileHeaderInfo>NONE|IGNORE|USE</FileHeaderInfo>
   <RecordDelimiter>base64 encode</RecordDelimiter>
   <FieldDelimiter>base64 encode</FieldDelimiter>
   <QuoteCharacter>base64 encode</QuoteCharacter>
   <CommentCharacter>base64 encode</CommentCharacter>
   <Range>line-range=start-end|split-range=start-end</Range>
  </CSV>
  </InputSerialization>
  <OutputSerialization>
        <CSV>
   <RecordDelimiter>base64 encode</RecordDelimiter>
   <FieldDelimiter>base64 encode</FieldDelimiter>
   <KeepAllColumns>false|true</KeepAllColumns>
  </CSV>
 <OutputRawData>false|true</OutputRawData>
  </OutputSerialization>
</SelectRequest>
```

Name	Type	Description
SelectRequest	Container	The container for storing Select requests Child node: Expression, InputSerialization, OutputSerialization Parent node: None
Expression	String	The SQL statement encoded in Base64 Child nodes: None Parent node: SelectRequest
InputSerialization	Container	Input serialized parameters (optional) Child node: CompressionType, CSV Parent node: SelectRequest
OutputSerialization	Container	Output serialized parameters (optional) Child node: CSV, OutputRawData Parent node: SelectRequest
CSV(InputSerialization)	Container	Input CSV-formatted parameters (optional) Child node: FileHeaderInfo, RecordDelimiter, FieldDelimiter, QuoteCharacter, CommentCharacter, Range Parent node: InputSerialization
CSV(OutputSerialization)	Container	Output CSV-formatted parameters (optional) Child node: RecordDelimiter, FieldDelimiter, KeepAllColumns Parent node: OutputSerialization

Name	Type	Description
OutputRawData	bool, default: false	Specifies output data as raw data, not Frame-based data (optional) Child node: None Parent node: OutputSerialization
CompressionType	Enumeration	Specifies file compression types. It can only be None as file compression is currently not supported Child node: None Parent node: InputSerialization
FileHeaderInfo	Enumeration	Specifies CSV files header information (optional) Value: • Use: The CSV file contains header information, and the CSV column name can be used as the column name in the Select. • Ignore: The CSV file contains header information, but the CSV column name can not be used as the column name in the Select. • None: The CSV file contains no header information, and the value can be default. Child node: None Parent node: CSV(input)
RecordDelimiter	String	Specifies line breaks for a CSV, encoded in Base64. The default value is $\backslash\mathrm{n}$ (optional). The value before decoding is at most two characters, expressed as an ANSI character. $\backslash$ n used in Java indicates a line break. Child node: None Parent node: CSV (input, output)
FieldDelimiter	String	Specifies the CSV column separator, encoded in Base64. The default is , (optional) The value before decoding must be expressed as an ANSI character , used in Java indicates a comma. Child Node: None Parent node: CSV (input and output)
QuoteCharacter	String	Specifies the quote character of the CSV, encoded in Base64. The default value is $\backslash$″

Name	Type	Description
		(optional). Inside the CSV quotes, the column separator is treated as a normal character. The value before encoding must be expressed as an ANSI character, such as \" in Java indicates quotation marks. Child node: None Parent node: CSV (input)
CommentCharacter	String	Specifies the CSV comment character, encoded in Base464. The default value is # (optional)
Range	String	Specifies the scope of the query file (optional). Two formats are supported: • Query by row: line-range=start-end • Query by Split: split-range=start-end Both start and end are inclusive. The format is the same as the range parameter in range get. Child node: None Parent node: CSV (input)
KeepAllColumns	Bool	Specifies the location in the response result that contains all of the CSV columns (optional, and default value is false). However, only the columns in the select statement contain values, otherwise they are empty. The data of each row in the response result will be sorted in ascending order of CSV columns. Take the following statement as example: `select _5, _1 from ossobject.` If the value of KeepAllColumn is true, with six columns of data in total, the returned data is as follows: Value of 1st column ...Value of 5th column,\n Child node: None Parent node: CSV(output)

Response results

The request results are returned as a Frame. The format of each frame is as follows, where checksum is CRC32:

Frame-Type | Payload Length | Header Checksum | Payload | Payload Checksum

<---4 bytes--><---4 bytes----------><-------4 bytes-------><variable><----4bytes---------->

There are three different frame types, which are as follows:

Name	Frame-Type Value	Payload format	Description
Data Frame	version | 8388609 <--1 byte><--3 bytes>	scanned size | data <-8 bytes----------><---variable-> The scanned size is the size of the scanned data, and the data is the data returned from the query.	Data Frame is used to return the query data and report its current progress at the same time.
Continuous Frame	version | 8388612 <--1 byte><--3 bytes->	scanned size <----8 bytes-->	Continuous Frame is used to report current progress and maintain HTTP connections. If the query does not return data within 5s, a Continuous Frame will be returned.
End Frame	version | 8388613	Offset | total scanned bytes | http status code | error message <--8bytes-><--8bytes --------------><----4 bytes--------><-variable ------> Where offset is the final location offset after scanning and total scanned bytes is the total bytes of all scanned data. http status code is the final processing result. and error message is the error message itself.	The reason it returns the status code is that when the SelectObject is streamed, only the first block is processed when the Response Header is sent. If the first block of data and SQL match, the Status in the Response Header is a 206 response, but if the following data is illegal, the Status in the Header cannot be changed, and the final Status and Error message includes only the End Frame. Therefore, the client

Name	Frame-Type Value	Payload format	Description
			should treat it as the final result.

Example request

```
POST /oss-select/bigcsv_normal.csv? x-oss-process=csv%2Fselect HTTP/1.
1
Date: Fri, 25 May 2018 22:11:39 GMT
Content-Type:
Authorization: OSS LTAIJPXxMLocA0fD:FC/9JRbBGRw4o2QqdaL246Pxuvk=
User-Agent: aliyun-sdk-dotnet/2.8.0.0(windows 16.7/16.7.0.0/x86;4.0.
30319.42000)
Content-Length: 748
Expect: 100-continue
Connection: keep-alive
Host: host name

<? xml version="1.0"? >
<SelectRequest>
 <Expression>c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2JqZWN0IHdoZXJlIF
80ID4gNDU=
 </Expression>
 <InputSerialization>
  <Compression>None</Compression>
  <CSV>
   <FileHeaderInfo>Ignore</FileHeaderInfo>
   <RecordDelimiter>Cg==</RecordDelimiter>
   <FieldDelimiter>LA==</FieldDelimiter>
   <QuoteCharacter>Ig==</QuoteCharacter>
   <Comments>Iw==</Comments>
  </CSV>
 </InputSerialization>
 <OutputSerialization>
  <CSV>
   <RecordDelimiter>Cg==</RecordDelimiter>
   <FieldDelimiter>LA==</FieldDelimiter>
   <QuoteCharacter>Ig==</QuoteCharacter>
   <KeepAllColumns>false</KeepAllColumns>
  </CSV>
  <OutputRawData>false</OutputRawData>
 </OutputSerialization>
</SelectRequest>
```

SQL statement regex

```
SELECT select-list from OSSObject where_opt limit_opt
```

The keywords SELECT, OSSOBJECT and WHERE cannot be changed.

```
select_list: column name

| column index (for example, _1, _2)

| function(column index | column name)
```

```
| select_list AS alias
```

The supported functions are AVG, SUM, MAX, MIN, COUNT, and CAST (type conversion function). Only * can be used after COUNT.

```
Where_opt:
| WHERE expr
expr:
| literal value
| column name
| column index
| expr op expr
| expr OR expr
| expr AND expr
| expr IS NULL
| expr IS NOT NULL
| expr IN (value1, value2,….)
| expr NOT in (value1, value2,…)
| expr between value1 and value2
| NOT (expr)
| expr op expr
| (expr)
| cast (column index or column name or literal as INT|DOUBLE|DATETIME)
```

op: includes > < >= <= ! = =, LIKE , +-*/%, and connection string ||.

cast: Cast can only be one type for the same column.

```
limit_opt:
```

| limit Integer

Mixing of aggregations and limit

```
Select avg(cast(_1 as int)) from ossobject limit 100
```

In the preceding statement, the AVG value of the first column in the first 100 rows is calculated . This statement is different from what MYSQL outputs, as aggregation in SelectObject always returns only one row of data, so there is no need to limit its output volumes. Therefore, the limit in SelectObject will be executed before the Aggregate function.

SQL statement restrictions are as follows:

- Only UTF-8 encoded text files are supported. Uncompressed GZIP text files can be processed . Support for processing compressed files is coming soon.

- Only single file queries are supported, not JOIN, ORDER BY, GROUP BY, and HAVING

- Not contains aggregation conditions in WHERE statement. For example, where max(cast(age as int)) > 100 is not allowed.

- Up to 1000 columns are supported and the maximum column name is 1024.

- Up to 5% wildcards are supported in the LIKE statement. * and % are equivalent, representing 0 or multiple arbitrary characters.

- Up to 1024 constant items are supported in the IN statement.

- The Projection after Select can be a column name, a column index (_1, _2, etc.), an aggregate function, or a CAST function. Other expressions are not supported Like select _ 1 + _2 from ossobject is not allowed.

- The length of maximum row and column are both 256 KB.

CreateSelectObjectMeta

CreateSelectObjectMeta API is used to obtain information about the target CSV file, such as the total number of rows, the total number of columns, and the number of Splits. If the information does not exist in the file, the whole CSV file is scanned for the preceding information. If the API executes correctly, a 200 response is returned. If the target CSV file is illegal or the specified delimiter does not match the target CSV file, a 400 error response is returned.

Request elements

Name	Type	Description
CsvMetaRequest	Container	Saves the container that created Select Meta requests Child node: Expression, InputSerialization, OutputSerialization Parent node: None
InputSerialization	Container	Inputs serialized parameters (optional) Child node: CompressionType, CSV Parent node: CsvMetaRequest
OverwriteIfExists	Bool	Recalculates SelectMeta to overwrite existing data (optional, the default value is false. If Select Meta already exists, then Select Meta is returned.) Child node: None Parent node: CsvMetaRequest
CompressionType	Enumeration	Specifies file compression types. It can only be None as file compression is currently not supported. Child node: None Parent node: InputSerialization
RecordDelimiter	String	Specifies line breaks for a CSV, encoded in Base64. The default value is \n (optional).

Name	Type	Description
		The value before decoding is at most two characters, expressed as an ANSI character. \n used in Java indicates a line break. Child node: None Parent node: CSV
FieldDelimiter	String	Specifies the CSV column separator, encoded in Base64. The default value is ，(optional). The value before decoding must be expressed as an ANSI character. ，used in Java indicates a comma. Child node: None Parent node: CSV (input and output)
QuoteCharacter	String	Specifies the CSV quote character, encoded in Base64. The default value is \" (optional). Line breaks in quotation marks in CSV, column separators will be treated as normal characters. The value before decoding must be expressed as an ANSI character. \" used in Java indicates a comma. Child node: None Parent node: CSV (input)
CSV	Container	Specifies CSV input format Child node: RecordDelimiter, FieldDelimiter, QuoteCharacter Parent node: InputSerialization

Response Body: empty

Response Header：

- x-oss-select-csv-lines: total number of rows

- x-oss-select-csv-columns: total number of columns

- x-oss-select-csv-splits: total number of Splits

- content-length: file content length

> **Note:**
>
> X-OSS-select-CSV-columns refers to the number of columns in the first row, assuming that the data in the first row is correct.

Example request

