

阿里云

对象存储 OSS API 参考

文档版本：20211029

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.简介	09
2.API概览	10
3.公共HTTP头定义	15
4.访问控制	17
4.1. 用户签名验证	17
4.2. 在Header中包含签名	17
4.3. 在URL中包含签名	23
4.4. Bucket权限控制	27
5.关于Service操作	28
5.1. GetService (ListBuckets)	28
6.关于Bucket操作	34
6.1. 基础操作	34
6.1.1. PutBucket	34
6.1.2. DeleteBucket	37
6.1.3. GetBucket (ListObjects)	39
6.1.4. GetBucketV2 (ListObjectsV2)	49
6.1.5. GetBucketInfo	58
6.1.6. GetBucketLocation	63
6.2. 合规保留策略 (WORM)	65
6.2.1. InitiateBucketWorm	65
6.2.2. AbortBucketWorm	66
6.2.3. CompleteBucketWorm	66
6.2.4. ExtendBucketWorm	67
6.2.5. GetBucketWorm	68
6.3. 权限控制 (ACL)	69
6.3.1. PutBucketAcl	69

6.3.2. GetBucketAcl	71
6.4. 生命周期 (Lifecycle)	73
6.4.1. PutBucketLifecycle	73
6.4.2. GetBucketLifecycle	83
6.4.3. DeleteBucketLifecycle	87
6.5. 传输加速 (TransferAcceleration)	88
6.5.1. PutBucketTransferAcceleration	88
6.5.2. GetBucketTransferAcceleration	90
6.6. 版本控制 (Versioning)	91
6.6.1. PutBucketVersioning	91
6.6.2. GetBucketVersioning	93
6.6.3. GetBucketVersions(ListObjectVersions)	95
6.7. 数据复制 (Replication)	103
6.7.1. PutBucketReplication	103
6.7.2. GetBucketReplication	109
6.7.3. GetBucketReplicationLocation	112
6.7.4. GetBucketReplicationProgress	114
6.7.5. DeleteBucketReplication	118
6.8. 授权策略 (Policy)	121
6.8.1. PutBucketPolicy	121
6.8.2. GetBucketPolicy	122
6.8.3. DeleteBucketPolicy	123
6.9. 清单 (Inventory)	124
6.9.1. PutBucketInventory	124
6.9.2. GetBucketInventory	129
6.9.3. ListBucketInventory	132
6.9.4. DeleteBucketInventory	137
6.10. 日志管理 (Logging)	138

6.10.1. PutBucketLogging	138
6.10.2. GetBucketLogging	141
6.10.3. DeleteBucketLogging	143
6.11. 静态网站 (Website)	144
6.11.1. PutBucketWebsite	144
6.11.2. GetBucketWebsite	156
6.11.3. DeleteBucketWebsite	165
6.12. 防盗链 (Referer)	166
6.12.1. PutBucketReferer	166
6.12.2. GetBucketReferer	169
6.13. 标签 (Tags)	171
6.13.1. PutBucketTags	171
6.13.2. GetBucketTags	173
6.13.3. DeleteBucketTags	175
6.14. 加密 (Encryption)	176
6.14.1. PutBucketEncryption	176
6.14.2. GetBucketEncryption	181
6.14.3. DeleteBucketEncryption	183
6.15. 请求者付费 (RequestPayment)	184
6.15.1. PutBucketRequestPayment	184
6.15.2. GetBucketRequestPayment	186
6.16. 跨域资源共享 (CORS)	187
6.16.1. 简介	187
6.16.2. PutBucketCors	187
6.16.3. GetBucketCors	191
6.16.4. DeleteBucketCors	193
6.16.5. Options	195
7.关于Object操作	197

7.1. 基础操作	197
7.1.1. PutObject	197
7.1.2. GetObject	203
7.1.3. CopyObject	211
7.1.4. AppendObject	219
7.1.5. DeleteObject	225
7.1.6. DeleteMultipleObjects	228
7.1.7. HeadObject	236
7.1.8. GetObjectMeta	242
7.1.9. PostObject	245
7.1.10. Callback	255
7.1.11. RestoreObject	266
7.1.12. SelectObject	271
7.2. 分片上传 (MultipartUpload)	294
7.2.1. InitiateMultipartUpload	295
7.2.2. UploadPart	299
7.2.3. UploadPartCopy	301
7.2.4. CompleteMultipartUpload	304
7.2.5. AbortMultipartUpload	311
7.2.6. ListMultipartUploads	312
7.2.7. ListParts	316
7.3. 权限控制 (ACL)	320
7.3.1. PutObjectACL	320
7.3.2. GetObjectACL	323
7.4. 软链接 (Symlink)	326
7.4.1. PutSymlink	326
7.4.2. GetSymlink	329
7.5. 标签 (Tagging)	331

7.5.1. PutObjectTagging	331
7.5.2. GetObjectTagging	334
7.5.3. DeleteObjectTagging	337
8.关于LiveChannel的操作	339
8.1. LiveChannel简介	339
8.2. RTMP推流地址及签名	339
8.3. PutLiveChannel	340
8.4. ListLiveChannel	346
8.5. DeleteLiveChannel	349
8.6. PutLiveChannelStatus	350
8.7. GetLiveChannelInfo	351
8.8. GetLiveChannelStat	354
8.9. GetLiveChannelHistory	358
8.10. PostVodPlaylist	360
8.11. GetVodPlaylist	361
8.12. LiveChannel常见问题	363

1.简介

阿里云对象存储服务（Object Storage Service，简称OSS），是阿里云对外提供的海量、安全、低成本、高可靠的云存储服务。您可以通过本文档提供的简单的REST接口，在任何时间、任何地点、任何互联网设备上上传和下载数据。基于OSS，您可以搭建出各种多媒体分享网站、网盘、个人和企业数据备份等基于大规模数据的服务。

使用限制

您使用的OSS资源和相关功能，都有一定的限制，具体请参见[OSS使用限制](#)。

使用说明

OSS API参考主要介绍接口的请求语法、相关参数含义以及请求和返回示例。如果要进行快速二次开发，建议您使用SDK开发包。关于SDK的安装和使用，请参见[OSS SDK参考](#)。

OSS定价

关于OSS的价格，请参见[OSS详细价格信息](#)。关于OSS的计量计费方式，请参见[OSS计量项和计费项](#)。

错误码

请参见[OSS常见错误响应](#)。

资源术语

中文	英文	说明
存储空间	Bucket	存储空间是您用于存储对象（Object）的容器，所有的对象都必须隶属于某个存储空间。
对象/文件	Object	对象是 OSS 存储数据的基本单元，也被称为OSS的文件。对象由元信息（Object Meta）、用户数据（Data）和文件名（Key）组成。对象由存储空间内部唯一的Key来标识。
地域	Region	地域表示 OSS 的数据中心所在物理位置。您可以根据费用、请求来源等综合选择数据存储的地域。详情请查看 OSS已经开通的Region 。
访问域名	Endpoint	Endpoint 表示OSS对外服务的访问域名。OSS以HTTP RESTful API的形式对外提供服务，当访问不同地域的时候，需要不同的域名。通过内网和外网访问同一个地域所需要的域名也是不同的。具体的内容请参见 各个Region对应的Endpoint 。
访问密钥	AccessKey	AccessKey，简称 AK，指的是访问身份验证中用到的 AccessKeyId 和AccessKeySecret。OSS通过使用 AccessKeyId 和AccessKeySecret对称加密的方法来验证某个请求的发送者身份。AccessKeyId用于标识用户，AccessKeySecret是用户用于加密签名字符串和OSS用来验证签名字符串的密钥，其中AccessKeySecret 必须保密。

2.API概览

本文介绍对象存储OSS提供的相关API接口及其各API接口的用法。

关于Service操作

API	描述
GetService (ListBuckets)	返回请求者拥有的所有存储空间（Bucket）。

关于Bucket操作

分类	API	描述
基础操作	PutBucket	创建Bucket。
	DeleteBucket	删除Bucket。
	GetBucket(ListObject)	列出Bucket中所有文件（Object）的信息。
	GetBucketV2 (ListObjectsV2)	
	GetBucketInfo	获取Bucket信息。
	GetBucketLocation	获得Bucket所属的位置信息。
合规保留策略（WORM）	InitiateBucketWorm	新建合规保留策略。
	AbortBucketWorm	删除未锁定的合规保留策略。
	CompleteBucketWorm	锁定合规保留策略。
	ExtendBucketWorm	延长已锁定的合规保留策略对应Bucket中Object的保留天数。
	GetBucketWorm	查看Bucket的合规保留策略信息。
权限控制（ACL）	PutBucketAcl	设置Bucket访问权限。
	GetBucketAcl	获得Bucket访问权限。
生命周期（Lifecycle）	PutBucketLifecycle	设置Bucket中Object的生命周期规则。
	GetBucketLifecycle	查看Bucket中Object的生命周期规则。
	DeleteBucketLifecycle	删除Bucket中Object的生命周期规则。
传输加速 (TransferAcceleration)	PutBucketTransferAcceleration	为存储空间（Bucket）配置传输加速。
	GetBucketTransferAcceleration	获取目标存储空间（Bucket）的传输加速配置。

分类	API	描述
版本控制 (Versioning)	PutBucketVersioning	设置Bucket的版本控制状态。
	GetBucketVersioning	获取Bucket的版本控制状态。
	GetBucketVersions(ListObjectVersions)	列举Bucket中所有Object的版本信息。
跨区域复制 (Replication)	PutBucketReplication	设置Bucket的跨区域复制规则。
	GetBucketReplication	查看Bucket已设置的跨区域复制规则。
	GetBucketReplicationLocation	查看可复制到的目标Bucket所在的地域。
	GetBucketReplicationProgress	查看Bucket的跨区域复制进度。
	DeleteBucketReplication	停止Bucket的跨区域复制任务并删除Bucket的复制配置。
授权策略 (Policy)	PutBucketPolicy	设置Bucket Policy。
	GetBucketPolicy	获取Bucket Policy。
	DeleteBucketPolicy	删除Bucket Policy。
清单 (Inventory)	PutBucketInventory	设置Bucket清单规则。
	GetBucketInventory	查看Bucket中指定的清单任务。
	ListBucketInventory	查看Bucket中所有的清单任务。
	DeleteBucketInventory	删除Bucket中指定的清单任务。
日志管理 (Logging)	PutBucketLogging	开启Bucket访问日志记录功能。
	GetBucketLogging	查看Bucket的访问日志配置情况。
	DeleteBucketLogging	关闭Bucket访问日志记录功能。
静态网站 (Website)	PutBucketWebsite	设置Bucket为静态网站托管模式。
	GetBucketWebsite	查看Bucket的静态网站托管状态。
	DeleteBucketWebsite	关闭Bucket的静态网站托管模式。
防盗链 (Referer)	PutBucketReferer	设置Bucket的防盗链规则。
	GetBucketReferer	查看Bucket的防盗链规则。
	PutBucketTags	添加或修改Bucket标签。

标签 (Tags)	API	描述
	GetBucketTags	查看Bucket标签信息。
	DeleteBucketTags	删除Bucket标签。
加密 (Encryption)	PutBucketEncryption	配置Bucket的加密规则。
	GetBucketEncryption	获取Bucket的加密规则。
	DeleteBucketEncryption	删除Bucket的加密规则。
请求者付费 (RequestPayment)	PutBucketRequestPayment	设置Bucket为请求者付费模式。
	GetBucketRequestPayment	查看Bucket请求者付费模式配置信息。
跨域资源共享 (CORS)	PutBucketCors	为指定的存储空间 (Bucket) 设置跨域资源共享 CORS (Cross-Origin Resource Sharing) 规则。
	GetBucketCors	获取指定存储空间 (Bucket) 当前的跨域资源共享CORS规则。
	DeleteBucketCors	关闭指定存储空间 (Bucket) 对应的跨域资源共享CORS功能并清空所有规则。
	Options	浏览器在发送跨域请求之前会发送一个preflight请求 (Options) 给OSS, 并带上特定的来源域、HTTP方法和header等信息, 以决定是否发送真正的请求。

关于Object的操作

分类	API	描述
基础操作	PutObject	上传Object。
	GetObject	获取Object。
	CopyObject	拷贝Object。
	AppendObject	以追加写的方式上传Object。
	DeleteObject	删除单个Object。
	DeleteMultipleObjects	删除多个Object。
	HeadObject	只返回某个Object的meta信息, 不返回文件内容。
	GetObjectMeta	返回Object的基本meta信息, 包括该Object的ETag、Size (文件大小) 以及LastModified等, 不返回文件内容。
	PostObject	通过HTML表单上传的方式上传Object。

分类	API	描述
	Callback	您只需在发送给OSS的请求中携带相应的Callback参数即能实现回调。
	RestoreObject	解冻归档类型（Archive）或冷归档（Cold Archive）的文件（Object）。
	SelectObject	对目标文件执行SQL语句，返回执行结果。
分片上传 (Multipart Upload)	InitiateMultipartUpload	初始化一个Multipart Upload事件。
	UploadPart	根据指定的Object名和uploadId来分块（Part）上传数据。
	UploadPartCopy	通过在UploadPart请求的基础上增加一个请求头x-oss-copy-source来调用UploadPartCopy接口，实现从一个已存在的Object中拷贝数据来上传一个Part。
	CompleteMultipartUpload	在将所有数据Part都上传完成后，您必须调用CompleteMultipartUpload接口来完成整个文件的分片上传。
	AbortMultipartUpload	取消Multipart Upload事件并删除对应的Part数据。
	ListMultipartUploads	列举所有执行中的Multipart Upload事件，即已经初始化但还未完成（Complete）或者还未中止（Abort）的Multipart Upload事件。
	ListParts	列举指定uploadId所属的所有已经上传成功Part。
权限控制（ACL）	PutObjectACL	修改Object的访问权限。
	GetObjectACL	查看Object的访问权限。
软链接（Symlink）	PutSymlink	创建软链接。
	GetSymlink	获取软链接。
标签（Tagging）	PutObjectTagging	设置或更新对象标签。
	GetObjectTagging	获取对象标签信息。
	DeleteObjectTagging	删除指定的对象标签。