```
POST /oss-select/bigcsv_normal.csv? x-oss-process=csv%2Fmeta HTTP/1.1
Date: Fri, 25 May 2018 23:06:41 GMT
Content-Type:
Authorization: OSS LTAIJPXxMLocA0fD:2WF2l6zozf+hzTj9OSXPDklQCvE=
User-Agent: aliyun-sdk-dotnet/2.8.0.0(windows 16.7/16.7.0.0/x86;4.0.
30319.42000)
Content-Length: 309
Expect: 100-continue
Connection: keep-alive
Host: Host

<? xml version="1.0"? >
<CsvMetaRequest>
 <InputSerialization>
  <CSV>
   <RecordDelimiter>Cg==</RecordDelimiter>
   <FieldDelimiter>LA==</FieldDelimiter>
   <QuoteCharacter>Ig==</QuoteCharacter>
  </CSV>
 </InputSerialization>
 <OverwriteIfExisting>false</OverwriteIfExisting>
</CsvMetaRequest>
```

Response code

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 25 May 2018 23:06:42 GMT
Content-Type: application/vnd.ms-excel
Content-Length: 0
Connection: close
x-oss-request-id: 5B089702461FB4C07B000C75
x-oss-location: oss-cn-hangzhou-a
x-oss-access-id: LTAIJPXxMLocA0fD
x-oss-sign-type: NormalSign
x-oss-object-name: bigcsv_normal.csv
Accept-Ranges: bytes
ETag: "3E1372A912B4BC86E8A51234AEC0CA0C-400"
Last-Modified: Wed, 09 May 2018 00:22:32 GMT
x-oss-object-type: Multipart
x-oss-bucket-storage-type: standard
x-oss-hash-crc64ecma: 741622077104416154
x-oss-storage-class: Standard
**x-oss-select-csv-rows: 54000049**
**x-oss-select-csv-columns: 4**
**x-oss-select-csv-splits: 960**
```

Python SDK

```
import os
import oss2

def select_call_back(consumed_bytes, total_bytes = None):
 print('Consumed Bytes:' + str(consumed_bytes) + '\n')

# First, initialize the information such as AccessKeyId, AccessKeyS
ecret, and Endpoint.
```

```
# Obtain the information through environment variables or replace the
information such as"<yourAccessKeyId>" with the real AccessKeyId, and
so on.
#
# Use Hangzhou region as an example. Endpoint can be:
# http://oss-cn-hangzhou.aliyuncs.com
# https://oss-cn-hangzhou.aliyuncs.com

access_key_id = os.getenv('OSS_TEST_ACCESS_KEY_ID', '<yourAccessKeyId
>')
access_key_secret = os.getenv('OSS_TEST_ACCESS_KEY_SECRET', '<
yourAccessKeySecret>')
bucket_name = os.getenv('OSS_TEST_BUCKET', '<yourBucket>')
endpoint = os.getenv('OSS_TEST_ENDPOINT', '<yourEndpoint>')

# Create a bucket instance, all object-related methods need to be
called through the bucket instance.
bucket = oss2. Bucket(oss2. Auth(access_key_id, access_key_secret),
endpoint, bucket_name)
key = 'python_select.csv'
content = 'Tom Hanks,USA,45\r\n'*1024
filename = 'python_select.csv'
# Upload files
bucket.put_object(key, content)
csv_meta_params = {'CsvHeaderInfo': 'None',
'RecordDelimiter': '\r\n'}
select_csv_params = {'CsvHeaderInfo': 'None',
'RecordDelimiter': '\r\n',
'LineRange': (500, 1000)}

csv_header = bucket.create_select_object_meta(key, csv_meta_params)
print(csv_header.csv_rows)
Print(csv_header.csv _ splits)
result = bucket.select_object(key, "select * from ossobject where _3
 > 44 limit 100000", select_call_back, select_csv_params)
content_got = b''
for chunk in result:
 content_got += chunk
print(content_got)

result = bucket.select_object_to_file(key, filename,
"select * from ossobject where _3 > 44 limit 100000", select_call_back
, select_csv_params)

bucket.delete_object(key)
```

Java SDK

```
package samples;

import com.aliyun.oss.event.ProgressEvent;
import com.aliyun.oss.event.ProgressListener;
import com.aliyun.oss.model.*;
import com.aliyun.oss.OSS;
Import com. aliyun. OSS;

import java.io.BufferedOutputStream;
import java.io.ByteArrayInputStream;
import java.io.FileOutputStream;

/**
```

```
 * Examples of create select object metadata and select object.
 *
 */
public class SelectObjectSample {
    private static String endpoint = "<endpoint, http://oss-cn-
hangzhou.aliyuncs.com>";
    private static String accessKeyId = "<accessKeyId>";
    private static String accessKeySecret = "<accessKeySecret>";
    private static String bucketName = "<bucketName>";
    private static String key = "<objectKey>";

    public static void main(String[] args) throws Exception {
        OSS client = new OSSClientBuilder().build(endpoint, accessKeyI
d, accessKeySecret);
        String content = "name,school,company,age\r\n" +
                "Lora Francis,School A,Staples Inc,27\r\n" +
                "Eleanor Little,School B,\"Conectiv, Inc\",43\r\n" +
                "Rosie Hughes,School C,Western Gas Resources Inc,44\r\
n" +
                "Lawrence Ross,School D,MetLife Inc.,24";

        client.putObject(bucketName, key, new ByteArrayInputStream(
content.getBytes()));

        SelectObjectMetadata selectObjectMetadata = client.createSele
ctObjectMetadata(
                new CreateSelectObjectMetadataRequest(bucketName, key)
                        .withInputSerialization(
                                new InputSerialization().withCsvInp
utFormat(
                                        new CSVFormat().withHeaderInfo
(CSVFormat.Header.Use).withRecordDelimiter("\r\n"))));
        System.out.println(selectObjectMetadata.getCsvObjectMetadata
().getTotalLines());
        System. Out. println (selectobjectmetadata. getcsvobje
ctmetadata (). getshares ());

        SelectObjectRequest selectObjectRequest =
                new SelectObjectRequest(bucketName, key)
                        .withInputSerialization(
                                new InputSerialization().withCsvInp
utFormat(
                                        new CSVFormat().withHeaderInfo
(CSVFormat.Header.Use).withRecordDelimiter("\r\n")))
                        .withOutputSerialization(new OutputSeri
alization().withCsvOutputFormat(new CSVFormat()));
        selectObjectRequest.setExpression("select * from ossobject
where _4 > 40");
        OSSObject ossObject = client.selectObject(selectObjectRequest
);
        // read object content from ossObject
        BufferedOutputStream outputStream = new BufferedOutputStream(
new FileOutputStream("result.data"));
        byte[] buffer = new byte[1024];
        int bytesRead;
        while ((bytesRead = ossObject.getObjectContent().read(buffer
)) != -1) {
            outputStream.write(buffer, 0, bytesRead);
        }
        outputStream.close();
    }
```

```
}
```

Best practices

If you want to perform Shard-Query on a massive file, we recommend that you:

1. Call the Create Select Object Meta API to get the total number of Splits for the file. If the file needs to call the SelectObject API, this API makes asynchronous calls before the query, which reduces scan time.

2. Select the appropriate concurrency n based on client-side resources, and divide the total number of Splits by the concurrency n to get the number of Splits that each shard query should contain.

3. Perform the Shard-Query in a form of split-range=1-20 in request body.

4. Merge the results if required.

Use SelectObject with Normal type files. Files of Multipart and Appendable types are not recommended due to poor performance caused by differences in their internal structure.

8 Multipart upload operations

8.1 Introduction

In addition to the PUT Object interface, OSS also provides the Multipart. You can apply the Multipart Upload mode for you to upload files. You can apply the Multipart Upload mode in the following scenarios (but not limited to the following):

• Breakpoint upload must be supported.

• The files to be uploaded are larger than 100 MB.

• The network conditions are poor, and the connection with the OSS server is frequently disconnected.

• Before a file is uploaded, the size of the file cannot be determined.

8.2 InitiateMultipartUpload

Before transmitting data in the Multipart Upload mode, you must call the `InitiateMultipartUpload` interface to notify OSS to initiate a Multipart Upload event.

The `InitiateMultipartUpload` interface returns a globally unique Upload ID created by the OSS server to identify this Multipart Upload event. You can initiate operations based on this ID, such as aborting Multipart Upload and querying Multipart Upload.

Request syntax

```
POST /ObjectName? uploads HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT date
Authorization: SignatureValue
```

Request parameters

During the Initiate Multipart Upload operation, you can use encoding-type to encode the Key in the returned result.