关于Live Channel的操作

API	描述
PutLiveChannelStatus	切换LiveChannel的状态。
PutLiveChannel	创建LiveChannel。

API	描述
GetVodPlaylist	获取播放列表。
PostVodPlaylist	生成播放列表。
GetLiveChannelStat	获取LiveChannel的推流状态信息。
GetLiveChannelInfo	获取LiveChannel的配置信息。
GetLiveChannelHistory	获取LiveChannel的推流记录。
ListLiveChannel	列举LiveChannel。
DeleteLiveChannel	删除LiveChannel。

3.公共HTTP头定义

本文介绍了对象存储OSS的公共请求头和公共响应头的详细说明。

公共请求头（Common Request Headers）

OSS的RESTful接口中使用了一些公共请求头。这些请求头可以被所有的OSS请求所使用，其详细定请参见下表。

名称	类型	描述
Authorization	字符串	用于验证请求合法性的认证信息。 默认值：无 使用场景：非匿名请求
Content-Length	字符串	RFC 2616 中定义的HTTP请求内容长度。 默认值：无 使用场景：需要向OSS提交数据的请求
Content-Type	字符串	RFC 2616 中定义的HTTP请求内容类型。 默认值：无 使用场景：需要向OSS提交数据的请求
Date	字符串	HTTP 1.1协议中规定的GMT时间，例如Sun, 05 Sep 2021 23:00:00 GMT。 默认值：无
Host	字符串	访问Host值，格式为 <bucketname>.oss-cn-hangzhou.aliyuncs.com 。 默认值：无

公共响应头（Common Response Headers）

OSS的RESTful接口中使用了一些公共响应头。这些响应头可以被所有的OSS请求所使用，其详细定义请参见下表。

名称	类型	描述
----	----	----

名称	类型	描述
Content-Length	字符串	RFC 2616中定义的HTTP请求内容长度。 默认值：无 使用场景：需要向OSS提交数据的请求
Connection	枚举	标明客户端和OSS服务器之间的链接状态。 有效值：<i>keep-alive</i>、<i>close</i> 默认值：无
Date	字符串	HTTP 1.1协议中规定的GMT时间，例如Sun, 05 Sep 2021 23:00:00 GMT。 默认值：无
ETag	字符串	ETag (entity tag) 在每个Object生成的时候被创建，用于标识一个Object的内容。对于Put Object请求创建的Object，ETag值是其内容的MD5值；对于其他方式创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。ETag值可以用于检查Object内容是否发生变化。 默认值：无
Server	字符串	生成Response的服务器。 默认值：AliyunOSS
x-oss-request-id	字符串	x-oss-request-id是由Aliyun OSS创建，并唯一标识这个response的UUID。如果在使用OSS服务时遇到问题，可以凭借该字段联系OSS工作人员，快速定位问题。 默认值：无

4. 访问控制

4.1. 用户签名验证

OSS通过使用AccessKey ID、AccessKey Secret对称加密的方法来验证某个请求的发送者身份。

AccessKey包含AccessKey ID和AccessKey Secret。其中，AccessKey ID用于标识用户，AccessKey Secret是用户用于加密签名字符串和OSS用来验证签名字符串的密钥，AccessKey Secret必须保密。AccessKey根据所属账号的类型有所区分。

- 阿里云账户AccessKey：每个阿里云账号提供的AccessKey对拥有的资源有完全的权限。
- RAM账户AccessKey：RAM账户由阿里云账号授权生成，所用的AccessKey拥有对特定资源限定的操作权限。
- STS临时访问凭证：通过STS服务生成的临时访问凭证包括安全令牌（SecurityToken）、临时访问密钥STS AccessKey（AccessKey ID和AccessKey Secret）。所用的AccessKey在限定时间内拥有对特定资源限定的操作权限。临时访问凭证过期后，权限将自动失效。

当您想以个人身份向OSS发送请求时，首先需要将发送的请求按照OSS指定的格式生成签名字符串，然后使用AccessKeySecret对签名字符串进行加密产生验证码。OSS收到请求后，通过AccessKey ID找到对应的AccessKey Secret，并以同样的方法提取签名字符串和验证码。如果计算出来的验证码和提供的一致则认为该请求有效；否则，OSS将拒绝处理这次请求，并返回HTTP 403错误。

有关OSS权限控制的详情，请参见[权限控制概述](#)。

4.2. 在Header中包含签名

您可以在HTTP请求中增加 `Authorization` 的Header来包含签名（Signature）信息，表明该消息已被授权。

SDK签名实现

OSS SDK已实现签名，您使用OSS SDK时无需关注签名问题。如果您想了解具体语言的签名实现，请参考OSS SDK的代码。OSS SDK签名实现的文件请参见下表。

SDK	签名实现
Java SDK	<code>OSSRequestSigner.java</code>
Python SDK	<code>auth.py</code>
.Net SDK	<code>OssRequestSigner.cs</code>
PHP SDK	<code>OssClient.php</code>
C SDK	<code>oss_auth.c</code>
JavaScript SDK	<code>client.js</code>
Go SDK	<code>auth.go</code>
Ruby SDK	<code>util.rb</code>
iOS SDK	<code>OSSModel.m</code>

SDK	签名实现
Android SDK	OSSUtils.java

当您自己实现签名，访问OSS报 `SignatureDoesNotMatch` 错误时，请参见[自签名计算失败排除错误](#)。

Authorization字段计算的方法

- 计算方法

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

- 参数

参数	是否必选	示例值	说明
AccessKeySecret	是	OtxrxlsfpFjA7Sw* *****8Bw21TLhqu hboDYROV	签名所需的密钥。
VERB	是	PUT	HTTP请求的Method，主要有PUT、GET、POST、HEAD、DELETE等。
\n	否	\n	换行符。
Content-MD5	否	eB5eJF1ptWaXm4 bjiSPyxw==	请求内容数据的MD5值，对消息内容（不包括头部）计算MD5值获得128比特位数字，对该数字进行base64编码得出。更多信息，请参见 RFC2616 Content-MD5 。 该请求头可用于消息合法性的检查（消息内容是否与发送时一致），也可以为空。 关于Content-MD5的计算方法，请参见 Content-MD5的计算方法 。
Content-Type	否	application/octet-stream	请求内容的类型，也可以为空。
Date	是	Sun, 22 Nov 2015 08:16:38 GMT	此次操作的时间，Date必须为GMT格式且不能为空。  注意 如果请求中的Date时间和OSS服务器的当前时间差15分钟以上，OSS服务器将拒绝该请求，并返回HTTP 403错误。

参数	是否必选	示例值	说明
CanonicalizedOSSHeaders	否	x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n	以x-oss-为前缀的HTTP Header的字典序排列，可以为空。 - 如果设置CanonicalizedOSSHeaders为空，则无需在最后添加分隔符 <code>\n</code>。 - 如果只有一个CanonicalizedOSSHeaders，则需要最后添加分隔符 <code>\n</code>，例如 <code>x-oss-meta-a:a\n</code>。 - 如果有多个CanonicalizedOSSHeaders，则需要每一个CanonicalizedOSSHeaders之后添加分隔符 <code>\n</code>，例如 <code>x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n</code>。 关于CanonicalizedOSSHeaders的构建方法，请参见构建CanonicalizedOSSHeaders的方法。
CanonicalizedResource	是	/examplebucket/	用户想要访问的OSS资源，不能为空。 关于CanonicalizedResource的构建方法，请参见构建CanonicalizedResource的方法。

● 签名示例

请求	签名字符串计算公式	签名字符串
PUT /nelson HTTP/1.0 Content-MD5: eB5eJF1ptWaXm4bjSPyxw== 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-meta-magic: abracadabra	Signature = base64(hmac-sha1(AccessKeySecret,VERB + "\n" + Content-MD5 + "\n" + Content-Type + "\n" + Date + "\n" + CanonicalizedOSSHeaders+ CanonicalizedResource))	"PUT\n eB5eJF1ptWaXm4bjSPyxw==\n text/html\n Thu, 17 Nov 2005 18:49:58 GMT\n x-oss-meta-magic: abracadabra\n x-oss-meta-author: foo@bar.com\n /oss-example/nelson

假设AccessKey ID为 “44CF959*****252F707” ， AccessKey Secret 为 “OtxrzxIsfpFjA7Sw*****8Bw21TLhquhboDYROV” ，此处以Python为例介绍计算签名（Signature）的方法。

```
import hmac
import hashlib
h = hmac.new(oss2.to_bytes("OtxrxIsfpFjA7Sw*****8Bw21TLhquhboDYROV"),
             oss2.to_bytes("PUT\nODBGOERFMDMzQTczRUY3NUE3NzA5QzdFNUYzMDQxNEM=\ntext/html\nThu,
17 Nov 2005 18:49:58 GMT\nx-oss-meta-magic:abracadabra\nx-oss-meta-author:foo@bar.com\n/oss-exa
mple/nelson"), hashlib.sha1)
signature = oss2.utils.b64encode_as_string(h.digest())
print("Signature: %s" % signature)
```

签名（Signature）计算结果应该为26NBxoKd*****Dv6inkoDft/yA=，由于Authorization = “OSS” + AccessKeyId + “:” + Signature，所以最后Authorization为 “OSS 44CF95900***BF252F707:26NBxoKd*****Dv6inkoDft/yA=”，然后加上Authorization头组成最后需要发送的消息。

```
PUT /nelson HTTP/1.0
Authorization:OSS 44CF95900***BF252F707:26NBxoKd*****Dv6inkoDft/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-meta-magic: abracadabra
```

细节分析如下：

- 如果传入的AccessKey ID不存在或未激活，则返回403 Forbidden错误，错误码为InvalidAccessKeyId；如果AccessKey ID已激活，但OSS判断用户的请求发生签名错误，则返回403 Forbidden错误，并在返回的response中显示正确的用于验证加密的签名字符串。用户可以根据OSS的response来检查自己的签名字符串是否正确。

返回示例如下：

```
<?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>
    1E446260FF9B****
  </RequestId>
  <HostId>
    oss-cn-hangzhou.aliyuncs.***
  </HostId>
  <SignatureProvided>
    y5H7yzPsA/tP4+0tH1HHvPEwUv8=
  </SignatureProvided>
  <StringToSign>
    GET
    Wed, 11 May 2011 07:59:25 GMT
    /oss-example?acl
  </StringToSign>
  <OSSAccessKeyId>
    AKIAIVAKMSMOY7VO****
  </OSSAccessKeyId>
</Error>
```

- 如果用户请求头中Authorization值的格式不对，则返回400 Bad Request错误，错误码为InvalidArgument。
- OSS所有的请求都必须使用HTTP 1.1协议规定的GMT时间格式。其中日期的格式为：

```
date1 = 2DIGIT SP month SP 4DIGIT; day month year (e.g., 02 Jun 1982)
```

 说明 上述日期格式中“day”所占位数都是“2DIGIT”，因此“Jun 2”、“2 Jun 1982”和“2-Jun-1982”均是非法日期格式。

- 如果签名验证时，头中没有传入Date或者Date格式不正确，则返回403 Forbidden错误，错误码为AccessDenied。
- 传入请求的时间必须在OSS服务器当前时间之后的15分钟以内，否则返回403 Forbidden错误，错误码为RequestTimeTooSkewed。

构建CanonicalizedOSSHeaders的方法

所有以x-oss-为前缀的HTTP Header被称为CanonicalizedOSSHeaders，构建方法如下：

1. 将所有以x-oss-为前缀的HTTP请求头的名称转换为小写的形式，例如 X-OSS-Meta-Name: TaoBao 转换为 x-oss-meta-name: TaoBao 。

2. 如果以从STS服务获取的临时访问凭证发送请求时，您还需要将获得的security-token值以 `x-oss-security-token:security-token` 的形式加入到签名字符串中。

 **说明** 关于搭建STS服务的具体操作，请参见开发指南中的[使用STS临时访问凭证访问OSS](#)。您可以通过调用STS服务的[AssumeRole](#)接口或者使用[各语言STS SDK](#)来获取临时访问凭证。临时访问凭证包括临时访问密钥（AccessKeyId和AccessKeySecret）和安全令牌（SecurityToken）。

3. 将步骤1中得到的所有HTTP请求头按照名称的字典序进行升序排列。
4. 删除请求头和内容之间分隔符两端出现的任何空格。例如 `x-oss-meta-name: TaoBao` 转换为 `x-oss-meta-name:TaoBao`。
5. 将每一个请求头和内容使用分隔符 `\n` 分隔拼成CanonicalizedOSSHeaders。

构建CanonicalizedResource的方法

用户发送请求中想访问的OSS目标资源被称为CanonicalizedResource，构建方法如下：

1. 将CanonicalizedResource置为空字符串 `""`。
2. 设置要访问的OSS资源，格式为 `/BucketName/ObjectName`。
 - 如果仅有Bucket Name而没有Object Name，则CanonicalizedResource为 `"/BucketName/"`。
 - 如果既没有Bucket Name也没有Object Name，则CanonicalizedResource为 `"/"`。
 - 如果请求的资源包括子资源（SubResource），则将所有的子资源按照字典序，从小到大排列并以 `&` 为分隔符生成子资源字符串。在CanonicalizedResource字符串尾添加 `?` 和子资源字符串。此时的CanonicalizedResource为 `/BucketName/ObjectName?acl&uploadId=UploadId`。

OSS目前支持的子资源（SubResource）包括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、response-content-encoding等。

子资源（SubResource）有以下三种类型：

- 资源标识，例如子资源中的acl、append、uploadId、symlink等。更多信息，请参见[关于Bucket的操作](#)和[关于Object的操作](#)。
- 指定返回Header字段，例如 `response-***`。更多信息，请参见[Get Object](#)的 `Request Parameters`。
- 文件（Object）处理方式，例如 `x-oss-process`。更多信息，请参见[图片处理](#)。

计算签名头规则

- 签名的字符串必须为 UTF-8 格式。含有中文字符的签名字符串必须先进行 UTF-8 编码，再与 AccessKey Secret 计算最终签名。
- 签名的方法用[RFC 2104](#)中定义的HMAC-SHA1方法，其中Key指的是AccessKey Secret。
- Content-Type 和 Content-MD5 在请求中不是必须的，如果请求需要签名验证，空值请以换行符 `\n` 代替。
- 在所有非HTTP标准定义的Header中，只有以 `x-oss-` 开头的Header需要加入签名字符串（例如签名示例中的x-oss-meta-magic则需要加入签名字符串）；其他非HTTP标准Header将被OSS忽略。

- ❓ 说明 以 `x-oss-` 开头的Header在签名验证前需要符合以下规范：
- Header的名称需要变为小写。
 - Header按字典序从小到大排序。
 - 分割header name和value的冒号前后不能有空格。
 - 每个Header之后都要有一个换行符 “\n”，如果没有Header，则设置CanonicalizedOSSHeaders为空。

Content-MD5的计算方法

以消息内容“123456789”为例，以下详细说明正确及错误计算该字符串的Content-MD5的方法。

- 正确计算示例
 - i. 先计算MD5加密的二进制数组（128位）。
 - ii. 对该二进制数组进行base64编码（而不是对32位字符串编码）。

以Python为例：

```
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789") //在Python 3中此处需要改为hash.update(b"0123456789")。
>>> base64.b64encode(hash.digest())
'bE5eJF1ptWaXm4bijSPyxw=='
```

`hash.digest()`，计算出二进制数组（128位）。

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

- 错误计算示例

❓ 说明 常见错误是直接对计算出的32位字符串进行base64编码。

```
# hash.hexdigest(), 计算得到可见的32位字符串编码。
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
# 错误的MD5值进行base64编码后的结果。
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVIMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

4.3. 在URL中包含签名

除了使用Authorization Header，您还可以在URL中加入签名信息，以便将该URL转给第三方实现授权访问。

注意事项

- 使用在URL中签名的方式，会将授权的数据在过期时间内暴露在互联网上，请预先评估使用风险。
- OSS不支持同时在URL和Header中包含签名。
- PUT和GET请求都支持在URL中签名。
- 您可以为PUT操作生成一个预签名的URL，该URL检查用户是否上传了正确内容。SDK对请求进行预签名

时，将计算请求正文的校验和，并生成包含在预签名URL中的MD5校验和。用户必须上传与SDK生成的MD5校验和相同的内容，否则操作失败。如果要验证MD5，只需在请求中增加Content-MD5头即可。

实现方式

• URL签名示例

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****
```

如果需要使用STS用户构造URL签名，则必须携带 security-token ，URL签名示例如下：

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?OSSAccessKeyId=nz2pc56s936****&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv****&security-token=SecurityToken
```

• 参数

名称	是否必选	描述
Expires	是	一个Unix时间戳（自UTC时间1970年01月01号开始的秒数），用于标识该URL的超时时间。如果OSS接收到该URL请求的时间晚于签名中包含的Expires参数时，则返回请求超时的错误码。例如当前时间是1141889060，开发者希望创建一个60秒后自动失效的URL，则可以设置Expires时间为1141889120。  说明 出于安全考虑，OSS控制台中默认URL的有效时间为3600秒，最大值为32400秒。
OSSAccessKeyId	是	密钥中的AccessKeyId。

名称	是否必选	描述
Signature	是	签名信息。 签名信息的格式如下： <pre>Signature = urlencode(base64(hmac-sha1(AccessKeySecret, VERB + "\n" + CONTENT-MD5 + "\n" + CONTENT-TYPE + "\n" + EXPIRES + "\n" + CanonicalizedOSSHeaders + CanonicalizedResource))))</pre> 所有OSS支持的请求和各种Header参数，在URL中进行签名的算法和在Header中包含签名的算法基本一样。 生成URL中的签名字符串时，除了将Date参数替换为Expires参数外，仍然包含 <code>CONTENT-TYPE</code>、<code>CONTENT-MD5</code>、<code>CanonicalizedOSSHeaders</code> 等在Header中包含签名中定义的Header（请求中虽然仍有Date请求Header，但无需将Date加入签名字符串中）。 在URL中包含签名时必须对URL进行urlencode。如果在URL中多次传入Signature、Expires或OSSAccessKeyId，则以第一次传入的值为准。 使用URL签名时，OSS会先验证请求时间是否晚于Expires时间，然后再验证签名。
security-token	否	安全令牌。只有当使用STS用户构造URL签名时，才需要设置此参数。  说明 关于搭建STS服务的具体操作，请参见开发指南中的使用STS临时访问凭证访问OSS。您可以通过调用STS服务的AssumeRole接口或者使用各语言STS SDK来获取临时访问凭证。临时访问凭证包括临时访问密钥（AccessKeyId和AccessKeySecret）和安全令牌（SecurityToken）。

- 使用SDK生成URL签名
- URL中添加签名的Python示例代码：

```
import base64
import hmac
import hashlib
import urllib
h = hmac.new("OtxrxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
            "GET\n\n1141889120\n/oss-example/oss-api.pdf",
            hashlib.sha1)
urllib.quote (base64.encodestring(h.digest()).strip())
```

OSS SDK的URL签名实现，请参见下表。

SDK	URL签名方法	实现文件
Java SDK	OSSClient.generatePresignedUrl	OSSClient.java
Python SDK	Bucket.sign_url	api.py
.Net SDK	OssClient.GeneratePresignedUri	OssClient.cs
PHP SDK	OssClient.signUrl	OssClient.php
JavaScript SDK	signatureUrl	Object.js
C SDK	oss_gen_signed_url	oss_object.c
C++ SDK	OssClient::GeneratePresignedUrl	OssClient.cc

SDK

通过签名URL上传或下载文件的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)
- [Browser.js](#)
- [Android](#)
- [iOS](#)

错误码

错误码	返回消息	描述
AccessDenied	403 Forbidden	在URL中添加签名时，Signature、Expires和OSSAccessKeyId顺序可以调换，但缺少Signature、Expires或OSSAccessKeyId中的一个或者多个。

错误码	返回消息	描述
AccessDenied	403 Forbidden	访问的当前时间晚于请求中设定的Expires时间或时间格式错误。
InvalidArgument	400 Bad Request	URL中包含Signature、Expires、OSSAccessKeyId中的一个或者多个，并且Header中也包含签名消息。

4.4. Bucket权限控制

对象存储OSS提供Bucket级别的权限访问控制。

Bucket目前有三种访问权限：public-read-write，public-read和private，它们的含义如下：

权限值	中文名称	权限对访问者的限制
public-read-write	公共读写	任何人（包括匿名访问者）都可以对该存储空间内文件进行读写操作。  警告 互联网上任何用户都可以对该 Bucket 内的文件进行访问，并且向该 Bucket 写入数据。这有可能造成您数据的外泄以及费用激增，若被人恶意写入违法信息还可能会侵害您的合法权益。除特殊场景外，不建议您配置公共读写权限。
public-read	公共读，私有写	只有该存储空间的拥有者可以对该存储空间内的文件进行写操作，任何人（包括匿名访问者）都可以对该存储空间中的文件进行读操作。  警告 互联网上任何用户都可以对该 Bucket 内文件进行访问，这有可能造成您数据的外泄以及费用激增，请谨慎操作。
private	私有读写	只有该存储空间的拥有者可以对该存储空间内的文件进行读写操作，其他人无法访问该存储空间内的文件。

说明

- 用户创建一个Bucket时，如果不指定Bucket权限，OSS会自动为该Bucket设置private权限。
- 对于private资源，需要获得授权才能访问。关于授权管理，可参考[权限控制](#)。
- 对于一个已经存在的Bucket，只有它的创建者可以通过OSS的 Put Bucket Acl接口修改该Bucket的权限。

5.关于Service操作

5.1. GetService (ListBuckets)

调用GetService (ListBuckets)接口列举请求者拥有的所有存储空间（Bucket）。您还可以通过设置prefix、marker或者max-keys参数列举满足指定条件的存储空间。

注意事项

要列举存储空间，您必须有 `oss:GetService (ListBuckets)` 权限。具体操作，请参见[为RAM用户授权自定义的RAM Policy](#)。

请求语法

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

请求参数

名称	类型	是否必选	示例值	描述
prefix	字符串	否	my	限定返回的Bucket名称必须以prefix作为前缀。如果不设定，则不过滤前缀信息。 默认值：无
marker	字符串	否	mybucket10	设定结果从marker之后按字母排序的第一个开始返回。如果不设定，则从头开始返回数据。 默认值：无
max-keys	Integer	否	10	限定此次返回Bucket的最大个数。 取值范围：1~1000 默认值：100

响应元素

 **说明** 调用GetService (ListBuckets)接口时，如果所有Bucket已返回，则返回的XML中不包含Prefix、Marker、MaxKeys、IsTruncated和NextMarker响应元素。

名称	类型	示例值	描述
ListAllMyBucketsResult	容器	不涉及	保存Get Service请求结果的容器。 子节点：Owner、Buckets 父节点：None

名称	类型	示例值	描述
Prefix	字符串	my	本次查询结果的前缀。 父节点：ListAllMyBucketsResult
Marker	字符串	mybucket	表示本次GetService (ListBucket)的起点。 父节点：ListAllMyBucketsResult
MaxKeys	字符串	10	响应请求内返回结果的最大个数。 父节点：ListAllMyBucketsResult
IsTruncated	枚举字符串	true	是否所有的结果都已经返回。取值范围如下： <i>true</i>：表示本次没有返回全部结果。 <i>false</i>：表示本次已经返回了全部结果。 父节点：ListAllMyBucketsResult
NextMarker	字符串	mybucket10	用于继续查询时给marker赋值。表示下一次GetService (ListBucket)可以以此为marker，将未返回的结果返回。 父节点：ListAllMyBucketsResult
Owner	容器	不涉及	用于存放Bucket拥有者信息的容器。 父节点：ListAllMyBucketsResult
ID	字符串	ut_test_put_bucket	Bucket拥有者的用户ID。 父节点：ListAllMyBucketsResult.Owner
DisplayName	字符串	ut_test_put_bucket	Bucket拥有者的名称（目前和ID一致）。 父节点：ListAllMyBucketsResult.Owner
Buckets	容器	不涉及	保存多个Bucket信息的容器。 子节点：Bucket 父节点：ListAllMyBucketsResult
Bucket	容器	不涉及	保存Bucket信息的容器。 子节点：Name, CreationDate, Location 父节点：ListAllMyBucketsResult.Buckets
Name	字符串	mybucket01	Bucket名称。 父节点：ListAllMyBucketsResult.Buckets.Bucket

名称	类型	示例值	描述
CreateDate	时间	2014-05-15T11:18:32.000Z	Bucket创建时间。格式为yyyy-mm-ddThh:mm:ss.timezone。 父节点：ListAllMyBucketsResult.Buckets.Bucket
Location	字符串	oss-cn-hangzhou	Bucket所在的数据中心。 父节点：ListAllMyBucketsResult.Buckets.Bucket
ExtranetEndpoint	字符串	oss-cn-hangzhou.aliyuncs.com	Bucket访问的外网域名。 父节点：ListAllMyBucketsResult.Buckets.Bucket
IntranetEndpoint	字符串	oss-cn-hangzhou-internal.aliyuncs.com	同地域ECS访问Bucket的内网域名。 父节点：ListAllMyBucketsResult.Buckets.Bucket
Region	字符串	cn-hangzhou	Bucket所在地域。 父节点：ListAllMyBucketsResult.Buckets.Bucket
StorageClass	字符串	Standard	Bucket存储类型，支持Standard、IA、Archive和ColdArchive四种存储类型。 父节点：ListAllMyBucketsResult.Buckets.Bucket

示例

- 获取请求者拥有的所有Bucket

请求示例

```
GET / HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: oss-cn-hangzhou.aliyuncs.com
Authorization: OSS nxj7dtlhcy15hpvnhi: COS3OQkfQPnKmYZTEHYv2*****
```

返回示例


```
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: 5374A2880232A65C2300****
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
  <Owner>
    <ID>512**</ID>
    <DisplayName>51264</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <CreationDate>2014-02-17T18:12:43.000Z</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>
      <Region>cn-shanghai</Region>
      <StorageClass>Standard</StorageClass>
    </Bucket>
    <Bucket>
      <CreationDate>2014-02-25T11:21:04.000Z</CreationDate>
      <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
      <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</IntranetEndpoint>
      <Location>oss-cn-hangzhou</Location>
      <Name>mybucket</Name>
      <Region>cn-hangzhou</Region>
      <StorageClass>IA</StorageClass>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

- 根据指定前缀、最大返回个数等获取Bucket

请求示例

```
GET /?prefix=my&max-keys=10 HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: oss-cn-hangzhou.aliyuncs.com
Authorization: OSS nxj7dtwhcyl5hpnvni: COS3OQkfQPnKmYZTEHYv2****
```

返回示例

```
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: 5374A2880232A65C2300****
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
  <Prefix>my</Prefix>
  <Marker>mybucket</Marker>
  <MaxKeys>10</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <NextMarker>mybucket10</NextMarker>
  <Owner>
    <ID>ut_test_put_bucket</ID>
    <DisplayName>ut_test_put_bucket</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <CreationDate>2014-05-14T11:18:32.000Z</CreationDate>
      <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
      <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</IntranetEndpoint>
      <Location>oss-cn-hangzhou</Location>
      <Name>mybucket01</Name>
      <Region>cn-hangzhou</Region>
      <StorageClass>Standard</StorageClass>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [iOS](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403	请求中没有用户验证信息（即匿名访问）。

6.关于Bucket操作

6.1. 基础操作

6.1.1. PutBucket

调用Put Bucket接口创建存储空间（Bucket）。

注意事项

- 要创建存储空间，您必须有oss:PutBucket权限。具体操作，请参见[RAM Policy常见示例](#)。
- 同一阿里云账号在同一地域（Region）内最多支持创建100个Bucket。
- 每个地域都有对应的访问域名（Endpoint）。关于地域与访问域名对应关系的更多信息，请参见[访问域名和数据中心](#)。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求语法