Name	Type	Description
`encoding-type`	String	Specify the encoding type of the Key in the returned result. Currently, the URL encoding is supported. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain

Name	Type	Description
		control characters, such as the characters with ASCII values from 0 to 10. In case, if the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Default: None Optional value: `url`

Request header

Name	Type	Description
`Cache-Control`	String	Specify the web page caching behavior when the object is downloaded. For more information, see *RFC2616*. Default: None
`Content-Disposition`	String	Specify the name of the object when the object is downloaded. For more information, see *RFC2616*. Default: None
`Content-Encoding`	String	Specify the content encoding format when the object is downloaded. For more information, see *RFC2616*. Default: None
`Expires`	Integer	Specify the expiration time in milliseconds. For more information, see *RFC2616*. Default: None
`x-oss-server-side-encryption`	String	Specify the server-side encryption algorithm used to upload each part of this object. OSS stores each uploaded part based on server-side encryption. Legal value: `AES256` or `KMS`

Name	Type	Description
		Note: You must enable the KMS (Key Management Service) on the console to use the KMS encryption algorithm. Otherwise, a KmsServiceNotenabled error code is reported.

Response elements

Name	Type	Description
Bucket	String	Name of a bucket for which a Multipart Upload event is initiated. Parent node: InitiateMu ltipartUploadResult
InitiateMultipartUpl oadResult	Container	The container that saves the result of the Initiate Multipart Upload request. Child Nodes: Bucket, Key, UploadId Parent node: None
Key	String	Name of an object for which a Multipart Upload event is initiated. Parent node: InitiateMu ltipartUploadResult
UploadId	String	Unique ID of a Multipart Upload event. Parent node: InitiateMu ltipartUploadResult
EncodingType	String	Specify the encoding type for the returned results. If encoding-type is specified in the request, the Key is encoded in the returned result. Parent node: Container

Detail analysis

- When using this operation to calculate the authentication signature, you must add "?uploads" to "CanonicalizedResource".

- The Initiate Multipart Upload request supports the following standard HTTP request headers : Cache-Control, Content-Disposition, Content-Encoding, Content-Type, Expires, and custom headers starting with x-oss-meta-. For the specific meaning of these headers, see the PUT Object interface.

- The Initiate Multipart Upload request does not affect an existing object with the same name.

- When receiving an Initiate Multipart Upload request, the server returns a request body in XML format. The request body has three elements: Bucket, Key, and UploadID. You must record the UploadID for subsequent Multipart operations.

- If the x-oss-server-side-encryption header is set in the Initiate Multipart Upload request, the server returns this header in the response header. During the upload of each part, the server automatically stores the part based on entropy encryption. Currently, the OSS server only supports the 256-bit advanced encryption standard (AES256). If values of other standards are specified, the OSS server returns the error code 400 and the error message InvalidEncryptionAlgorithmError. When uploading each part, you do not need to add the x-oss-server-side-encryption request header. If this request header is specified, OSS returns the error code 400 and the error message InvalidArgument.

Example

Request example:

```
POST /multipart.data? uploads HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:/cluRFtRwMTZpC2hTj4F6
7AGdM4=
```

Response example:

```
HTTP/1.1 200 OK
Content-Length: 230
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 42c25703-7503-fbd8-670a-bda01eaec618
Date: Wed, 22 Feb 2012 08:32:21 GMT
Content-Type: application/xml
<? xml version="1.0" encoding="UTF-8"? >
<InitiateMultipartUploadResult xmlns="http://doc.oss-cn-hangzhou.
aliyuncs.com">
    <Bucket> multipart_upload</Bucket>
    <Key>multipart.data</Key>
```

```
    <UploadId>0004B9894A22E5B1888A1E29F8236E2D</UploadId>
</InitiateMultipartUploadResult>
```

8.3 UploadPart

After initiating a Multipart Upload event, you can upload data in parts based on the specified object name and Upload ID. Each uploaded part has a part number ranging from 1 to 10,000.

For the same Upload ID, this part number identifies not only this part of data but also the location of this part in the entire file. If you upload new data using the same part number, OSS overwrites the existing data identified by this part number. Except the last part, the minimum size of other parts is 100 KB. The size of the last part is not restricted.

Request syntax

```
PUT /ObjectName? partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
```

Detail analysis

- Before calling the Initiate Multipart Upload interface to upload a part of data, you must call this interface to obtain an Upload ID issued by the OSS server.

- In the Multipart Upload mode, except the last part, all other parts must be larger than 100 KB. However, the Upload Part interface does not immediately verify the size of the uploaded part (because it does not know whether the part is the last one). It verifies the size of the uploaded part only when Multipart Upload is completed.

- OSS puts the MD5 value of the part data received by the server in the ETag header and return it to the user.

- The part number ranges from 1 to 10,000. If the part number exceeds this range, OSS returns the InvalidArgument error code.

- If the x-oss-server-side-encryption request header is specified when the Initiate Multipart Upload interface is called, OSS encrypts the uploaded part and return the x-oss-server-side -encryption header in the Upload Part response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm used for this part.

- To make sure that the data transmitted over the network is free from errors, the user includes Content-MD5 in the request. The OSS calculates the MD5 value for the uploaded data and compares it with the MD5 value uploaded by the user. If they are inconsistent, OSS returns the InvalidDigest error code.

Examples

Request example:

```
PUT /multipart.data? partNumber=1&uploadId=0004B9895DBBB6EC98E36 HTTP/
1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length:6291456
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:J/lICfXEvPmmSW86bBAfMm
UmWjI=
[6291456 bytes data]
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
ETag: 7265F4D211B56873A381D321F586E4A9
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0c21a
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

8.4 UploadPartCopy

`UploadPartCopy` uploads a part by copying data from an existing object.

You can add an x-oss-copy-source header in the Upload Part request to call the Upload Part Copy
interface. When copying a file larger than 1 GB, you must use the Upload Part Copy method. For
the Upload Part Copy operation, the source bucket and the target bucket must be in the same
region. If you want to copy a file that is less than 1 GB by a single operation, you can use the
Copy Object method.

Request syntax

```
PUT /ObjectName? partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
x-oss-copy-source: /SourceBucketName/SourceObjectName
x-oss-copy-source-range:bytes=first-last
```

Request header

Except the common request header, other headers in the Upload Part Copy request are used to
specify the address of the copied source object and copying range.

Name	Type	Description
`x-oss-copy-source`	String	Specifies the copy source address (the requester must

Name	Type	Description
		have the permission to read the source object). Default: None
`x-oss-copy-source-range`	Integer	Copying range of the copied source object. For example, if the range is set to bytes = 0-9, the system transfers byte 0 to byte 9. This request header is not required when the entire source object is copied. Default: None

The following request header is used for the source objects specified by x-oss-copy-source.

Name	Type	Description
`x-oss-copy-source-if-match`	String	If the ETag value of the source object is equal to the ETag value provided by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed error. Default: None
`x-oss-copy-source-if-none-match`	String	If the source object has not been modified since the time specified by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed error. Default: None
`x-oss-copy-source-if-unmodified-since`	String	If the time specified by the received parameter is the same as or later than the modification time of the file, the system transfers the file normally, and returns 200 OK; otherwise, the system returns

Name	Type	Description
		the 412 Precondition Failed error. Default: None
`x-oss-copy-source-if-modified-since`	String	If the source object has been modified since the time specified by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed error. Default: None

Response elements

Name	Type	Description
`x-oss-copy-source-if-match`	String	If the ETag value of the source object is equal to the ETag value provided by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed error. Default: None
`x-oss-copy-source-if-none-match`	String	If the source object has not been modified since the time specified by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed error. Default: None
`x-oss-copy-source-if-unmodified-since`	String	If the time specified by the received parameter is the same as or later than the modification time of the file, the system transfers the file normally, and returns 200 OK; otherwise, the system returns the 412 Precondition Failed error.

Name	Type	Description
		Default: None
`x-oss-copy-source-if-modified-since`	String	If the source object has been modified since the time specified by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed error. Default: None

Detail analysis

- Before calling the InitiateMultipartUpload interface to upload a part of data, you must call this interface to obtain an Upload ID issued by the OSS server.

- In the Multipart Upload mode, besides the last part, all other parts must be larger than 100 KB . However, the Upload Part interface does not immediately verify the size of the uploaded part (because it cannot immediately determine which part is the last one). It verifies the size of the uploaded part only when Multipart Upload is completed.

- If the x-oss-copy-source-range request header is not specified, the entire source object is copied. If the request header is specified, the returned message includes the length of the entire file and the COPY range. For example, if the returned message is Content-Range: bytes 0-9/44, which means that the length of the entire file is 44, and the COPY range is 0 to 9. If the specified range does not conform to the range rules, OSS copies the entire file and does not contain Content-Range in the result.

- If the x-oss-server-side-encryption request header is specified when the InitiateMultipartUpload interface is called, OSS encrypts the uploaded part and return the x-oss-server-side-encryption header in the Upload Part response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm used for this part. For more information, see the InitiateMultipartUpload API.

- This operation cannot be used to copy objects created by Append Object.

- If the bucket type is Archive, you cannot call this interface; otherwise, the system returns Error 400 with the error code "OperationNotSupported".

Example

Request example:

```
PUT /multipart.data? Partnumber = 1 & sealadid = porterhttp/1.1
```

```
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length:6291456
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:J/lICfXEvPmmSW86bBAfMm
UmWjI=
x-oss-copy-source: /oss-example/ src-object
x-oss-copy-source-range:bytes=100-6291756
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0c21a
Date: Thu, 17 Jul 2014 06:27:54 GMT'
<? xml version="1.0" encoding="UTF-8"? >
<CopyPartResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
    <LastModified>2014-07-17T06:27:54.000Z </LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE"</ETag>
</CopyPartResult>
```

8.5 CompleteMultipartUpload

After uploading all data parts, you must call the `CompleteMultipartUpload` API to complete Multipart Upload for the entire file.

During this operation, you must provide the list (including the part number and ETags) of all valid data parts. After receiving the part list you have submitted, OSS verifies the validity of each data part individually. After all the data parts have been verified, OSS combines these parts into a complete object.

Request syntax

```
POST /ObjectName? uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: Signature

<CompleteMultipartUpload>
<Part>
<PartNumber>PartNumber</PartNumber>
<ETag>ETag</ETag>
</Part>