```
PUT / HTTP/1.1
Host: bucket.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>
```

请求参数

请求Header参数

参数名称	类型	是否必选	示例	说明
x-oss-acl	BucketACL	否		指定Bucket的访问权限ACL。 关于Bucket访问权限ACL的更多信息，请参见 设置存储空间访问权限ACL 。

请求Body参数

参数名称	类型	是否必选	示例	说明
------	----	------	----	----

参数名称	类型	是否必选	示例	说明
bucket	String	是	oss-example	Bucket名称。存储空间的命名规范如下： - 只能包括小写字母、数字和短划线（-）。 - 必须以小写字母或者数字开头和结尾。 - 长度必须在3~63字符之间。
createBucketConfiguration	CreateBucketConfiguration	否		Bucket配置信息。

响应体语法

响应参数

无响应参数

示例

请求示例

```
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:77Dvh5wQgljWjwO/KyRt8dOP****
<?xml version="1.0" encoding="UTF-8"?>
<CreateBucketConfiguration>
  <StorageClass>Standard</StorageClass>
</CreateBucketConfiguration>
```

正常返回示例

JSON 格式

```
HTTP/1.1 200 OK
```

返回示例补充说明

```
...
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 24 Feb 2017 03:15:40 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
...
```

错误码

访问[错误中心](#)查看更多错误码。

开发者资源

- [SDK](#)

阿里云为您提供多种语言的SDK，帮助您快速通过API集成阿里云的产品和服务，推荐您使用SDK调用API，已免除您手动签名验证。

- [OpenAPI Explorer](#)

快速检索，可视化调试API，在线命令行工具，同步动态生成可执行的SDK代码示例。

- [阿里云CLI](#)

阿里云资产管理和配置工具，可通过命令方式同时管理多个阿里云产品和服务，简单快捷，是您上云好帮手。

SDK

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
InvalidBucketName	400	创建Bucket时，定义的Bucket名称不符合命名规范。

错误码	HTTP状态码	描述
AccessDenied	403	返回该错误的可能原因如下： · 发起PutBucket请求时没有传入用户验证信息。 · 没有操作权限。
TooManyBuckets	400	创建的Bucket数量超过上限。同一阿里云账号在同一地域（Region）内最多可创建100个Bucket。
BucketAlreadyExists	409	该Bucket已存在或被其他用户占用。请尝试使用其他符合命名规范的Bucket名称新建Bucket。

6.1.2. DeleteBucket

调用DeleteBucket删除某个存储空间（Bucket）。

说明

- 只有Bucket的拥有者才有权限删除该Bucket。
- 为了防止误删除的发生，OSS不允许删除一个非空的Bucket。

请求语法

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

示例

- 正常删除的请求示例

```
DELETE / HTTP/1.1
Host: test.oss-cn-hangzhou.aliyuncs.com
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: */*
Connection: keep-alive
date: Tue, 15 Jan 2019 08:19:04 GMT
authorization: OSS qn6qrrqx02oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3****
Content-Length: 0
```

返回示例

```
HTTP/1.1 204 No Content
Server: AliyunOSS
Date: Tue, 15 Jan 2019 08:19:04 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5C3D9778CC1C2AEDF85B****
x-oss-server-time: 190
```

- 删除的Bucket不存在的请求示例

```
DELETE / HTTP/1.1
Host: test.oss-cn-hangzhou.aliyuncs.com
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: */*
Connection: keep-alive
date: Tue, 15 Jan 2019 07:53:24 GMT
authorization: OSS qn6qrrqx02oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3****
Content-Length: 0
```

返回示例

```
HTTP/1.1 404 Not Found
Server: AliyunOSS
Date: Tue, 15 Jan 2019 07:53:25 GMT
Content-Type: application/xml
Content-Length: 288
Connection: keep-alive
x-oss-request-id: 5C3D9175B6FC201293AD****
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchBucket</Code>
  <Message>The specified bucket does not exist.</Message>
  <RequestId>5C3D9175B6FC201293AD****</RequestId>
  <HostId>test.oss-cn-hangzhou.aliyuncs.com</HostId>
  <BucketName>test</BucketName>
</Error>
```

- 删除的Bucket非空的请求示例

```
DELETE / HTTP/1.1
Host: test.oss-cn-hangzhou.aliyuncs.com
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: */*
Connection: keep-alive
date: Tue, 15 Jan 2019 07:35:06 GMT
authorization: OSS qn6qrrqx02oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3****
Content-Length: 0
```

返回示例


```
HTTP/1.1 409 Conflict
Server: AliyunOSS
Date: Tue, 15 Jan 2019 07:35:06 GMT
Content-Type: application/xml
Content-Length: 296
Connection: keep-alive
x-oss-request-id: 5C3D8D2A0ACA54D87B43****
x-oss-server-time: 16
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>BucketNotEmpty</Code>
  <Message>The bucket you tried to delete is not empty.</Message>
  <RequestId>5C3D8D2A0ACA54D87B43****</RequestId>
  <HostId>test.oss-cn-hangzhou.aliyuncs.com</HostId>
  <BucketName>test</BucketName>
</Error>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403 Forbidden	没有删除该Bucket的权限。只有Bucket的拥有者才能删除该Bucket。

6.1.3. GetBucket (ListObjects)

GetBucket (List Objects)接口用于列举存储空间（Bucket）中所有文件（Object）的信息。

注意事项

调用该接口时，有如下注意事项：

- GetBucket (List Objects)接口已修订为GetBucketV2 (List ObjectsV2)。建议您在开发应用程序时使用较新的版本GetBucketV2 (List ObjectsV2)。为保证向后兼容性，OSS继续支持GetBucket (List Objects)。有

关于GetBucketV2 (List ObjectsV2)的更多信息，请参见[GetBucketV2 \(List ObjectsV2\)](#)。

- 执行GetBucket (List Objects)请求时不会返回Object中自定义的元信息。

请求语法

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

请求头

此接口仅涉及公共请求头，例如 `Authorization`、`Host` 等更多信息，请参见[公共请求头 \(Common Request Headers\)](#)。

请求参数

名称	类型	是否必选	示例值	描述
delimiter	字符串	否	/	对Object名字进行分组的字符。所有Object名字包含指定的前缀，第一次出现delimiter字符之间的Object作为一组元素（即CommonPrefixes）。 默认值：无
marker	字符串	否	test1.txt	设定从marker之后按字母排序开始返回Object。 marker用来实现分页显示效果，参数的长度必须小于1024字节。 做条件查询时，即使marker在列表中不存在，也会从符合marker字母排序的下一个开始打印。 默认值：无
max-keys	字符串	否	200	指定返回Object的最大数。 如果因为max-keys的设定无法一次完成列举，返回结果会附加NextMarker元素作为下一次列举的marker。 取值：大于0小于1000 默认值：100

名称	类型	是否必选	示例值	描述
prefix	字符串	否	fun	限定返回文件的Key必须以prefix作为前缀。 - prefix参数的长度必须小于1024字节。 - 使用prefix查询时，返回的Key中仍会包含prefix。 如果把prefix设为某个文件夹名，则列举以此Prefix开头的文件，即该文件夹下递归的所有文件和子文件夹。 在设置prefix的基础上，将delimiter设置为正斜线（/）时，返回值中只列举该文件夹下的文件，文件夹下的子文件夹名返回在CommonPrefixes中，子文件夹下递归的所有文件和文件夹不显示。 例如，一个Bucket中有三个Object，分别为fun/test.jpg、fun/movie/001.avi和fun/movie/007.avi。如果设定prefix为fun/，则返回三个Object；如果在prefix设置为fun/的基础上，将delimiter设置为正斜线（/），则返回fun/test.jpg和fun/movie/。 默认值：无
encoding-type	字符串	否	URL	对返回的内容进行编码并指定编码的类型。 默认值：无 可选值：URL  注意 delimiter、marker、prefix、NextMarker以及Key使用UTF-8字符。如果delimiter、marker、prefix、NextMarker以及Key中包含XML 1.0标准不支持的控制字符，您可以通过指定encoding-type对返回结果中的Delimiter、Marker、Prefix、NextMarker以及Key进行编码。

响应元素

名称	类型	示例值	描述
Contents	容器	不涉及	保存每个返回Object元信息的容器。 父节点：ListBucketResult
CommonPrefixes	字符串	不涉及	如果请求中指定了Delimiter参数，则会在返回的响应中包含CommonPrefixes元素。该元素表明以Delimiter结尾，并有共同前缀的Object名称的集合。 父节点：ListBucketResult
Delimiter	字符串	/	对Object名字进行分组的字符。所有名字包含指定的前缀且第一次出现Delimiter字符之间的Object作为一组元素CommonPrefixes。 父节点：ListBucketResult

名称	类型	示例值	描述
EncodingType	字符串	不涉及	指明了返回结果中编码使用的类型。如果请求的参数中指定了encoding-type, 则会对返回结果中的Delimiter、Marker、Prefix、NextMarker和Key这些元素进行编码。 父节点: ListBucketResult
DisplayName	字符串	user_example	Object拥有者名称。 父节点: ListBucketResult.Contents.Owner
ETag	字符串	5B3C1A2E053D763E1B002CC607C5A0FE1****	ETag (Entity Tag) 在每个Object生成时创建, 用于标识一个Object的内容。 - 对于PutObject请求创建的Object, ETag值是其内容的MD5值。 - 对于其他方式创建的Object, ETag值是基于一定计算规则生成的唯一值, 但不是其内容的MD5值。 - ETag值可以用于检查Object内容是否发生变化。不建议使用ETag值作为Object内容的MD5校验数据完整性的依据。 父节点: ListBucketResult.Contents
ID	字符串	0022012****	Bucket拥有者的用户ID。 父节点: ListBucketResult.Contents.Owner
IsTruncated	枚举字符串	false	请求中返回的结果是否被截断。 返回值: true、false - true表示本次没有返回全部结果。 - false表示本次已经返回了全部结果。 父节点: ListBucketResult
Key	字符串	fun/test.jpg	Object的Key。 父节点: ListBucketResult.Contents
LastModified	时间	2012-02-24T08:42:32.000Z	Object最后被修改的时间。 父节点: ListBucketResult.Contents
ListBucketResult	容器	不涉及	保存GetBucket请求结果的容器。 子节点: Name、Prefix、Marker、MaxKeys、Delimiter、IsTruncated、Nextmarker、Contents 父节点: None
Marker	字符串	test1.txt	标识此次GetBucket (ListObjects) 的起点。 父节点: ListBucketResult

名称	类型	示例值	描述
MaxKeys	字符串	100	响应请求内返回结果的最大数目。 父节点：ListBucketResult
Name	字符串	oss-example	Bucket名称。 父节点：ListBucketResult
Owner	容器	不涉及	保存Bucket拥有者信息的容器。 子节点：DisplayName、ID 父节点：ListBucketResult
Prefix	字符串	fun/	本次查询结果的前缀。 父节点：ListBucketResult
Size	字符串	344606	返回Object大小，单位为字节。 父节点：ListBucketResult.Contents
StorageClass	字符串	Standard	Object的存储类型。 父节点：ListBucketResult.Contents

此接口涉及的其他公共响应头，例如 `x-oss-request-id`、`Content-Type` 等的更多信息，请参见[公共响应头 \(Common Response Headers\)](#)。

示例

- 简单请求示例

```
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/ZghT7cGa0ykbo****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
  <Type>Normal</Type>
  <Size>344606</Size>
  <StorageClass>Standard</StorageClass>
  <Owner>
    <ID>0022012****</ID>
    <DisplayName>user-example</DisplayName>
  </Owner>
</Contents>
<Contents>
  <Key>fun/movie/007.avi</Key>
  <LastModified>2012-02-24T08:43:27.000Z</LastModified>
  <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
  <Type>Normal</Type>
  <Size>344606</Size>
  <StorageClass>Standard</StorageClass>
  <Owner>
    <ID>0022012****</ID>
    <DisplayName>user-example</DisplayName>
  </Owner>
</Contents>
<Contents>
  <Key>fun/test.jpg</Key>
  <LastModified>2012-02-24T08:42:32.000Z</LastModified>
  <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
  <Type>Normal</Type>
  <Size>344606</Size>
  <StorageClass>Standard</StorageClass>
  <Owner>
    <ID>0022012****</ID>
    <DisplayName>user-example</DisplayName>
  </Owner>
</Contents>
<Contents>
  <Key>oss.jpg</Key>
  <LastModified>2012-02-24T06:07:48.000Z</LastModified>
  <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
  <Type>Normal</Type>
  <Size>344606</Size>
  <StorageClass>Standard</StorageClass>
  <Owner>
    <ID>0022012****</ID>
    <DisplayName>user-example</DisplayName>
  </Owner>
</Contents>
</ListBucketResult>

```

- 带prefix参数的请求示例

```
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/ZghT7cGa0ykbo****
```

返回示例

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/movie/007.avi</Key>
    <LastModified>2012-02-24T08:43:27.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>fun/test.jpg</Key>
    <LastModified>2012-02-24T08:42:32.000Z</LastModified>
    <ETag>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>

```

- 带prefix和delimiter参数的请求示例


```
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:DNrnX7xHk3sgysx7l8U9l9lY****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>"5B3C1A2E053D763E1B002CC607C5A0FE1****"</ETag>
    <Type>Normal</Type>
    <Size>344606</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>0022012****</ID>
      <DisplayName>user_example</DisplayName>
    </Owner>
  </Contents>
  <CommonPrefixes>
    <Prefix>fun/movie/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

- 指定marker分页列举的请求示例

此示例中指定了max-keys=2，即最多返回2条列举信息。

```
GET /?max-keys=2&marker=test1.txt HTTP/1.1
Host: test-bucket-xxx.oss-cn-shenzhen.aliyuncs.com
Accept-Encoding: identity
Accept: */*
Connection: keep-alive
User-Agent: aliyun-sdk-python/2.11.0(Darwin/18.2.0/x86_64;3.4.1)
date: Tue, 26 May 2020 08:39:48 GMT
authorization: OSS LTAIB1VW9xxxxxxx:MmY11jLlO8UlaqjqHK3Ckp****
```

返回示例

结果中的Next Marker可以做为下次列举请求的Marker。

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 26 May 2020 08:39:48 GMT
Content-Type: application/xml
Content-Length: 1032
Connection: keep-alive
x-oss-request-id: 5ECCD5D4881816373582xxx
x-oss-server-time: 3
<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult>
  <Name>test-bucket-xxx</Name>
  <Prefix></Prefix>
  <Marker>test1.txt</Marker>
  <MaxKeys>2</MaxKeys>
  <Delimiter></Delimiter>
  <EncodingType>url</EncodingType>
  <IsTruncated>true</IsTruncated>
  <NextMarker>test100.txt</NextMarker>
  <Contents>
    <Key>test10.txt</Key>
    <LastModified>2020-05-26T07:50:18.000Z</LastModified>
    <ETag>"C4CA4238A0B923820DCC509A6F75*****"</ETag>
    <Type>Normal</Type>
    <Size>1</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>1305433xxx</ID>
      <DisplayName>1305433xxx</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>test100.txt</Key>
    <LastModified>2020-05-26T07:50:20.000Z</LastModified>
    <ETag>"C4CA4238A0B923820DCC509A6F75*****"</ETag>
    <Type>Normal</Type>
    <Size>1</Size>
    <StorageClass>Standard</StorageClass>
    <Owner>
      <ID>1305433xxx</ID>
      <DisplayName>1305433xxx</DisplayName>
    </Owner>
  </Contents>
</ListBucketResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)

- C
- .NET
- Android
- Node.js
- Browser.js
- Ruby

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有访问该Bucket的权限。
InvalidArgument	400	• max-keys小于0或者大于1000。 • prefix、marker、delimiter参数的长度不符合要求。

6.1.4. GetBucketV2 (ListObjectsV2)

GetBucketV2 (List ObjectsV2)接口用于列举存储空间（Bucket）中所有文件（Object）的信息。

 说明

执行GetBucketV2 (List ObjectsV2)请求时不会返回Object中自定义的元信息。

请求语法

```
GET /?list-type=2 HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
```

请求参数

名称	类型	是否必选	描述
List-type	数字	是	取值只能为2。
Delimiter	字符串	否	对Object名字进行分组的字符。所有Object名字包含指定的前缀，第一次出现Delimiter字符之间的Object作为一组元素（即CommonPrefixes）。 默认值：无

名称	类型	是否必选	描述
Start-after	字符串	否	设定从Start-after之后按字母排序开始返回Object。 Start-after用来实现分页显示效果，参数的长度必须小于1024字节。 做条件查询时，即使Start-after在列表中不存在，也会从符合Start-after字母排序的下一个开始打印。 默认值：无
Continuation-token	字符串	否	指定list操作需要从此token开始。您可从ListObjectsV2结果中的NextContinuationToken获取此token。
Max-keys	字符串	否	指定返回Object的最大数。 取值：大于0小于1000 默认值：100  说明 如果因为Max-keys的设定无法一次完成列举，返回结果会附加一个 <NextContinuationToken> 作为下一次列举的Continuation-token。
Prefix	字符串	否	限定返回文件的Key必须以Prefix作为前缀。 如果把Prefix设为某个文件夹名，则列举以此Prefix开头的文件，即该文件夹下递归的所有文件和子文件夹。 在设置Prefix的基础上，将Delimiter设置为正斜线（/）时，返回值就只列举该文件夹下的文件，文件夹下的子文件夹名返回在CommonPrefixes中，子文件夹下递归的所有文件和文件夹不显示。 例如，一个Bucket中有三个Object，分别为fun/test.jpg、fun/movie/001.avi和fun/movie/007.avi。若设定Prefix为fun/，则返回三个Object；如果在Prefix设置为fun/的基础上，将Delimiter设置为正斜线（/），则返回fun/test.jpg和fun/movie/。  说明 - 参数的长度必须小于1024字节。 - 使用Prefix查询时，返回的Key中仍会包含Prefix。 默认值：无

名称	类型	是否必选	描述
Encoding-type	字符串	否	对返回的内容进行编码并指定编码的类型。 默认值：无 可选值：url  说明 Delimiter、Start-after、Prefix、NextContinuationToken以及Key使用UTF-8字符。如果Delimiter、Start-after、Prefix、NextContinuationToken以及Key中包含XML 1.0标准不支持的控制字符，您可以通过指定Encoding-type对返回结果中的Delimiter、Start-after、Prefix、NextContinuationToken以及Key进行编码。
Fetch-owner	布尔值	否	指定是否在返回结果中包含owner信息。 合法值：<i>true</i>、<i>false</i> • <i>true</i>：表示返回结果中包含owner信息。 • <i>false</i>：表示返回结果中不包含owner信息。 默认值：<i>false</i>

响应元素

名称	类型	描述
Contents	容器	保存每个返回Object元信息的容器。 父节点：ListBucketResult
CommonPrefixes	字符串	如果请求中指定了Delimiter参数，则会在返回的响应中包含CommonPrefixes元素。该元素表明以Delimiter结尾，并有共同前缀的Object名称的集合。 父节点：ListBucketResult
Delimiter	字符串	对Object名字进行分组的字符。所有名字包含指定的前缀且第一次出现Delimiter字符之间的Object作为一组元素CommonPrefixes。 父节点：ListBucketResult
EncodingType	字符串	指明返回结果中编码使用的类型。如果请求的参数中指定了Encoding-type，则会对返回结果中的Delimiter、StartAfter、Prefix、NextContinuationToken和Key这些元素进行编码。 父节点：ListBucketResult
DisplayName	字符串	Object拥有者名称。 父节点：ListBucketResult.Contents.Owner

名称	类型	描述
ETag	字符串	ETag在每个Object生成时创建，用于标识一个Object的内容。 父节点：ListBucketResult.Contents - 对于Put Object请求创建的Object，ETag值是其内容的MD5值。 - 对于其他方式创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。 - ETag值可以用于检查Object内容是否发生变化。不建议使用ETag值作为Object内容的MD5校验数据完整性的依据。
ID	字符串	Bucket拥有者的用户ID。 父节点：ListBucketResult.Contents.Owner
IsTruncated	枚举字符串	请求中返回的结果是否被截断。 返回值：<i>true</i>、<i>false</i> <i>true</i>表示本次没有返回全部结果。 <i>false</i>表示本次已经返回了全部结果。 父节点：ListBucketResult
Key	字符串	Object的Key。 父节点：ListBucketResult.Contents
LastModified	时间	Object最后被修改的时间。 父节点：ListBucketResult.Contents
ListBucketResult	容器	保存GetBucket请求结果的容器。 子节点：Name、Prefix、StartAfter、MaxKeys、Delimiter、IsTruncated、NextContinuationToken、Contents 父节点：None
StartAfter	字符串	如果请求中指定了StartAfter参数，则会在返回的响应中包含StartAfter元素。
MaxKeys	字符串	响应请求内返回结果的最大数目。 父节点：ListBucketResult
Name	字符串	Bucket名称。 父节点：ListBucketResult
Owner	容器	保存Bucket拥有者信息的容器。 子节点：DisplayName、ID 父节点：ListBucketResult

名称	类型	描述
Prefix	字符串	本次查询结果的前缀。 父节点：ListBucketResult
Size	字符串	Object的字节数。 父节点：ListBucketResult.Contents
StorageClass	字符串	Object的存储类型。 父节点：ListBucketResult.Contents
ContinuationToken	字符串	如果请求中指定了ContinuationToken参数，则会在返回的响应中包含ContinuationToken元素。
KeyCount	数字	此次请求返回的Key的个数。如果指定了Delimiter，则KeyCount为Key和CommonPrefixes的元素之和。
NextContinuationToken	字符串	表明此次GetBucketV2 (ListObjectsV2)请求包含后续结果，需要将NextContinuationToken指定为ContinuationToken继续获取结果。

示例

● 简单请求示例

```
GET /?list-type=2 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/ZghT7cGa0ykbo****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>
  <MaxKeys>100</MaxKeys>
  <EncodingType>url</EncodingType>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>a</Key>
    <LastModified>2020-05-18T05:45:43.000Z</LastModified>
    <ETag>"35A27C2B9EAE6B6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
```

```

<Contents>
  <Key>a/b</Key>
  <LastModified>2020-05-18T05:45:47.000Z</LastModified>
  <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
  <Size>25</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
  <Key>b</Key>
  <LastModified>2020-05-18T05:45:50.000Z</LastModified>
  <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
  <Size>25</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
  <Key>b/c</Key>
  <LastModified>2020-05-18T05:45:54.000Z</LastModified>
  <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
  <Size>25</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
  <Key>bc</Key>
  <LastModified>2020-05-18T05:45:59.000Z</LastModified>
  <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
  <Size>25</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<Contents>
  <Key>c</Key>
  <LastModified>2020-05-18T05:45:57.000Z</LastModified>
  <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
  <Size>25</Size>
  <StorageClass>STANDARD</StorageClass>
</Contents>
<KeyCount>6</KeyCount>
</ListBucketResult>

```

- 带Prefix参数的请求示例

```

GET /?list-type=2&prefix=a 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/ZghT7cGa0ykbo****

```

返回示例


```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>a</Prefix>
  <MaxKeys>100</MaxKeys>
  <EncodingType>url</EncodingType>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>a</Key>
    <LastModified>2020-05-18T05:45:43.000Z</LastModified>
    <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>a/b</Key>
    <LastModified>2020-05-18T05:45:47.000Z</LastModified>
    <ETag>"35A27C2B9EAEEB6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <KeyCount>2</KeyCount>
</ListBucketResult>
```

- 带Prefix和Delimiter参数的请求示例

```
GET /?list-type=2&prefix=a/&delimiter=/ HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:DNrn7xHk3sgysx7I8U9I9IY****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>a/</Prefix>
  <MaxKeys>100</MaxKeys>
  <Delimiter>/</Delimiter>
  <EncodingType>url</EncodingType>
  <IsTruncated>>false</IsTruncated>
  <Contents>
    <Key>a/b</Key>
    <LastModified>2020-05-18T05:45:47.000Z</LastModified>
    <ETag>"35A27C2B9EAE6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>a/b/</Prefix>
  </CommonPrefixes>
  <KeyCount>2</KeyCount>
</ListBucketResult>
```

- 带StartAfter、MaxKeys和FetchOwner参数的请求示例

```
GET /?list-type=2&start-after=b&max=keys=3&fetch-owner=true HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:DNrn7xHk3sgysx7I8U9I9IY****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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></Prefix>
  <StartAfter>b</StartAfter>
  <MaxKeys>3</MaxKeys>
  <EncodingType>url</EncodingType>
  <IsTruncated>true</IsTruncated>
  <NextContinuationToken>CgJiYw--</NextContinuationToken>
  <Contents>
    <Key>b/c</Key>
    <LastModified>2020-05-18T05:45:54.000Z</LastModified>
    <ETag>"35A27C2B9EAE6B6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1686240967192623</ID>
      <DisplayName>1686240967192623</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>ba</Key>
    <LastModified>2020-05-18T11:17:58.000Z</LastModified>
    <ETag>"35A27C2B9EAE6B6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1686240967192623</ID>
      <DisplayName>1686240967192623</DisplayName>
    </Owner>
  </Contents>
  <Contents>
    <Key>bc</Key>
    <LastModified>2020-05-18T05:45:59.000Z</LastModified>
    <ETag>"35A27C2B9EAE6B6F48FD7FB5861D****"</ETag>
    <Size>25</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1686240967192623</ID>
      <DisplayName>1686240967192623</DisplayName>
    </Owner>
  </Contents>
  <KeyCount>3</KeyCount>
</ListBucketResult>
```

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有访问该Bucket的权限。
InvalidArgument	400	- Max-keys小于0或者大于1000。 - Prefix、Start-after、Delimiter参数的长度不符合要求。

6.1.5. GetBucketInfo

调用GetBucketInfo接口查看存储空间（Bucket）的相关信息。

注意事项

 **说明** 该请求可以从任何一个OSS的Endpoint发起。

要获取存储空间的相关信息，您必须有 `oss:GetBucketInfo` 权限。具体操作，请参见[为RAM用户授权自定义的RAM Policy](#)。

请求语法

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

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

响应元素

名称	类型	示例值	描述
BucketInfo	容器	不涉及	保存Bucket信息的容器。 子节点：Bucket 父节点：无
Bucket	容器	不涉及	保存Bucket信息的容器。 父节点：BucketInfo

名称	类型	示例值	描述
CreationDate	时间	2013-07-31T10:56:21.000Z	Bucket的创建时间。 时间格式：2013-07-31T10:56:21.000Z 父节点：BucketInfo.Bucket
ExtranetEndpoint	字符串	oss-cn-hangzhou.aliyuncs.com	Bucket的外网域名。 父节点：BucketInfo.Bucket
IntranetEndpoint	字符串	oss-cn-hangzhou-internal.aliyuncs.com	同地域ECS访问Bucket的内网域名。 父节点：BucketInfo.Bucket
Location	字符串	oss-cn-hangzhou	Bucket所在的地域。 父节点：BucketInfo.Bucket
StorageClass	字符串	Standard	Bucket的存储类型。 有效值： <i>Standard</i> 、 <i>IA</i> 、 <i>Archive</i> 和 <i>ColdArchive</i> 关于存储类型的更多信息，请参见 存储类型介绍 。
Name	字符串	oss-example	Bucket名称。 父节点：BucketInfo.Bucket
Owner	容器	不涉及	存放Bucket所有者信息的容器。 父节点：BucketInfo.Bucket
ID	字符串	27183473914****	Bucket拥有者的用户ID。 父节点：BucketInfo.Bucket.Owner
DisplayName	字符串	username	Bucket拥有者的名称（目前和用户ID一致）。 父节点：BucketInfo.Bucket.Owner
AccessControlList	容器	不涉及	Bucket读写权限（ACL）信息的容器。 关于Bucket ACL的更多信息，请参见 设置存储空间读写权限（ACL） 。 父节点：BucketInfo.Bucket