</CompleteMultipartUpload>
```

Request parameters

During the Complete Multipart Upload operation, you can use encoding-type to encode the Key in the returned result.

Name	Type	Description
encoding-type	String	Specify the encoding type of the Key in the returned result. Currently, the URL encoding is supported. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Default: None Optional value: `url`

Request elements

Name	Type	Description
CompleteMultipartUpload	Container	Container used for storing the content of the Complete Multipart Upload request. Sub-node: one or more part elements Parent node: None
ETag	String	ETag value returned by OSS after data parts are successfully uploaded. Parent node: Part
Part	Container	The container that saves uploaded data parts. Sub-nodes: ETag, PartNumber Parent node: InitiateMu ltipartUploadResult
PartNumber	Integer	Number of parts. Parent node: Part

Response elements

Name	Type	Description
Bucket	String	Specify the bucket name. Parent node: CompleteMu ltipartUploadResult
CompleteMultipartUpl oadResult	Container	The container that stores the result of the Complete Multipart Upload request. Sub-nodes: Bucket, Key, ETag , Location Location Parent node: None
ETag	String	The ETag (entity tag) is created when an object is generated and is used to indicate the content of the object. For the objects created based on the Complete Multipart Upload request, the value of ETag is the UUID of the object content. The value of ETag can be used to check whether the content of the object is changed. Parent node: CompleteMu ltipartUploadResult
Location	String	Specify the URL of the newly created object. Parent node: CompleteMu ltipartUploadResult
Key	String	Name of the newly created object. Parent node: CompleteMu ltipartUploadResult
EncodingType	String	Specify the encoding type for the returned results. If encoding-type is specified in the request, the Key is encoded in the returned result. Parent node: Container

Detail analysis

- When receiving a Complete Multipart Upload request, OSS verifies that all parts except the last part are larger than 100 KB and checks each part number and ETag in the part list submitted by the user. Therefore, when uploading data parts, the client must record not only the part number but also the ETag value returned by OSS each time a part is uploaded successfully.

- It takes a few minutes for OSS to process the Complete Multipart Upload request. During this time, if the client is disconnected from OSS, OSS continues to complete the request.

- The part numbers in the part list submitted by a user can be non-consecutive. For example, the first part number is 1 and the second part number is 5.

- After OSS successfully processes the Complete Multipart Upload request, the corresponding Upload ID becomes invalid.

- The same object may have different Upload IDs. When an Upload ID is completed, other Upload IDs of this object are not affected.

- If the x-oss-server-side-encryption request header is specified when the Initiate Multipart Upload interface is called, OSS returns the x-oss-server-side-encryption header in the Complete Multipart Upload response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm used for this object.

- If you have uploaded the Content-MD5 request header, the OSS calculates the body's Content -MD5 and check if the two are consistent. If the two are different, the error code InvalidDigest is returned.

Example

Request example:

```
POST /multipart.data? uploadId=0004B9B2D2F7815C432C9057C03134D4 HTTP/1
.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 1056
Date: Fri, 24 Feb 2012 10:19:18 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:8VwFhFUWmVecK6jQlHlXMK/
zMT0=

<CompleteMultipartUpload>
    <Part>
        <PartNumber>1</PartNumber>
        <ETag>"3349DC700140D7F86A078484278075A9"</ETag>
    </Part>
    <Part>
        <PartNumber>5</PartNumber>
        <ETag>"8EFDA8BE206636A695359836FE0A0E0A"</ETag>
    </Part>
    <Part>
        <PartNumber>8</PartNumber>
```

```
        <ETag>"8C315065167132444177411FDA149B92"</ETag>
    </Part>
</CompleteMultipartUpload>
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Content-Length: 329
Content-Type: Application/xml
Connection: keep-alive
X-OSS-request-ID: 594f0751-3b1e-168f-4501-4ac71d217d6e
Date: Fri, 24 Feb 2012 10:19:18 GMT

<? xml version="1.0" encoding="UTF-8"? >
<CompleteMultipartUploadResult xmlns="http://doc.oss-cn-hangzhou.
aliyuncs.com">
    <Location>http://oss-example.oss-cn-hangzhou.aliyuncs.com /
multipart.data</Location>
    <Bucket>oss-example</Bucket>
    <Key>multipart.data</Key>
    <ETag>B864DB6A936D376F9F8D3ED3BBE540DD-3</ETag>
</CompleteMultipartUploadResult>
```

8.6 AbortMultipartUpload

`AbortMultipartUpload` can be used to stop a Multipart Upload event based on the Upload ID you provide.

When a Multipart Upload event is aborted, you cannot use this Upload ID to perform any operations and the uploaded data parts are deleted.

Request syntax

```
DELETE /ObjectName? uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

Detail analysis

•   When you stop a Multipart Upload event, parts still being uploaded are not deleted. Therefore, if concurrent accesses exist, you must call the AbortMultipartUpload interface several times to completely release the space of OSS.

•   If the entered Upload ID does not exist, OSS returns an error 404 with the error code: NoSuchUpload.

Example

Request example:

```
Delete /multipart.data? &uploadId=0004B9895DBBB6EC98E HTTP/1.1
```

```
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:J/lICfXEvPmmSW86bBAfMm
UmWjI=
```

Response example:

```
HTTP/1.1 204
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 059a22ba-6ba9-daed-5f3a-e48027df344d
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

8.7 ListMultipartUploads

The `ListMultipartUploads` interface can be used to list all Multipart Upload events in
execution, that is, Multipart Upload events that have been initiated but not completed or aborted.

The listing result returned by OSS contains a maximum of 1000 Multipart Upload messages. If
you want to specify the number of Multipart Upload messages in the listing result returned by OSS
, you can add the max-uploads parameter to the request. In addition, the IsTruncated element
 in the listing result returned by OSS indicates whether other Multipart Upload messages are
contained.

Request syntax

```
Get /? uploads HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

Request parameters

Name	Type	Description
delimiter	String	A character used to group object names. All those objects whose names contain the specified prefix and behind which the delimiter occurs for the first time act as a group of elements - CommonPrefixes.
max-uploads	String	Specify the maximum number of multipart upload tasks returned for one request. If this parameter is not specified, the default value 1,000 is used.

Name	Type	Description
		The max-uploads value cannot exceed 1,000.
`key-marker`	String	Used together with the upload-id-marker parameter to specify the starting position of the returned result. • If the upload-id-marker parameter is not set, the query result includes Multipart tasks in which the lexicographic orders of all object names are greater than the value of the key-marker parameter. • If the upload-id-marker parameter is set, the query result includes Multipart tasks in which the lexicographic orders of all object names are greater than the value of the key-marker parameter, and all Multipart Upload tasks in which the object names are the same as the value of the key-marker parameter, but the Upload IDs are greater than the value of the upload-id-marker parameter.
`prefix`	String	Limit that the returned object key must be prefixed accordingly. Note that the keys returned from queries using a prefix still contain the prefix.
`upload-id-marker`	String	Used together with the key-marker parameter to specify the starting position of the returned result.

Name	Type	Description
		• If the key-marker parameter is not set, OSS ignores the upload-id-marker parameter. • If the key-marker parameter is set, the query result includes Multipart tasks in which the lexicographic orders of all object names are greater than the value of the key-marker parameter, and all Multipart Upload tasks in which the object names are the same as the value of the key-marker parameter, but the Upload IDs are greater than the value of the upload-id-marker parameter.
`encoding-type`	String	Specify the encoding of the returned content and the encoding type. Delimiter, KeyMarker, Prefix, NextKeyMarker, and Key use UTF-8 characters, but the XML 1.0 Standard does not support parsing certain control characters, such as characters with ASCII values ranging from 0 to 10. If some elements in the returned results contain control characters that are not supported by the XML 1.0 Standard, encoding-type can be specified to encode these elements, such as Delimiter, KeyMarker, Prefix, NextMarker, and Key. Default: None

Response elements

Name	Type	Description
`ListMultipartUploads Result`	Container	The container that saves the result of the List Multipart Upload request. Sub-nodes: Bucket, KeyMarker , UploadIdMarker, NextKeyMar ker, NextUploadIdMarker, MasUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated , Upload Parent node: None
`Bucket`	String	Bucket name. Parent node: ListMultip artUploadsResult
`EncodingType`	String	Specify the encoding type for the returned results. If encoding-type is specified in the request, those elements including Delimiter, KeyMarker, Prefix, NextKeyMarker, and Key are encoded in the returned result. Parent node: ListMultip artUploadsResult
`KeyMarker`	String	Position of the starting object in the list. Parent node: ListMultip artUploadsResult
`UploadIdMarker`	String	Position of the starting Upload ID in the list. Parent node: ListMultip artUploadsResult
`NextKeyMarker`	String	If not all results are returned this time, the response request includes the NextKeyMarker element to indicate the value of KeyMarker in the next request. Parent node: ListMultip artUploadsResult

Name	Type	Description
NextUploadMarker	String	If not all results are returned this time, the response request includes the NextUploadMarker element to indicate the value of UploadMarker in the next request. Parent node: ListMultip artUploadsResult
MaxUploads	Integer	The maximum upload number returned by the OSS. Parent node: ListMultip artUploadsResult
IsTruncated	enumerative string	Specify whether the returned Multipart Upload result list is truncated. The "true" indicates that not all results are returned; "false" indicates that all results are returned. Valid values: `false`, `true` Default: `false` Parent node: ListMultip artUploadsResult
Upload	Container	The container that saves the information about the Multipart Upload event. Sub-nodes: Key, UploadId, Initiated Parent node:ListMultip artUploadsResult
Key	String	Name of an object for which a Multipart Upload event is initiated. Parent node: Upload
UploadId	String	ID of a Multipart Upload event. Parent node: Upload
Initiated	Date	Time when a Multipart Upload event is initiated. Parent node: Upload

Detail analysis

- The maximum value of the `max-uploads` parameter is 1,000.

- The results returned by OSS are listed in ascending order based on the lexicographic orders of object names; for the same object, the results are listed in ascending time order.

- Using the prefix parameter, you can flexibly manage objects in a bucket in groups (similar to the folder function).

- The List Multipart Uploads request supports five request parameters: prefix, marker, delimiter, upload-id-marker, and max-uploads. Based on the combinations of these parameters, you can set rules for querying Multipart Uploads events to obtain the expected query results.