名称	类型	示例值	描述
Grant	枚举字符串	private	Bucket的ACL权限。 有效值: <i>private</i> 、 <i>public-read</i> 、 <i>public-read-write</i> 父节点: BucketInfo.Bucket.AccessControlList
DataRedundancyType	枚举字符串	LRS	Bucket的数据容灾类型。 有效值: <i>LRS</i> 、 <i>ZRS</i> 父节点: BucketInfo.Bucket
Versioning	字符串	Enabled	Bucket的版本控制状态。 有效值: <i>Enabled</i> 、 <i>Suspended</i> 关于版本控制状态的更多信息, 请参见 PutBucketVersioning 。 父节点: BucketInfo.Bucket
ServerSideEncryptionRule	容器	不涉及	服务器端加密方式的容器。 关于服务器端加密方式的更多信息, 请参见 服务器端加密 。 父节点: BucketInfo.Bucket
ApplyServerSideEncryptionByDefault	容器	不涉及	服务器端默认加密方式的容器。 父节点: BucketInfo.Bucket
SSEAlgorithm	字符串	KMS	显示服务器端默认加密方式。 有效值: <i>KMS</i> 、 <i>AES256</i> 父节点: BucketInfo.Bucket
KMSMasterKeyID	字符串	*****	显示当前使用的KMS密钥ID。仅当SSEAlgorithm为KMS, 且指定了密钥ID时返回取值。其他情况下, 返回为空。 父节点: BucketInfo.Bucket
CrossRegionReplication	字符串	Disabled	显示Bucket的跨区域复制状态。 有效值: <i>Enabled</i> 、 <i>Disabled</i> 父节点: BucketInfo.Bucket

名称	类型	示例值	描述
TransferAcceleration	字符串	Disabled	显示Bucket的传输加速状态。 有效值： <i>Enabled</i> 、 <i>Disabled</i> 父节点： BucketInfo.Bucket
AccessMonitor	字符串	Enabled	显示Bucket的访问跟踪状态。 有效值： <i>Enabled</i> 、 <i>Disabled</i> 父节点： BucketInfo.Bucket

示例

请求示例

```
Get /?bucketInfo HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Sat, 12 Sep 2015 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****: BuG4rRK+zNhH1AcF51NNHD39****
```

返回示例

- 成功获取Bucket信息的返回示例

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>
    <AccessMonitor>Enabled</AccessMonitor>
    <CreateDate>2013-07-31T10:56:21.000Z</CreateDate>
    <ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
    <IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</IntranetEndpoint>
    <Location>oss-cn-hangzhou</Location>
    <StorageClass>Standard</StorageClass>
    <TransferAcceleration>Disabled</TransferAcceleration>
    <CrossRegionReplication>Disabled</CrossRegionReplication>
    <Name>oss-example</Name>
    <Owner>
      <DisplayName>username</DisplayName>
      <ID>27183473914****</ID>
    </Owner>
    <AccessControlList>
      <Grant>private</Grant>
    </AccessControlList>
    <Comment>test</Comment>
  </Bucket>
</BucketInfo>
```

- 获取不存在的Bucket信息的返回示例

```
HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906****
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>568D547F31243C673BA1****</RequestId>
  <HostId>nosuchbucket.oss.aliyuncs.com</HostId>
  <BucketName>nosuchbucket</BucketName>
</Error>
```

- 获取没有权限访问的Bucket信息的返回示例


```
HTTP/1.1 403
x-oss-request-id: 534B371674E88A4D8906****
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>568D5566F2D0F89F5C0E****</RequestId>
  <HostId>test.oss.aliyuncs.com</HostId>
</Error>
```

SDK

此接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有查看该Bucket信息的权限。只有Bucket的拥有者才能查看Bucket的信息。

6.1.6. GetBucketLocation

GetBucketLocation接口用于查看存储空间（Bucket）的位置信息。只有Bucket的拥有者才能查看Bucket的位置信息。

请求语法

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

响应元素

名称	类型	描述
LocationConstraint	字符串	Bucket所在的地域。 有效值：<i>oss-cn-hangzhou</i>、<i>oss-cn-shanghai</i>、<i>oss-cn-qingdao</i>、<i>oss-cn-beijing</i>、<i>oss-cn-zhangjiakou</i>、<i>oss-cn-hongkong</i>、<i>oss-cn-shenzhen</i>、<i>oss-us-west-1</i>、<i>oss-us-east-1</i>、<i>oss-ap-southeast-1</i>等。  说明 Bucket所在地域与数据中心的对应关系参见访问域名与数据中心。

示例

请求示例

```
Get /?location HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
```

返回示例

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403	没有查看Bucket位置信息的权限。只有Bucket的拥有者才能查看Bucket的位置信息。

6.2. 合规保留策略（WORM）

6.2.1. InitiateBucketWorm

调用InitiateBucketWorm接口新建一条合规保留策略。

注意事项

对象存储OSS支持WORM（Write Once Read Many）特性，允许以不可删除、不可篡改的方式保存和使用数据。OSS允许针对存储空间（Bucket）设置基于时间的合规保留策略，保护周期为1天到70年。

- 当基于时间的合规保留策略创建24小时后未提交锁定，则该策略自动失效。当合规保留策略锁定后，您可以在Bucket中上传和读取文件（Object），但是在Object的保留时间到期之前，不允许删除Object及合规保留策略。Object的保留时间到期后，才可以删除Object。关于合规保留策略的更多信息，请参见[合规保留策略](#)。
- 同一个Bucket中，版本控制和合规保留策略无法同时配置。如果Bucket已开启版本控制功能，则无法再配置保留策略。关于版本控制功能更多信息，请参见[版本控制介绍](#)。

请求元素

名称	类型	是否必须	描述
InitiateWormConfiguration	容器	是	根节点 子节点：RetentionPeriodInDays
RetentionPeriodInDays	正整数	是	指定Object保留天数。

示例

- 请求示例

```
POST /?worm HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Length: 556
Content-Type: application/xml
Host: BucketName.oss.aliyuncs.com
Authorization: OSS nxj7dtlhcy15hp****:COS3OQkfQPnKmyZTEHYv2****
<InitiateWormConfiguration>
  <RetentionPeriodInDays>365</RetentionPeriodInDays>
</InitiateWormConfiguration>
```

- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5374A2880232A65C2300****
x-oss-worm-id: 1666E2CFB2B3418****
Date: Thu, 15 May 2014 11:18:32 GMT
```

6.2.2. AbortBucketWorm

AbortBucketWorm用于删除未锁定的合规保留策略。

注意事项

当基于时间的合规保留策略创建后，该策略默认处于InProgress状态，且该状态的有效期为24小时。在有效期24小时内，此策略对应的Bucket数据处于保护状态。

- 启动合规保留策略24小时内：若此策略未提交锁定，则Bucket所有者以及授权用户可以删除此策略；若该保留策略已提交锁定，则不允许删除此策略，且无法缩短策略保护周期，仅可以延长保护周期。
- 启动合规保留策略24小时后：若超过24小时该保留策略未提交锁定，则该策略自动失效。

若Bucket内有文件处于保护周期内，那么您将无法删除合规保留策略，同时也无法删除Bucket。当Bucket为空时，Bucket的所有者可以删除该Bucket，从而间接删除该Bucket的保留策略。

示例

- 请求示例

```
DELETE /?worm HTTP/1.1
Date: Mon, 3 Aug 2020 03:15:40 GMT
Host: BucketName.oss.aliyuncs.com
Authorization: OSS qn6qrrqx02oawuk53otf****:77Dvh5wQgljWjwO/KyRt8dOP****
```

- 返回示例

```
HTTP/1.1 204 NoContent
x-oss-request-id: 534B371674E88A4D8906****
Date: Mon, 3 Aug 2020 03:15:40 GMT
```

6.2.3. CompleteBucketWorm

CompleteBucketWorm用于锁定合规保留策略。

注意事项

当基于时间的合规保留策略创建后，此策略默认处于InProgress状态，且该状态的有效期为24小时。在有效期24小时内，此策略对应的Bucket数据处于保护状态。

- 启动合规保留策略24小时内：若此策略未提交锁定，则Bucket所有者以及授权用户可以删除此策略；若此保留策略已提交锁定，则不允许删除此策略，且无法缩短策略保护周期，仅可以延长保护周期。
- 启动合规保留策略24小时后：若超过24小时此保留策略未提交锁定，则此策略自动失效。

示例

- 请求示例

```
POST /?wormId=xxx HTTP/1.1
Date: GMT Date
Host: BucketName.oss.aliyuncs.com
Authorization: SignatureValue
```

- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: RequestId
Date: Date
Content-Type: application/xml
Content-Length: length
```

6.2.4. ExtendBucketWorm

ExtendBucketWorm用于延长已锁定的合规保留策略对应Bucket中Object的保留天数。

 **说明** 若指定用于延长Object保留天数对应的WORM ID不存在，则返回404。

注意事项

当基于时间的合规保留策略创建后，该策略默认处于InProgress状态，且该状态的有效期为24小时。在有效期24小时内，此策略对应的Bucket数据处于保护状态。

- 启动合规保留策略24小时内：若此策略未提交锁定，则Bucket所有者以及授权用户可以删除此策略；若该保留策略已提交锁定，则不允许删除此策略，且无法缩短策略保护周期，仅可以延长保护周期。
- 启动合规保留策略24小时后：若超过24小时该保留策略未提交锁定，则该策略自动失效。

若Bucket内有文件处于保护周期内，那么您将无法删除合规保留策略，同时也无法删除Bucket。当Bucket为空时，Bucket的所有者可以删除该Bucket，从而间接删除该Bucket的保留策略。

请求元素

名称	类型	是否必须	描述
ExtendWormConfiguration	容器	是	根节点 子节点：RetentionPeriodInDays
RetentionPeriodInDays	正整数	是	指定Object的保留天数。

示例

- 请求示例

```
POST /?wormId=xxx&wormExtend HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: BucketName.oss.aliyuncs.com
Authorization: OSS nxj7dtlhcy15hp****:COS3OQkfQPnKmyZTEHYv2****
<ExtendWormConfiguration>
  <RetentionPeriodInDays>366</RetentionPeriodInDays>
</ExtendWormConfiguration>
```

- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5374A2880232A65C2300****
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 556
```

6.2.5. GetBucketWorm

GetBucketWorm用于获取指定存储空间（Bucket）的合规保留策略信息。

 **说明** 若指定用来获取Bucket的合规保留策略信息对应的WORM ID不存在，则返回404。

注意事项

对象存储OSS支持WORM（Write Once Read Many）特性，允许您以不可删除、不可篡改的方式保存和使用数据。OSS允许针对Bucket设置基于时间的合规保留策略，保护周期为1天到70年。

当合规保留策略锁定后，您可以在Bucket中上传和读取文件（Object），但是在Object的保留时间到期之前，不允许删除Object及合规保留策略。Object的保留时间到期后，才可以删除Object。

请求头

此接口仅涉及公共请求头，详情请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头，详情请参见[公共响应头（Common Response Headers）](#)。

响应元素

名称	类型	描述
WormConfiguration	容器	根节点。 子节点：WormId、State、RetentionPeriodInDays、CreationDate
WormId	字符串	合规保留策略的ID。
State	字符串	合规保留策略所处的状态。 可选值： - InProgress：合规保留策略创建后，该策略默认处于“InProgress”状态，且该状态的有效期为24小时。 - Locked：合规保留策略处于锁定状态。
RetentionPeriodInDays	正整数	Object的指定保留天数。
CreationDate	字符串	合规保留策略的创建时间。

示例

- 请求示例

```
GET /?worm HTTP/1.1
Date: Fri, 16 Oct 2020 11:18:32 GMT
Host: BucketName.oss.aliyuncs.com
Authorization: SignatureValue
```

- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5374A2880232A65C2300****
Date: Fri, 16 Oct 2020 11:18:32 GMT
Content-Type: application/xml
Content-Length: length
<WormConfiguration>
  <WormId>1666E2CFB2B3418****</WormId>
  <State>Locked</State>
  <RetentionPeriodInDays>1</RetentionPeriodInDays>
  <CreationDate>2020-10-15T15:50:32</CreationDate>
</WormConfiguration>
```

6.3. 权限控制（ACL）

6.3.1. PutBucketAcl

PutBucketAcl接口用于设置或修改存储空间（Bucket）的访问权限（ACL）。

注意事项

使用PutBucketAcl接口时，有如下注意事项：

- 该接口的请求者需要对Bucket有写入ACL的权限。
- PutBucketAcl为覆盖语义，即新传入的ACL将覆盖原有的ACL。
- 如果指定要设置ACL的Bucket不存在，调用该接口时将新建Bucket。

请求语法

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

请求头

名称	类型	是否必选	描述
----	----	------	----

名称	类型	是否必选	描述
x-oss-acl	字符串	是	指定Bucket的访问权限ACL。 PutBucketAcl接口通过Put请求中的x-oss-acl请求头来设置访问权限，如果没有该请求头，则访问权限设置不生效。 取值：public-read-write、public-read、private。 - public-read-write（公共读写）：所有用户都有该Bucket内的文件的读写权限。请谨慎使用该访问权限。 - public-read（公共读）：Bucket的拥有者和授权用户有该Bucket内的文件的读写权限，其他用户只有该Bucket内的文件的读权限。请谨慎使用该访问权限。 - private：Bucket的拥有者和授权用户有该Bucket内的文件的读写权限，其他用户没有权限操作该Bucket内的文件。

此接口涉及的其他公共请求头，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头，请参见[公共响应头（Common Response Headers）](#)。

示例

请求示例

```
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 qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrT****
```

返回示例

- 正常返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 24 Feb 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- 设置读写权限无效的返回示例


```
HTTP/1.1 400 Bad Request
x-oss-request-id: 56594298207FB3044385****
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>5***9</RequestId>
  <HostId>***-test.example.com</HostId>
  <ArgumentName>x-oss-acl</ArgumentName>
  <ArgumentValue>error-acl</ArgumentValue>
</Error>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403	• 发起PutBucketAcl请求时，没有传入用户验证信息。 • 没有发起PutBucketAcl请求的权限。

6.3.2. GetBucketAcl

GetBucketAcl接口用于获取某个存储空间（Bucket）的访问权限（ACL）。只有Bucket的拥有者才能获取Bucket的访问权限。

请求语法

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

响应元素

名称	类型	描述
AccessControllist	容器	存储ACL信息的容器类 父节点：AccessControlPolicy
AccessControlPolicy	容器	保存GetBucketACL结果的容器 父节点：None
DisplayName	字符串	Bucket拥有者的名称（目前和用户ID一致） 父节点：AccessControlPolicy.Owner
Grant	枚举字符串	Bucket的ACL权限 有效值： <i>private</i> 、 <i>public-read</i> 、 <i>public-read-write</i> 父节点：AccessControlPolicy.AccessControllist
ID	字符串	Bucket拥有者的用户ID 父节点：AccessControlPolicy.Owner
Owner	容器	保存Bucket拥有者信息的容器 父节点：AccessControlPolicy

示例

请求示例

```
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+WwIfNm0Fmgb****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
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>0022012****</ID>
    <DisplayName>user_example</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read</Grant>
  </AccessControlList>
</AccessControlPolicy>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Android](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有操作权限。只有Bucket的拥有者才能获取Bucket的访问权限。

6.4. 生命周期（Lifecycle）

6.4.1. PutBucketLifecycle

调用PutBucketLifecycle接口为存储空间（Bucket）设置生命周期规则。生命周期规则开启后，OSS将按照规则中指定的过期时间，自动转换与规则相匹配文件（Object）的存储类型或将其删除。

注意事项

- 要为存储空间设置生命周期规则，您必须有 `oss:PutBucketLifecycle` 权限。具体操作，请参见[为RAM用户授权自定义的RAM Policy](#)。
- 如果未为Bucket设置过生命周期规则，则执行此操作会创建一个新的生命周期规则；如果已为Bucket设置了生命周期规则，则执行此操作会覆写已有的规则配置。
- `PutBucketLifecycle`为覆盖语义。当您需要追加生命周期规则时，请先调用`GetBucketLifecycle`接口获取当前生命周期规则配置，然后追加新的生命周期规则配置，最后调用`PutBucketLifecycle`接口更新生命周期规则配置。
- 支持对Object以及Part（以分片方式上传，但最后未提交的分片）设置过期时间。
- 支持基于last access time（最后一次访问时间）或者last modify time（最后一次修改时间）匹配规则。

请求语法

```
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>
    <Transition>
      <Days>Days</Days>
      <StorageClass>StorageClass</StorageClass>
    </Transition>
    <AbortMultipartUpload>
      <Days>Days</Days>
    </AbortMultipartUpload>
  </Rule>
</LifecycleConfiguration>
```

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求元素

名称	类型	是否必选	示例值	描述
LifecycleConfiguration	容器	是	不涉及	Lifecycle配置的容器，最多支持1000条规则。 子节点：Rule 父节点：无

名称	类型	是否必选	示例值	描述
Rule	容器	是	不涉及	生命周期规则的容器。 - 不支持为ColdArchive存储类型的Bucket创建转储规则。 - Object的过期时间必须大于转储为IA或Archive类型的时间。 子节点：ID、Prefix、Status、Expiration 父节点：LifecycleConfiguration
ID	字符串	否	rule1	标识规则的唯一ID，最多由255个字符组成。如果未指定或者指定该值为空时，OSS会自动生成一个唯一ID。 子节点：无 父节点：Rule
Prefix	字符串	是	tmp/	指定规则所适用的前缀（Prefix）。指定的Prefix不能重复。 - 如果指定了Prefix，则表示此规则仅适用于Bucket中与Prefix匹配的Object。 - 如果Prefix指定为空，则表示此规则适用于Bucket中的所有Object。 子节点：无 父节点：Rule
Status	字符串	是	Enabled	是否启用规则。取值范围如下： - Enabled：表示启用规则，OSS会定期执行该规则。 - Disabled：表示不启用规则，OSS会忽略该规则。 父节点：Rule
Expiration	容器	否	不涉及	指定Object的过期属性。对于已开启版本控制的Bucket，指定的过期时间只对Object当前版本生效。 子节点：Days、CreatedBeforeDate或ExpiredObjectDeleteMarker 父节点：Rule
Days	正整数	Days与CreatedBeforeDate互斥	1	指定生命周期规则在距离Object最后更新多少天后生效。 父节点：Expiration或AbortMultipartUpload

名称	类型	是否必选	示例值	描述
CreatedBeforeDate	字符串	CreatedBeforeDate与Days互斥	2002-10-11T00:00:00.000Z	指定一个日期，OSS会对最后更新时间早于该日期的数据执行生命周期规则。日期格式为 yyyy-MM-ddT00:00:00.000Z。 日期需要按照ISO8601标准表示并使用UTC时间，且必须为UTC的零点。 父节点：Expiration或者AbortMultipartUpload
ExpiredObjectDeleteMarker	字符串	否	true	指定是否自动移除过期删除标记。取值范围如下： • true：表示自动移除过期删除标记。当取值为true时，不支持指定Days或CreatedBeforeDate。 • false：表示不会自动移除过期删除标记。当取值为false时，必须指定Days或CreatedBeforeDate。 父节点：Expiration
Transition	容器	否	不涉及	指定Object在有效生命周期中，OSS何时转换Object的存储类型。 Standard Bucket中Standard Object可以转储为IA、Archive或ColdArchive存储类型，但转储Archive存储类型的时间必须比转储IA存储类型的时间长。例如Transition IA设置Days为30，Transition Archive设置Days必须大于30。 父节点：Rule 子节点：Days、CreatedBeforeDate和StorageClass  注意 Days或CreatedBeforeDate只能二选一。

名称	类型	是否必选	示例值	描述
StorageClass	字符串	如果存在父节点 Transition 或 Noncurrent VersionTransition, 则必选	IA	指定Object转储的存储类型。取值范围如下： - IA：低频访问 当基于last access time匹配规则时，Object转储的存储类型只能指定为IA。 - Archive：归档存储 - ColdArchive：冷归档存储  注意 IA Bucket中的Object可以转储为Archive或者ColdArchive存储类型，但不支持转储为Standard存储类型。 父节点：Transition
AbortMultipartUpload	容器	否	不涉及	指定未完成分片上传的过期属性。 子节点：Days或CreatedBeforeDate 父节点：Rule
Tag	容器	否	不涉及	指定规则所适用的对象标签，可设置多个。 父节点：Rule 子节点：Key、Value
Key	字符串	如果存在父节点Tag, 则必选	TagKey1	对象标签的键。 父节点：Tag
Value	字符串	如果存在父节点Tag, 则必选	TagValue1	对象标签的值。 父节点：Tag
NoncurrentVersionExpiration	容器	否	不涉及	指定Object非当前版本生命周期规则的过期属性。 子节点：NoncurrentDays
NoncurrentVersionTransition	容器	否	不涉及	在有效的生命周期规则中，OSS何时将指定Object的非当前版本转储为IA或者Archive存储类型。 Standard类型的Object转储为Archive类型的时间必须大于转储为IA类型的时间。 子节点：NoncurrentDays、StorageClass

名称	类型	是否必选	示例值	描述
NoncurrentDays	字符串	如果存在父节点 NoncurrentVersionExpiration 或 NoncurrentVersionTransition，则必选	5	指定生命周期规则在Object成为非当前版本多少天后生效。 父节点：NoncurrentVersionTransition、NoncurrentVersionExpiration
IsAccessTime	字符串	否	true	指定是否基于last access time匹配规则。取值范围如下： • true：采用Object的last access time匹配规则。当基于last access time匹配规则时，Object只支持从标准存储转储为低频存储。 • false（默认）：采用Object的last modify time（最后一次修改时间）匹配规则。 父节点：Transition或NoncurrentVersionTransition。
ReturnToStandardWhenVisit	字符串	否	false	指定Object转为低频存储后，再次访问时是否将Object转为标准存储。只有当IsAccessTime设置为true时才有效。取值范围如下： • true：Object由低频存储转为标准存储。 • false（默认）：Object仍为低频存储。 父节点：Transition或NoncurrentVersionTransition。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 为未开启版本控制的Bucket设置生命周期
请求示例

```
PUT /?lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 443
Date: Thu , 8 Jun 2017 13:08:38 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:PYbzsdWAIWAIMW8luk****
<?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>transit objects to IA after 30, to Archive 60, expire after 10 years</ID>
  <Prefix>data</Prefix>
  <Status>Enabled</Status>
  <Transition>
    <Days>30</Days>
    <StorageClass>IA</StorageClass>
  </Transition>
  <Transition>
    <Days>60</Days>
    <StorageClass>Archive</StorageClass>
  </Transition>
  <Expiration>
    <Days>3600</Days>
  </Expiration>
</Rule>
<Rule>
  <ID>transit objects to Archive after 60 days</ID>
  <Prefix>important</Prefix>
  <Status>Enabled</Status>
  <Transition>
    <Days>6</Days>
    <StorageClass>Archive</StorageClass>
  </Transition>
</Rule>
<Rule>
  <ID>delete created before date</ID>
  <Prefix>backup</Prefix>
  <Status>Enabled</Status>
  <Expiration>
    <CreatedBeforeDate>2017-01-01T00:00:00.000Z</CreatedBeforeDate>
  </Expiration>
  <AbortMultipartUpload>
    <CreatedBeforeDate>2017-01-01T00:00:00.000Z</CreatedBeforeDate>
  </AbortMultipartUpload>
</Rule>
<Rule>
  <ID>r1</ID>
  <Prefix>rule1</Prefix>
  <Tag><Key>xx</Key><Value>1</Value></Tag>
  <Tag><Key>yy</Key><Value>2</Value></Tag>
  <Status>Enabled</Status>
  <Expiration>
    <Days>30</Days>
  </Expiration>
</Rule>
<Rule>

```

```

<ID>r2</ID>
<Prefix>rule2</Prefix>
<Tag><Key>xx</Key><Value>1</Value></Tag>
<Status>Enabled</Status>
<Transition>
  <Days>60</Days>
  <StorageClass>Archive</StorageClass>
</Transition>
</Rule>
<Rule>
  <ID>mtime transition1</ID>
  <Prefix>logs1</Prefix>
  <Status>Enabled</Status>
  <Transition>
    <Days>30</Days>
    <StorageClass>IA</StorageClass>
  </Transition>
</Rule>
<Rule>
  <ID>mtime transition2</ID>
  <Prefix>logs2</Prefix>
  <Status>Enabled</Status>
  <Transition>
    <Days>30</Days>
    <StorageClass>IA</StorageClass>
    <IsAccessTime>false</IsAccessTime>
  </Transition>
</Rule>
<Rule>
  <ID>atime transition1</ID>
  <Prefix>logs3</Prefix>
  <Status>Enabled</Status>
  <Transition>
    <Days>30</Days>
    <StorageClass>IA</StorageClass>
    <IsAccessTime>true</IsAccessTime>
    <ReturnToStdWhenVisit>false</ReturnToStdWhenVisit>
  </Transition>
</Rule>
<Rule>
  <ID>atime transition2</ID>
  <Prefix>logs4</Prefix>
  <Status>Enabled</Status>
  <Transition>
    <Days>30</Days>
    <StorageClass>IA</StorageClass>
    <IsAccessTime>true</IsAccessTime>
    <ReturnToStdWhenVisit>true</ReturnToStdWhenVisit>
  </Transition>
</Rule>
</LifecycleConfiguration>

```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674A4D890****
Date: Thu, 8 Jun 2017 13:08:38 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- 为已开启版本控制的Bucket设置生命周期
请求示例

```
PUT /?lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 336
Date: Mon , 6 May 2019 15:23:20 GMT
Authorization: OSSWnjI3fg9fdv8fg4b****:Phuu8bBhS8dsff2a****
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>delete example</ID>
    <Prefix>logs</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
    </Expiration>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>5</NoncurrentDays>
    </NoncurrentVersionExpiration>
    <AbortMultipartUpload>
      <Days>1</Days>
    </AbortMultipartUpload>
  </Rule>
  <Rule>
    <ID>transit example</ID>
    <Prefix>data</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
    <NoncurrentVersionTransition>
      <NoncurrentDays>10</NoncurrentDays>
      <StorageClass>IA</StorageClass>
    </NoncurrentVersionTransition>
  </Rule>
  <Rule>
    <ID>atime transition3</ID>
    <Prefix>logs5</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>10</NoncurrentDays>
      <StorageClass>IA</StorageClass>
      <IsAccessTime>true</IsAccessTime>
      <ReturnToStdWhenVisit>false</ReturnToStdWhenVisit>
    </NoncurrentVersionTransition>
  </Rule>
</LifecycleConfiguration>
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 7D3435J59A9812B*****
Date: Mon, 6 May 2019 15:23:20 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
InvalidArgument	400	返回此错误的可能原因如下： • OSS支持Standard Bucket中Standard Objects转储为IA、Archive存储类型。Standard Bucket可以针对一个Object同时配置转储IA和Archive存储类型规则，但转储Archive存储类型必须晚于转储IA存储类型的时间。 • 指定Object的过期时间必须晚于转储为IA或者Archive存储类型的时间。
AccessDenied	403	没有操作权限。仅支持拥有 <code>oss:PutBucketLifecycle</code> 权限的用户设置生命周期规则。