Example

Request example:

```
Get /? uploads HTTP/1.1
HOST: OSS-example.
Date: Thu, 23 Feb 2012 06:14:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JX75CtQqsmBBz+dcivn7kwBM
vOY=
```

Response example:

```
HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1839
Content-type: application/xml
x-oss-request-id: 58a41847-3d93-1905-20db-ba6f561ce67a
Date: Thu, 23 Feb 2012 06:14:27 GMT

<? xml version="1.0" encoding="UTF-8"? >
<ListMultipartUploadsResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs
.com">
    <Bucket>oss-example</Bucket>
    <KeyMarker></KeyMarker>
    <UploadIdMarker></UploadIdMarker>
    <NextKeyMarker>oss.avi</NextKeyMarker>
    <NextUploadIdMarker>0004B99B8E707874FC2D692FA5D77D3F</NextUpload
IdMarker>
    <Delimiter></Delimiter>
    <Prefix></Prefix>
    <MaxUploads>1000</MaxUploads>
    <IsTruncated>false</IsTruncated>
    <Upload>
        <Key>multipart.data</Key>
        <UploadId>0004B999EF518A1FE585B0C9360DC4C8</UploadId>
        <Initiated>2012-02-23T04:18:23.000Z</Initiated>
    </Upload>
    <Upload>
        <Key>multipart.data</Key>
        <UploadId>0004B999EF5A239BB9138C6227D69F95</UploadId>
        <Initiated>2012-02-23T04:18:23.000Z</Initiated>
```

```
        </Upload>
    <Upload>
        <Key>oss.avi</Key>
        <UploadId>0004B99B8E707874FC2D692FA5D77D3F</UploadId>
        <Initiated>2012-02-23T06:14:27.000Z</Initiated>
    </Upload>
</ListMultipartUploadsResult>
```

8.8 ListParts

`ListParts` can be used to list all successfully uploaded parts mapped to a specific upload ID.

Request syntax

```
Get /ObjectName? uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

Request parameters

Name	Type	Description
`uploadId`	String	ID of a Multipart Upload event. Default: None
`max-parts`	Integer	The maximum part number in the response of the OSS. Default: `1,000`
`part-number-marker`	Integer	Starting position of a specific list. A part is listed only when the part number is greater than the value of this parameter. Default: None
`encoding-type`	String	Specify the encoding of the returned content and the encoding type. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key.

Name	Type	Description
		Default: None Optional value: `url`

Response elements

Name	Type	Description
`ListPartsResult`	Container	The container that saves the result of the List Parts request. Sub-nodes: Bucket, Key, UploadId, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part Parent node: None
`Bucket`	String	Bucket name. Parent node: ListPartsResult
`EncodingType`	String	Specify the encoding type for the returned result. If the encoding type is specified in the request, the Key is encoded in the returned result. Parent node: ListPartsResult
`Key`	String	The object name. Parent node: ListPartsResult
`UploadId`	String	ID of an Upload event. Parent node: ListPartsResult
`PartNumberMarker`	Integer	Starting position of the part numbers in the listing result. Parent node: ListPartsResult
`NextPartNumberMarker`	Integer	If not all results are returned this time, the response request includes the NextPartNumberMarker element to indicate the value of PartNumberMarker in the next request. Parent node: ListPartsResult
`MaxParts`	Integer	The maximum part number in the returned request.

Name	Type	Description
		Parent node: ListPartsResult
IsTruncated	Enumerating strings	Whether the returned result list for List Part is truncated. The "true" indicates that not all results are returned; "false" indicates that all results are returned. Valid values: true, false Parent node: ListPartsResult
Part	String	The container that saves part information. Sub-nodes: PartNumber, LastModified, ETag, Size Parent node: ListPartsResult
PartNumber	Integer	Part number. Parent node: ListPartsResult. Part
LastModified	Date	Time when a part is uploaded. Parent node: ListPartsResult. part
ETag	String	ETag value in the content of the uploaded part. Parent node: ListPartsResult. Part
Size	Integer	Size of the uploaded part. Parent node: ListPartsResult. Part

Detail analysis

- ListParts supports two request parameters: max-parts and part-number-marker.

- The maximum value of the max-parts parameter is 1,000; its default value is also 1,000.

- The results returned by OSS are listed in ascending order based on the part numbers.

- The errors may occur in network transmission, it is not recommended that you use the result (part number and ETag value) of List Parts to generate the final part list of Complete Multipart.

Example

Request example:

```
Get /multipart.data? uploadId=0004B999EF5A239BB9138C6227D69F95 HTTP/1.
1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 23 Feb 2012 07:13:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:4qOnUMc9UQWqkz8wDqD3
lIsa9P8=
```

Response example:

```
HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1221
Content-type: application/xml
x-oss-request-id: 106452c8-10ff-812d-736e-c865294afc1c
Date: Thu, 23 Feb 2012 07:13:28 GMT

<? xml version="1.0" encoding="UTF-8"? >
<ListPartsResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com">
    <Bucket>multipart_upload</Bucket>
    <Key>multipart.data</Key>
    <UploadId>0004B999EF5A239BB9138C6227D69F95</UploadId>
    <NextPartNumberMarker>5</NextPartNumberMarker>
    <MaxParts>1000</MaxParts>
    <IsTruncated>false</IsTruncated>
    <Part>
        <PartNumber>1</PartNumber>
        <LastModified>2012-02-23T07:01:34.000Z</LastModified>
        <ETag>&quot;3349DC700140D7F86A078484278075A9&quot;</ETag>
        <Size>6291456</Size>
    </Part>
    <Part>
        <PartNumber>2</PartNumber>
        <LastModified>2012-02-23T07:01:12.000Z</LastModified>
        <ETag>&quot;3349DC700140D7F86A078484278075A9&quot;</ETag>
        <Size>6291456</Size>
    </Part>
    <Part>
        <PartNumber>5</PartNumber>
        <LastModified>2012-02-23T07:02:03.000Z</LastModified>
        <ETag>&quot;7265F4D211B56873A381D321F586E4A9&quot;</ETag>
        <Size>1024</Size>
    </Part>
</ListPartsResult>
```

9 Cross-Origin Resource Sharing

9.1 Introduction

Cross-Origin Resource Sharing (CORS) allows web applications to access resources in other domains.

With the CORS support, OSS allows users to develop more flexible web applications. OSS provides an interface for developers to easily control various permissions for a cross-domain access.

9.2 PutBucketcors

With the `PutBucketcors` operation, you can set a CORS rule for a specified bucket. If an original rule exists, it is overwritten.

Request syntax

```
PUT /?cors HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
    <CORSRule>
      <AllowedOrigin>the origin you want allow CORS request from</
AllowedOrigin>
      <AllowedOrigin>…</AllowedOrigin>
      <AllowedMethod>HTTP method</AllowedMethod>
      <AllowedMethod>…</AllowedMethod>
        <AllowedHeader> headers that allowed browser to send</
AllowedHeader>
          <AllowedHeader>…</AllowedHeader>
          <ExposeHeader> headers in response that can access from
client app</ExposeHeader>
          <ExposeHeader>…</ExposeHeader>
          <MaxAgeSeconds>time to cache pre-fight response</MaxAgeSeco
nds>
    </CORSRule>
    <CORSRule>
      …
    </CORSRule>
…
```

```
</CORSConfiguration >
```

Request elements

Name	Type	Description	Required
CORSRule	Container	CORS rule container. Each bucket allows up to 10 rules. Parent node: CORSConfiguration	Yes
AllowedOrigin	String	The origins allowed for cross-domain requests. Multiple elements can be used to specify multiple allowed origins. Each rule allows up to one wildcard "*". If "*" is specified, cross-domain requests of all origins are allowed. Parent node: CORSRule	Yes
AllowedMethod	enumeration (GET, PUT, DELETE, POST , HEAD)	Specify the allowed methods for cross-domain requests. Parent node: CORSRule	Yes
AllowedHeader	String	Control whether the headers specified by Access-Control-Request-Headers in the OPTIONS prefetch command are allowed. Each header specified by Access-Control-Request-Headers must match a value in AllowedHeader. Each rule allows up to one wildcard "*" .	No

Name	Type	Description	Required
		Parent node: CORSRule	
`ExposeHeader`	String	Specify the response headers allowing users to access from an application (for example, a Javascript XMLHttpRequest object). The wildcard "*" is not allowed. Parent node: CORSRule	No
`MaxAgeSeconds`	Integer	Specify the cache time for the returned result of a browser prefetch (OPTIONS) request to a specific resource. The unit is seconds. One CORSRule allows not more than one such parameter. Parent node: CORSRule	No
`CORSConfiguration`	Container	CORS rule container of a bucket Parent node: None	Yes

Detail analysis

- CORS is disabled for buckets by default. The origins of all cross-domain requests are forbidden.

- To use CORS in applications, for example, accessing OSS from www.a.com through the XMLHttpRequest function of browser, you must manually upload a CORS rule through this interface to enable CORS. This rule is described in an XML document.

- The CORS setting for each bucket is specified by multiple CORS rules. Each bucket allows a maximum of 10 rules. The uploaded XML document cannot be larger than 16 KB.

- When OSS receives a cross-domain request (or an OPTIONS request), it reads the bucket's CORS rules and then checks the relevant permissions. OSS checks each rule sequentially and

uses the first rule that matches to approve the request and return the corresponding header. If none of the rules match, OSS does not attach any CORS header.

- Successful CORS rule matching must satisfy three conditions. First, the request Origin must match the AllowedOrigin. Second, the request method (for example, GET, PUT) or the method corresponding to the Access-Control-Request-Method header in an OPTIONS request must match the AllowedMethod. Third, each header contained in the Access-Control-Request-Headers in an OPTIONS request must match the AllowedHeader.

- If you have uploaded the Content-MD5 request header, OSS calculates the body's Content-MD5 and check if the two are the same. If the two are different, the error code: InvalidDigest is returned.

Example

Example of adding a bucket CORS rule:

```
PUT /?cors HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3J
xrTZHiA=
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
    <CORSRule>
      <AllowedOrigin>*</AllowedOrigin>
      <AllowedMethod>PUT</AllowedMethod>
      <AllowedMethod>GET</AllowedMethod>
      <AllowedHeader>Authorization</AllowedHeader>
    </CORSRule>
    <CORSRule>
      <AllowedOrigin>http://www.a.com</AllowedOrigin>
      <AllowedOrigin>http://www.b.com</AllowedOrigin>
      <AllowedMethod>GET</AllowedMethod>
      <AllowedHeader> Authorization</AllowedHeader>
      <ExposeHeader>x-oss-test</ExposeHeader>
      <ExposeHeader>x-oss-test1</ExposeHeader>
      <MaxAgeSeconds>100</MaxAgeSeconds>
    </CORSRule>
</CORSConfiguration >
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 50519080C4689A033D00235F
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
```

```
Server: AliyunOSS
```

9.3 GetBucketcors

The `GetBucketcors` operation is used to obtain the current CORS rules of a specified bucket.

Request syntax

```
GET /? cors HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Type	Description
`CORSRule`	Container	CORS rule container. Each bucket allows up to 10 rules. Parent node: CORSConfig uration
`AllowedOrigin`	String	The origins allowed for cross-domain requests. Multiple elements can be used to specify multiple allowed origins. Each rule allows up to one wildcard "*". If "*" is specified, cross-domain requests of all origins are allowed. Parent node: CORSRule
`AllowedMethod`	enumeration (GET, PUT, DELETE, POST, HEAD)	Specify the allowed methods for cross-domain requests. Parent node: CORSRule
`AllowedHeader`	String	Control whether the headers specified by Access-Control-Request-Headers in the OPTIONS prefetch command are allowed. Each header specified by Access-Control-Request-Headers must match a value in AllowedHeader. Each rule allows up to one wildcard "*" Parent node: CORSRule

Name	Type	Description
ExposeHeader	String	Specify the response headers allowing users to access from an application (for example, a Javascript XMLHttpRequest object). The wildcard "*" is not allowed. Parent node: CORSRule
MaxAgeSeconds	Integer	Specify the cache time for the returned result of a browser prefetch (OPTIONS) request to a specific resource. The unit is seconds. One CORSRule allows not more than one such parameter. Parent node: CORSRule
CORSConfiguration	Container	CORS rule container of a bucket Parent node: None

Detail analysis

- If a bucket does not exist, the error "404 no content" is returned. Error code: NoSuchBucket.

- Only the bucket owner can obtain CORS rules. Otherwise, the error 403 Forbidden is returned with the error code: AccessDenied.

- If CORS rules do not exist, OSS returns the "404 Not Found" error with the error code: NoSuchCORSConfiguration.

Example

Request example:

```
GET /? cors HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51
NNHD39zXw=
```

Response example with CORS rules already set:

```
HTTP/1.1 200
x-oss-request-id: 50519080C4689A033D00235F
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
```

```
Server: AliyunOSS

<? xml version="1.0" encoding="UTF-8"? >
<CORSConfiguration>
    <CORSRule>
      <AllowedOrigin>*</AllowedOrigin>
      <AllowedMethod>GET</AllowedMethod>
      <AllowedHeader>*</AllowedHeader>
      <ExposeHeader>x-oss-test</ExposeHeader>
      <MaxAgeSeconds>100</MaxAgeSeconds>
    </CORSRule>
</CORSConfiguration>
```

9.4 DeleteBucketcors

`DeleteBucketcors` is used to turn off the CORS function for the specified bucket and clear all

rules.

Request syntax

```
DELETE /? cors HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

Detail analysis

- If the bucket does not exist, OSS returns the 404 no content error, error code: NoSuchBucket.

- Only the owner of the bucket can delete the CORS rules corresponding to the bucket. If you try

   to operate a bucket which does not belong to you, OSS returns the 403 Forbidden error, error

   code: accessdenied.

Examples

Request example:

```
DELETE /? cors HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:LnM4AZ1OeIduZF5vGFWi
cOMEkVg=
```

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 5051845BC4689A033D0022BC
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: keep-alive
Content-Length: 0
```

```
Server: AliyunOSS
```

9.5 OptionObject

Before sending a cross-domain request, the browser sends a preflight request (OPTIONS)

containing a specific origin, HTTP method, and header information to OSS to determine whether

to send a real request.

OSS can enable CORS for a bucket through the Put Bucket cors interface. After CORS is enabled

, OSS assesses whether to allow the preflight request of the browser based on the specified rules

. If OSS does not allow this request or CORS is disabled, the error 403 Forbidden is returned.

Request syntax

```
OPTIONS /ObjectName HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Origin:Origin
Access-Control-Request-Method:HTTP method
Access-Control-Request-Headers:Request Headers
```

Request header

Name	Type	Description
`Origin`	String	Origin of a request, used to identify a cross-domain request. Default: None
`Access-Control-Request-Method`	String	Methods to be used in an actual request. Default: None
`Access-Control-Request-Headers`	String	Headers, except simple headers, to be used in an actual request. Default: None

Response header

Name	Type	Description
`Access-Control-Allow-Origin`	String	Origin contained in a request. This header is not contained if this request is not allowed.
`Access-Control-Allow-Methods`	String	HTTP method used by a request. This header is not

Name	Type	Description
		contained if this request is not allowed.
`Access-Control-Allow-Headers`	String	Header list carried in a request . If the request contains forbidden headers, this header is not contained and the request is rejected.
`Access-Control-Expose-Headers`	String	Header list that can be accessed by the client's JavaScript application.
`Access-Control-Max-Age`	Integer	Time duration when the browser can buffer the preflight results. The unit is seconds.

Example

Request example:

```
OPTIONS /testobject HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Origin:http://www.example.com
Access-Control-Request-Method:PUT
Access-Control-Request-Headers:x-oss-test
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 5051845BC4689A033D0022BC
Date: Fri, 24 Feb 2012 05:45:34 GMT
Access-Control-Allow-Origin: http://www.example.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-oss-test
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

10 關於**LiveChannel**的操作

10.1 LiveChannel簡介

用戶可以使用RTMP協議將音視頻數據上傳到OSS，轉儲為指定格式的音視頻檔案。上傳前，用戶首先需要建立一個LiveChannel，以獲取對應的推流地址，更詳細的資訊請參考對應的API。

通過RTMP協議上傳音視頻數據目前有以下限制：

- 只能使用RTMP推流的方式，不支援拉流。

- 必須包含視頻流，且視頻流格式為H264。

- 音頻流是可選的，並且只支援AAC格式，其他格式的音頻流會被丟棄。

- 轉儲只支援HLS協議。

- 一個LiveChannel同時只能有一個客戶端向其推流。

10.2 RTMP推流地址及簽名

RTMP推流地址形如：rtmp://your-bucket.oss-cn-hangzhou.aliyuncs.com/live/test-channel

其組成規則為: `rtmp://${bucket}.${host}/live/${channel}?${params}`

- live為RTMP協議的app名稱，OSS固定使用live。

- params為推流的參數，格式與HTTP請求的query string相同，即形如"varA=valueA&varB=valueB"。

- BucketAcl非public-read-write時，推流地址需要簽名才可以使用；簽名方法類似OSS的Url簽名，但有一些細節上的不同，後文會描述具體的規則。

RTMP推流支援的url參數

名稱	描述
`playlistName`	用來指定生成的m3u8檔案名稱，其值覆蓋LiveChannel中的配置。注意：生成的m3u8名稱仍然會被添加"${channel_name}/"首碼。

推流地址的簽名規則

一個帶簽名的推流地址形如：`rtmp://${bucket}.${host}/live/${channel}?`

`OSSAccessKeyId=xxx&Expires=yyy&Signature=zzz&${params}`

參數名稱	描述
OSSAccessKeyId	意義同OSS的HTTP簽名的AccessKeyId

參數名稱	描述
Expires	過期時間戳記，格式採用Unit timestamp
Signature	簽名字元串，後文會描述其計算方法
params	其他參數，所有的參數都需要參與簽名

Signature的計算規則如下：

```
base64(hmac-sha1(AccessKeySecret,
    + Expires + "\n"
    + CanonicalizedParams
    + CanonicalizedResource))
```

名稱	描述
CanonicalizedResource	格式為 "/BucketName/ChannelName"
CanonicalizedParams	按照param key字典序拼接"key:value\n",將所有的參數拼起來，如果參數個數為0，那麼這一項為空。參數中不包含SecurityToken、OSSAccessKeyId和Expire以及Signature。每一個param key只能出現一次。

10.3 PutLiveChannelStatus

LiveChannel有兩種Status：enabled和disabled，用戶可以使用本介面在兩種Status之間進行切換。

處於disabled狀態時，OSS會禁止用戶向該LiveChannel進行推流操作；如果有用戶正在向該LiveChannel推流，那麼推流的客戶端會被強制斷開（可能會有10s左右的延遲）。

請求文法

PUT /ChannelName?live&status=NewStatus HTTP/1.1Date: GMT dateHost: BucketName.oss-cn-hangzhou.aliyuncs.comAuthorization: SignatureValue

請求參數

名稱	描述	是否必需
NewStatus	指定LiveChannel的目標Status。 有效值：enabled、disabled	是

細節分析

- 當沒有客戶端向該LiveChannel推流時，調用PutLiveChannel重新建立LiveChannel也可以達到修改Status的目的。

- 當有客戶端向該LiveChannel推流時，無法調用PutLiveChannel重新建立LiveChannel，只能通過本介面修改LiveChannel的狀態為disabled。

樣本

請求樣本

```
PUT /test-channel?live&status=disabled HTTP/1.1
Date: Thu, 25 Aug 2016 05:37:38 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:X/mBrSbkNoqM/JoAfRC0ytyQ5pY=
```

返回樣本

```
HTTP/1.1 200
content-length: 0
server: AliyunOSS
connection: close
x-oss-request-id: 57BE8422B92475920B002030
date: Thu, 25 Aug 2016 05:37:39 GMT
```

10.4 PutLiveChannel

通過RTMP協議上傳音視頻數據前，必須先調用該介面來建立一個LiveChannel。該介面會返回RTMP推流地址，以及對應的播放地址。

用戶可以使用返回的地址進行推流、播放。另外，用戶可以根據該LiveChannel的名稱來發起相關的操作，如查詢推流狀態、查詢推流記錄、禁止推流等。

請求文法

```
PUT /ChannelName?live HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT date
Content-Length: Size
Authorization: SignatureValue
<?xml version="1.0" encoding="UTF-8"?>
<LiveChannelConfiguration>
  <Description>ChannelDescription</Description>
  <Status>ChannelStatus</Status>
  <Target>
     <Type>HLS</Type>
     <FragDuration>FragDuration</FragDuration>
     <FragCount>FragCount</FragCount>
     <PlayListName>PlaylistName</PlayListName>
  </Target>
  <Snapshot>
    <RoleName>Snapshot ram role</RoleName>
    <DestBucket>Snapshot dest bucket</DestBucket>
```

```
    <NotifyTopic>Notify topic of MNS</NotifyTopic>
    <Interval>Snapshot interval in second</Interval>
  </Snapshot>
</LiveChannelConfiguration>
```

請求元素(Request Elements)

名稱	類型	描述	是否必需
LiveChannelConfiguration	容器	保存LiveChannel配置的容器。 子節點：Description、Status、Target 父節點：無	是
Description	字元串	LiveChannel的描述資訊，最長128位元組。 子節點：無 父節點：LiveChannelConfiguration	否
Status	枚舉字元串	指定LiveChannel的狀態。 子節點：無 父節點：LiveChannelConfiguration 有效值：enabled、disabled 預設值：enabled	否
Target	容器	保存轉儲配置的容器。 子節點：Type、FragDuration、FragCount、PlaylistName 父節點：LiveChannelConfiguration	是
Type	枚舉字元串	指定轉儲的類型。 子節點：無 父節點：Target 有效值：HLS	是
FragDuration	字元串	當Type為HLS時，指定每個ts檔案的時長（單位：秒），取值範圍為[1, 100]的整數。 子節點：無 父節點: Target	否

名稱	類型	描述	是否必需
		預設值：5	
FragCount	字元串	當Type為HLS時，指定m3u8檔案中包含ts檔案的個數，取值範圍為[1, 100]的整數。 子節點：無 父節點：Target 預設值: 3	否
PlaylistName	字元串	當Type為HLS時，指定生成的m3u8檔案的名稱，必須以".m3u8"結尾，長度範圍為[6, 128]。 子節點：無 父節點：Target 預設值：playlist.m3u8	否
Snapshot	容器	保存高頻截圖操作Snapshot 選項的容器。 子節點：RoleName，DestBucket，NotifyTopic，Interval，PornRec 父節點：Snapshot	否
RoleName	字元串	用於高頻截圖操作的角色名稱稱，要求有DestBucket的寫入權限和向NotifyTopic發消息的許可權。 子節點：無 父節點：Snapshot	否
DestBucket	字元串	保存高頻截圖目標Bucket，要求與當前Bucket是同一個Owner。 子節點：無 父節點：Snapshot	否

名稱	類型	描述	是否必需
NotifyTopic	字元串	用於通知用戶高頻截圖操作結果的MNS的Topic。 子節點：無 父節點：Snapshot	否
Interval	數字	高頻截圖的間隔長度，單位為s，如果該段間隔時間內沒有主要畫面格（I幀），那麼該間隔時間不截圖。 子節點：無 父節點：Snapshot	否

細節分析

- ChannelName必須符合ObjectName的命名規範，另外，ChannelName不能包含"/"。

- FragDuration和FragCount的預設值只有在兩者都未指定時才會生效；指定了其中一個，則另一個的值也必須指定。

- 轉儲類型為HLS時，OSS會在生成每個ts檔案後，更新m3u8檔案；m3u8檔案中最多包含最近的FragCount個ts檔案。

- 轉儲類型為HLS時，寫入當前ts檔案的音視頻數據時長達到FragDuration指定的時長後，OSS會在收到下一個主要畫面格的時候切換到下一個ts檔案；如果max(2*FragDuration, 60s)後仍未收到下一個主要畫面格，OSS強制切換檔案，此時可能引起播放時卡頓。

響應元素

名稱	類型	描述
CreateLiveChannelResult	容器	保存CreateLiveChannel請求結果的容器。 子節點：PublishUrls,PlayUrls 父節點：無
PublishUrls	容器	保存推流地址的容器。 子節點：Url 父節點：CreateLiveChannelResult
Url	字元串	推流地址。 子節點：無 父節點：PublishUrls

名稱	類型	描述
PlayUrls	容器	保存推流地址的容器。 子節點：Url 父節點：CreateLive ChannelResult
Url	字元串	播放地址。 子節點：無 父節點：PlayUrls

細節分析

- 推流地址是未加簽名的url，如果bucket acl非public-read-write，那麼需要首先進行簽名才能進行訪問。

- 播放地址是未加簽名的url，如果bucket acl為private，那麼需要首先進行簽名才能進行訪問。

實例

請求樣本

```
PUT /test-channel?live HTTP/1.1
Date: Wed, 24 Aug 2016 11:11:28 GMT
Content-Length: 333
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:hvwOZJRh8toAj3DZvtsuPgf+agA=
<?xml version="1.0" encoding="utf-8"?>
<LiveChannelConfiguration>
    <Description/>
    <Status>enabled</Status>
    <Target>
        <Type>HLS</Type>
        <FragDuration>2</FragDuration>
        <FragCount>3</FragCount>
    </Target>
    <Snapshot>
        <RoleName>role_for_snapshot</RoleName>
        <DestBucket>snapshotdest</DestBucket>
        <NotifyTopic>snapshotnotify</NotifyTopic>
        <Interval>1</Interval>
     </Snapshot>
</LiveChannelConfiguration>
```

返回樣本

```
HTTP/1.1 200
content-length: 259
server: AliyunOSS
x-oss-server-time: 4
connection: close
x-oss-request-id: 57BD8419B92475920B0002F1
date: Wed, 24 Aug 2016 11:11:28 GMT
x-oss-bucket-storage-type: standard
content-type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CreateLiveChannelResult>
  <PublishUrls>
    <Url>rtmp://test-bucket.oss-cn-hangzhou.aliyuncs.com/live/test-
channel</Url>
  </PublishUrls>
  <PlayUrls>
    <Url>http://test-bucket.oss-cn-hangzhou.aliyuncs.com/test-channel/
playlist.m3u8</Url>
  </PlayUrls>
</CreateLiveChannelResult>
```

10.5 PostVodPlaylist

`PostVodPlaylist`介面用來為指定LiveChannel推流生成的，指定時間段內的ts檔案生成一個點播用的播放清單（m3u8檔案）。

請求文法

```
POST /ChannelName/PlaylistName?vod&endTime=EndTime&startTime=StartTime
 HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

請求參數

名稱	描述	是否必需
`ChannelName`	指定LiveChannel名稱，該LiveChannel必須存在。	是
`PlaylistName`	指定生成的點播播放清單的名稱，必須以".m3u8"結尾。	是
`StartTime`	指定查詢ts檔案的起始時間，格式為Unix timestamp。	是
`EndTime`	指定查詢ts檔案的終止時間，格式為Unix timestamp。	是

細節分析

- EndTime必須大於StartTime，且時間跨度不能大於1天。

- OSS會查詢指定時間範圍內的所有該LiveChannel推流生成的ts檔案，並將其拼裝為一個播放清單。

樣本

請求樣本

```
POST /test-channel/vod.m3u8?vod&endTime=1472020226&startTime=
1472020031 HTTP/1.1
Date: Thu, 25 Aug 2016 07:13:26 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:ABIigvnLtCHK+7fMHLeRlOUnzv0=
```

返回樣本

```
HTTP/1.1 200
content-length: 0
server: AliyunOSS
connection: close
etag: "9C6104DD9CF1A0C4D0CFD21F43905D59"
x-oss-request-id: 57BE9A96B92475920B002359
date: Thu, 25 Aug 2016 07:13:26 GMT
```

10.6 GetLiveChannelStat

`GetLiveChannelStat`介面用來獲取指定LiveChannel的推流狀態資訊。

請求文法

```
GET /ChannelName?live&comp=stat HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

響應元素

名稱	類型	描述
`LiveChannelStat`	容器	保存GetLiveChannelStat返回結果的容器。 子節點：Status，ConnectedTime，Video，Audio 父節點：無
`Status`	枚舉字元串	LiveChannel當前的推流狀態原因。 子節點：無 父節點：LiveChannelStat 有效值：`Disabled`，`Live`，`Idle`
`ConnnectedTime`	字元串	當Status為Live時，表示當前客戶端開始推流的時間，使用ISO8601格式表示。

名稱	類型	描述
		子節點：無 父節點：LiveChannelStat
RemoteAddr	字元串	當Status為Live時，表示當前推流客戶端的ip地址。 子節點：無 父節點：LiveChannelStat
Video	容器	當Status為Live時，保存視頻流資訊的容器。 子節點：Width，Heigth，FrameRate，Bandwidth，Codec 父節點：LiveChannelStat
Width	字元串	表示當前視頻流的畫面寬度（單位：像素）。 子節點：無 父節點：Video
Heigth	字元串	表示當前視頻流的畫面高度（單位：像素）。 子節點：無 父節點：Video
FrameRate	字元串	表示當前視頻流的幀率。 子節點：無 父節點：Video
Bandwidth	字元串	表示當前視頻流的碼率（單位：B/s)。 子節點：無 父節點：Video
Codec	枚舉字元串	表示當前視頻流的編碼格式。 子節點：無 父節點：Video
Audio	容器	當Status為Live時，保存音頻流資訊的容器。 子節點：SampleRate，Bandwidth，Codec 父節點：LiveChannelStat
SampleRate	字元串	表示當前音頻流的採樣率。 子節點：無 父節點：Audio

名稱	類型	描述
Bandwidth	字元串	表示當前音頻流的碼率（單位：B/s)。 子節點：無 父節點：Audio
Codec	枚舉字元串	表示當前音頻流的編碼格式。 子節點：無 父節點：Audio

細節分析

- Video，Audio容器只有在Status為Live時才會返回，但Status為Live時不一定會返回Video，Audio容器，例如，客戶端已經串連到LiveChannel，但尚未發送音視頻數據時不會返回。

- Bandwidth為音頻流/視頻流最近一段時間內的平均碼率，LiveChannel剛切換到Live狀態時，返回的Bandwidth值可能為0。

樣本

請求樣本I

```
GET /test-channel?live&comp=stat HTTP/1.1
Date: Thu, 25 Aug 2016 06:22:01 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:fOzwkAgVTVSO1VKLPIInQ0JYyOA=
```

返回樣本I

```
HTTP/1.1 200
content-length: 100
server: AliyunOSS
connection: close
x-oss-request-id: 57BE8E89B92475920B002164
date: Thu, 25 Aug 2016 06:22:01 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<LiveChannelStat>
  <Status>Idle</Status>
</LiveChannelStat>
```

請求樣本II

```
GET /test-channel?live&comp=stat HTTP/1.1
Date: Thu, 25 Aug 2016 06:25:26 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:WeC5joEaRzfSSS8xK0tlo7WTK1I=
```

返回樣本II

```
HTTP/1.1 200
```

```
content-length: 469
server: AliyunOSS
connection: close
x-oss-request-id: 57BE8F56B92475920B002187
date: Thu, 25 Aug 2016 06:25:26 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<LiveChannelStat>
  <Status>Live</Status>
  <ConnectedTime>2016-08-25T06:25:15.000Z</ConnectedTime>
  <RemoteAddr>10.1.2.3:47745</RemoteAddr>
  <Video>
    <Width>1280</Width>
    <Height>536</Height>
    <FrameRate>24</FrameRate>
    <Bandwidth>0</Bandwidth>
    <Codec>H264</Codec>
  </Video>
  <Audio>
    <Bandwidth>0</Bandwidth>
    <SampleRate>44100</SampleRate>
    <Codec>ADPCM</Codec>
  </Audio>
</LiveChannelStat>
```

10.7 GetLiveChannelInfo

`GetLiveChannelInfo`介面用來獲取指定LiveChannel的配置資訊。

請求文法

```
GET /ChannelName?live HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

響應元素

名稱	類型	描述
`LiveChannelConfiguration`	容器	保存GetLiveChannelInfo返回結果的容器。 子節點：Description、Status、Target 父節點：無
`Description`	字元串	LiveChannel的描述資訊。 子節點：無 父節點：LiveChannelConfiguration
`Status`	枚舉字元串	LiveChannel的狀態資訊。 子節點：無 父節點：LiveChannelConfiguration

名稱	類型	描述
		有效值：`enabled`、`disabled`
Target	容器	保存LiveChannel轉儲配置的容器。 子節點：Type、FragDuration、FragCount、PlaylistName 父節點：LiveChannelConfiguration
Type	枚舉字元串	當Type為HLS時，指定推流時轉儲檔案類型。 子節點：無 父節點：Target 有效值：`HLS`
FragDuration	字元串	當Type為HLS時，指定每個ts檔案的時長（單位：秒）。 子節點：無 父節點: Target
FragCount	字元串	當Type為HLS時，指定m3u8檔案中包含ts檔案的個數。 子節點：無 父節點：Target
PlaylistName	字元串	當Type為HLS時，指定生成的m3u8檔案的名稱。 子節點：無 父節點：Target

細節分析

Target的子節點FragDuration，FragCount，PlaylistName只有當Type取值為HLS時才會返回。

樣本

請求樣本

```
GET /test-channel?live HTTP/1.1
Date: Thu, 25 Aug 2016 05:52:40 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:D6bDCRXKht58hin1BL83wxyGvl0=
```

返回樣本

```
HTTP/1.1 200
content-length: 475
```

```
server: AliyunOSS
connection: close
x-oss-request-id: 57BE87A8B92475920B002098
date: Thu, 25 Aug 2016 05:52:40 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<LiveChannelConfiguration>
  <Description></Description>
  <Status>enabled</Status>
  <Target>
    <Type>HLS</Type>
    <FragDuration>2</FragDuration>
    <FragCount>3</FragCount>
    <PlaylistName>playlist.m3u8</PlaylistName>
  </Target>
</LiveChannelConfiguration>
```

10.8 GetLiveChannelHistory

`GetLiveChannelHistory`介面用來獲取指定LiveChannel的推流記錄。

請求文法

```
GET /ChannelName?live&comp=history HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

響應元素

名稱	類型	描述
`LiveChannelHistory`	容器	保存GetLiveChannelHistory返回結果的容器。 子節點：LiveRecord 父節點：無
`LiveRecord`	容器	保存一次推流記錄資訊的容器。 子節點：StartTime，EndTime，RemoteAddr 父節點：LiveChannelHistory
`StartTime`	字元串	推流開始時間，使用ISO8601格式表示。 子節點：無 父節點：LiveRecord
`EndTime`	字元串	推流結束時間，使用ISO8601格式表示。 子節點：無 父節點：LiveRecord

名稱	類型	描述
RemoteAddr	字元串	推流客戶端的ip地址。 子節點：無 父節點：LiveRecord

細節分析

目前最多會返回指定LiveChannel最近的10次推流記錄。

樣本

請求樣本

```
GET /test-channel?live&comp=history HTTP/1.1
Date: Thu, 25 Aug 2016 07:00:12 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:pqgDBP8JXTXAytBoXpvNoZfo68k=
```

返回實例

```
HTTP/1.1 200
content-length: 1892
server: AliyunOSS
connection: close
x-oss-request-id: 57BE977CB92475920B0022FB
date: Thu, 25 Aug 2016 07:00:12 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<LiveChannelHistory>
  <LiveRecord>
    <StartTime>2016-07-30T01:53:21.000Z</StartTime>
    <EndTime>2016-07-30T01:53:31.000Z</EndTime>
    <RemoteAddr>10.101.194.148:56861</RemoteAddr>
  </LiveRecord>
  <LiveRecord>
    <StartTime>2016-07-30T01:53:35.000Z</StartTime>
    <EndTime>2016-07-30T01:53:45.000Z</EndTime>
    <RemoteAddr>10.101.194.148:57126</RemoteAddr>
  </LiveRecord>
  <LiveRecord>
    <StartTime>2016-07-30T01:53:49.000Z</StartTime>
    <EndTime>2016-07-30T01:53:59.000Z</EndTime>
    <RemoteAddr>10.101.194.148:57577</RemoteAddr>
  </LiveRecord>
  <LiveRecord>
    <StartTime>2016-07-30T01:54:04.000Z</StartTime>
    <EndTime>2016-07-30T01:54:14.000Z</EndTime>
    <RemoteAddr>10.101.194.148:57632</RemoteAddr>
  </LiveRecord>
```

```
</LiveChannelHistory>
```

10.9 ListLiveChannel

ListLiveChannel介面用於羅列指定的LiveChannel。

請求文法

```
GET /?live HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

請求參數

名稱	描述	是否必需
marker	設定結果從marker之後按字母排序的第一個開始返回。	否
max-keys	限定此次返回LiveChannel的最大數，如果不設定，預設為100，max-keys取值不能大於1000。 預設值：100	否
prefix	限定返回的LiveChannel必須以prefix作為首碼。注意使用prefix查詢時，返回的key中仍會包含prefix。	否

響應元素

名稱	類型	描述
ListLiveChannelResult	容器	保存ListLiveChannel請求結果的容器。 子節點：Prefix，Marker，MaxKeys，IsTruncated，NextMarker，LiveChannel 父節點：無
Prefix	字元串	本次查詢結果的開始首碼。 子節點：無 父節點：ListLiveChannelResult
Marker	字元串	本次ListLiveChannel的起點。 子節點：無

名稱	類型	描述
		父節點：ListLiveChannelResult
MaxKeys	字元串	響應請求內返回結果的最大數目。 子節點：無 父節點：ListLiveChannelResult
IsTruncated	字元串	指明是否所有的結果都已經返回；"true"表示本次沒有返回全部結果；"false"表示本次已經返回了全部結果。 子節點：無 父節點：ListLiveChannelResult
NextMarker	字元串	如果本次沒有返回全部結果，響應請求中將包含NextMarker元素，用於標明接下來請求的Marker值。 子節點：無 父節點：ListLiveChannelResult
LiveChannel	容器	保存返回每個LiveChannel資訊的容器。 子節點：Name，Description，Status，LastModified，PublishUrls，PlayUrls 父節點：ListLiveChannelResult
Name	字元串	LiveChannel的名稱。 子節點：無 父節點：LiveChannel
Description	字元串	LiveChannel的描述。 子節點：無 父節點：LiveChannel
Status	枚舉字元串	LiveChannel的狀態。 子節點：無 父節點：LiveChannel 有效值：disabled，enabled

名稱	類型	描述
`LastModified`	字元串	LiveChannel配置的最後修改時間，使用ISO8601格式表示。 子節點：無 父節點：LiveChannel
`PublishUrls`	容器	保存LiveChannel對應的推流地址的容器。 子節點：Url 父節點：LiveChannel
`Url`	字元串	LiveChannel對應的推流地址。 子節點：無 父節點：PublishUrls
`PlayUrls`	容器	保存LiveChannel對應的播放地址的容器。 子節點：Url 父節點：LiveChannel
`Url`	字元串	LiveChannel對應的播放地址。 子節點：無 父節點：PlayUrls

樣本

請求樣本

```
GET /?live&max-keys=1 HTTP/1.1
Date: Thu, 25 Aug 2016 07:50:09 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:TaX+tlc/Xsgpz6uRuqcbmUJsIHw=
```

返回樣本

```
HTTP/1.1 200
content-length: 656
server: AliyunOSS
connection: close
x-oss-request-id: 57BEA331B92475920B00245E
date: Thu, 25 Aug 2016 07:50:09 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<ListLiveChannelResult>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <NextMarker>channel-0</NextMarker>
  <LiveChannel>
    <Name>channel-0</Name>
    <Description></Description>
```

```
   <Status>disabled</Status>
   <LastModified>2016-07-30T01:54:21.000Z</LastModified>
   <PublishUrls>
      <Url>rtmp://test-bucket.oss-cn-hangzhou.aliyuncs.com/live/
channel-0</Url>
   </PublishUrls>
   <PlayUrls>
      <Url>http://test-bucket.oss-cn-hangzhou.aliyuncs.com/channel-0/
playlist.m3u8</Url>
   </PlayUrls>
 </LiveChannel>
```

10.10 DeleteLiveChannel

DeleteLiveChannel介面用來刪除指定的LiveChannel。

請求文法

```
DELETE /ChannelName?live HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

細節分析

- 當有客戶端正在向LiveChannel推流時，刪除請求會失敗。

- 本介面只會刪除LiveChannel本身，不會刪除推流生成的檔案。

樣本

請求樣本

```
DELETE /test-channel?live HTTP/1.1
Date: Thu, 25 Aug 2016 07:32:26 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJjHKOKWDWINLKXv:ZbfvQ3XwmYEE8O9CX8kwVQYNbzQ=
```

返回樣本

```
HTTP/1.1 204
content-length: 0
server: AliyunOSS
connection: close
x-oss-request-id: 57BE9F0AB92475920B0023E0
date: Thu, 25 Aug 2016 07:32:26 GMT
```