6.4.2. GetBucketLifecycle

调用GetBucketLifecycle接口查看存储空间（Bucket）的生命周期规则（Lifecycle）。

注意事项

要查看存储空间的生命周期规则，您必须有 `oss:GetBucketLifecycle` 权限。具体操作，请参见[为RAM用户授权自定义的RAM Policy](#)。

请求语法

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

示例

请求示例

```
Get /?lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Mon, 14 Apr 2014 01:17:29 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3****
```

返回示例

- 基于last modify time设置生命周期规则的返回示例

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>logs1</Prefix>
    <Status>Enabled</Status>
    <Expiration>
      <Days>1</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>mtime transition1</ID>
    <Prefix>logs2</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>IA</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>mtime transition2</ID>
    <Prefix>logs3</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>IA</StorageClass>
      <IsAccessTime>false</IsAccessTime>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- 基于last access time设置生命周期规则的返回示例

② 说明 当Bucket存在基于last access time匹配规则的生命周期规则时，返回示例中会包含AtimeBase元素。该元素表示默认last access time的时间戳（从1970-01-01 00:00:00 UTC计算起的秒数），即为Bucket开启访问跟踪时间点的时间戳。时间戳的单位为秒。

```

HTTP/1.1 200
x-oss-request-id: ****
Date: Mon, 26 Jul 2021 01:17:29 GMT
Connection: keep-alive
Content-Length: length
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
  <Rule>
    <ID>atime transition1</ID>
    <Prefix>logs1</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>IA</StorageClass>
      <IsAccessTime>true</IsAccessTime>
      <ReturnToStdWhenVisit>false</ReturnToStdWhenVisit>
    </Transition>
    <AtimeBase>1631698332</AtimeBase>
  </Rule>
  <Rule>
    <ID>atime transition2</ID>
    <Prefix>logs2</Prefix>
    <Status>Enabled</Status>
    <NoncurrentVersionTransition>
      <NoncurrentDays>10</NoncurrentDays>
      <StorageClass>IA</StorageClass>
      <IsAccessTime>true</IsAccessTime>
      <ReturnToStdWhenVisit>false</ReturnToStdWhenVisit>
    </NoncurrentVersionTransition>
    <AtimeBase>1631698332</AtimeBase>
  </Rule>
</LifecycleConfiguration>

```

- 未设置生命周期规则的返回示例

```

HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906****
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>534B372974E88A4D8906****</RequestId>
  <HostId> BucketName.oss.example.com</HostId>
</Error>

```

SDK

此接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403 Forbidden	没有权限查看Bucket的生命周期规则。只有Bucket的拥有者才能查看Bucket的生命周期规则。
NoSuchBucket或NoSuchLifecycle	404 Not Found	Bucket不存在或没有配置生命周期规则。

6.4.3. DeleteBucketLifecycle

DeleteBucketLifecycle接口用于删除指定存储空间（Bucket）的生命周期规则。使用DeleteBucketLifecycle接口删除指定Bucket所有的生命周期规则后，该Bucket中的文件（Object）不会被自动删除。只有Bucket的拥有者才能删除该Bucket的生命周期规则。

请求语法

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

示例

请求示例

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

返回示例

 **说明** 如果删除不存在的Lifecycle，则返回204状态码。

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906****
Date: Mon, 14 Apr 2014 01:17:35 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有删除该Bucket生命周期规则的权限。只有Bucket的拥有者才可以删除Bucket的生命周期规则。

6.5. 传输加速（TransferAcceleration）

6.5.1. PutBucketTransferAcceleration

PutBucketTransferAcceleration接口用于为存储空间（Bucket）配置传输加速。开启传输加速后，可提升全球各地用户对OSS的访问速度，适用于远距离数据传输、GB或TB级大文件上传和下载的场景。

注意事项

- 只有Bucket拥有者以及被授予oss:PutBucketTransferAcceleration权限的RAM用户才能发起配置传输加速的请求。
- 开启传输加速后，Bucket会在保留默认Endpoint的基础上新增传输加速域名，但必须使用OSS的传输加速域名才会提升访问速度。
- 使用传输加速域名访问Bucket时，OSS会收取传输加速费用。详情请参见[传输加速费用](#)。

有关传输加速的更多信息，请参见开发指南的[传输加速](#)。

请求语法

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

请求参数

名称	类型	是否必选	示例值	描述
TransferAccelerationConfiguration	容器	是	不涉及	传输加速配置的容器。
Enabled	字符串	是	true	目标Bucket是否开启传输加速。取值如下： • true: 开启传输加速。 • false: 关闭传输加速。 注意 传输加速开启及关闭操作在30分钟内生效。

此接口涉及Authorization、Content-Length等其他公共请求头的更多信息，请参见公共请求头（Common Request Headers）。

响应头

此接口仅涉及x-oss-request-id、Date等公共响应头。有关公共响应头的更多信息，请参见公共响应头（Common Response Headers）。

示例

- 请求示例
为目标存储空间examplebucket开启传输加速的请求示例如下：

```
PUT /?transferAcceleration HTTP/1.1
Date: Fri, 30 Apr 2021 13:08:38 GMT
Content-Length: 443
Content-Type: application/xml
Host: examplebucket.oss.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otf****:PYbzsdWAIWAIMW8luk****
<TransferAccelerationConfiguration>
  <Enabled>true</Enabled>
</TransferAccelerationConfiguration>
```

- 返回示例

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674A4D890****
Date: Fri, 30 Apr 2021 13:08:38 GMT
Content-Length: 443
Connection: keep-alive
Server: AliyunOSS

```

错误码

错误码	HTTP状态码	描述
AccessDenied	404	没有操作权限。仅支持拥有oss:PutBucketTransferAcceleration权限的用户配置传输加速。
MalformedXML	400	请求的XML格式不合法。例如，请求字段<Enabled>设置为true或者false以外的非法值。

6.5.2. GetBucketTransferAcceleration

GetBucketTransferAcceleration接口用于获取目标存储空间（Bucket）的传输加速配置。

注意事项

- 只有Bucket拥有者以及被授予oss:GetBucketTransferAcceleration权限的RAM用户才能发起获取传输加速配置的请求。
- 如果Bucket未配置过传输加速，调用该接口时不返回加速配置状态。

有关传输加速的更多信息，请参见开发指南的[传输加速](#)。

请求语法

```

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

```

返回参数

名称	类型	示例值	描述
TransferAccelerationConfiguration	容器	不涉及	保存传输加速配置信息的容器。
Enabled	字符串	true	传输加速状态。取值范围如下： - true: 开启传输加速。 - false: 关闭传输加速。

此接口涉及的x-oss-request-id、Date等其他公共响应头的更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

● 请求示例

以下请求示例用于获取目标存储空间examplebucket的传输加速状态。

```
GET /?transferAcceleration HTTP/1.1
Date: Fri, 30 Apr 2021 13:08:38 GMT
Content-Length: 443
Content-Type: application/xml
Host: examplebucket.oss.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otf****:PYbzsdWAIWAIMW8luk****
```

● 返回示例

以下返回示例表明examplebucket已开启传输加速。

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 30 Apr 2021 13:08:38 GMT
<TransferAccelerationConfiguration>
  <Enabled>true</Enabled>
</TransferAccelerationConfiguration>
```

错误码

错误码	HTTP状态码	描述
NoSuchTransferAccelerationConfiguration	404	请求的目标Bucket未配置传输加速。

6.6. 版本控制（Versioning）

6.6.1. PutBucketVersioning

调用PutBucketVersioning设置指定存储空间（Bucket）的版本控制状态。

注意事项

- 要配置版本控制，您必须有PutBucketVersioning权限。
- Bucket包括未开启、开启（Enabled）或者暂停（Suspended）三种版本控制状态。默认情况下Bucket处于未开启版本控制状态。
- 在Bucket处于开启版本控制状态下，所有新添加的文件（Object）都将拥有唯一的版本ID，OSS将累积所添加Object的多个版本。
- 在Bucket处于暂停版本控制状态下，所有新添加Object的版本ID将为null，且OSS将不再为此状态下添加的Object累积更多的版本。

关于版本控制的更多信息，请参见[版本控制介绍](#)。

请求语法

```
PUT /?versioning HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

示例

- 开启版本控制请求示例

```
PUT /?versioning HTTP/1.1
Host: bucket-versioning.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:20:12 GMT
Authorization: OSS e7thre3jj5mlvqk:12ztpkaR8a74glGFzOaZZQe****
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC015CB7AEADE01700****
Date: Tue, 09 Apr 2019 02:20:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- 暂停版本控制请求示例

```
PUT /?versioning HTTP/1.1
Host: bucket-versioning.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:28:18 GMT
Authorization: OSS m2qa99e9tpkaehr:DWAZr2EkqDwFJNke1Nuaogn7****
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Suspended</Status>
</VersioningConfiguration>
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC0342B7AEADE01700****
Date: Tue, 09 Apr 2019 02:28:18 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403	无设置版本控制状态的操作权限。
InvalidArgument	400	无效的Versioning状态。Versioning状态只能设置为Enabled或Suspended。

6.6.2. GetBucketVersioning

GetBucketVersioning接口用于获取指定Bucket的版本控制状态。

请求语法

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

响应元素

名称	类型	描述
VersioningConfiguration	容器	保存版本控制状态的容器 子节点: Status 父节点: 无
Status	字符串	版本控制状态 父节点: VersioningConfiguration 有效值: • Enabled: 开启版本控制状态 • Suspended: 暂停版本控制状态

示例

请求示例

```
GET /?versioning HTTP/1.1
Host: bucket-versioning.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:28:18 GMT
Authorization: OSS pu4kyewh6qk6nut:2TeHBmWY36GwdwS0JFiRb+NV****
```

返回示例

- 已开启版本控制的返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC0342B7AEADE01700****
Date: Tue, 09 Apr 2019 02:28:18 GMT
Content-Length: 121
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

- 未曾开启版本控制的返回示例

如果该Bucket未曾开启版本控制状态，则XML中不会返回版本控制Status信息。

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC015CB7AEADE01700****
Date: Tue, 09 Apr 2019 02:20:12 GMT
Content-Length: 74
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<VersioningConfiguration xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com"/>
```

SDK

此接口所对应的各语言SDK如下：

- Java
- Python
- Go
- C++
- .NET
- Node.js

错误码

错误码	HTTP状态码	描述
AccessDenied	403	无权限查看Bucket的版本控制状态。 只有Bucket拥有者及授予了GetBucketVersioning权限的RAM用户才能查看Bucket的版本控制状态。

错误码	HTTP状态码	描述
NoSuchBucket	404	访问的Bucket不存在。

6.6.3. GetBucketVersions(ListObjectVersions)

GetBucketVersions(ListObjectVersions)接口用于列出Bucket中包括删除标记（Delete Marker）在内的所有Object的版本信息。

注意事项

对于已开启版本控制的Bucket，调用GetBucketVersions(ListObjectVersions)与GetBucket(ListObjects)接口的返回结果区别如下：

- GetBucket(ListObjects)接口仅返回Object的当前版本，且当前版本不能为删除标记。
 - GetBucketVersions(ListObjectVersions)接口返回Bucket中所有Object的所有版本。
- 不同Object之间按字母排序返回，同一个Object的不同版本则按从新到旧排序，与版本ID的字母序无关。

请求头

此接口仅涉及公共请求头，例如 `Authorization`、`Host` 等更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求参数

调用GetBucketVersions(ListObjectVersions)接口时，可通过设置prefix、key-marker、version-id-marker、delimiter和max-keys参数限制仅返回部分结果。

名称	类型	是否必选	示例值	描述
delimiter	字符串	否	/	对Object名字进行分组的字符。所有Object名字包含指定的前缀（prefix），第一次出现delimiter字符之间的Object作为一组元素（即CommonPrefixes）。 如果将prefix设为文件夹名称后，再把delimiter设置为正斜线（/），则只返回该文件夹下的文件，该文件夹下的子文件名在CommonPrefixes中返回，子文件夹下递归的文件和文件夹不显示。 默认值：无
key-marker	字符串	如果version-id-marker不为空，则key-marker不能为空	example	设定结果从key-marker之后按字母序开始返回，与version-id-marker组合使用。 参数的长度必须小于1024字节。 默认值：无

名称	类型	是否必选	示例值	描述
version-id-marker	字符串	否	CAEQMxiBgICbof2 D0BYilGRhZjgwMz JiMjA3MjQ0ODE5M WYxZDYwMzJlZjU 1****	设定结果从key-marker对象的version-id-marker之后按新旧版本排序开始返回。如果version-id-marker未设定，则默认从key-marker按字母序排序的下一个Key的第一个版本开始返回。 默认值：无 有效值：版本ID
max-keys	字符串	否	100	限定此次返回Object的最大个数。 如果因为max-keys的设定无法一次完成列举，返回结果会附加 <code>NextKeyMarker</code> 和 <code>NextVersionIdMarker</code> 作为下一次列举的marker。列举结果中包含 <code>NextKeyMarker</code> 和 <code>NextVersionIdMarker</code> 的值。 取值：大于0小于1000 默认值：100
prefix	字符串	否	fun	限定返回的Object Key必须以prefix作为前缀。 - prefix的长度必须小于1024字节。 - 使用prefix查询时，返回结果的Key仍会包含prefix。 如果将prefix设为某个文件夹名，则列举以此prefix开头的文件，即该文件夹下递归的所有文件和子文件夹。 默认值：无
encoding-type	字符串	否	URL	对返回的内容进行编码并指定编码类型。 默认值：无 可选值：URL  注意 delimiter、marker、prefix、NextMarker以及Key使用UTF-8字符。如果delimiter、marker、prefix、NextMarker以及Key中包含XML 1.0标准不支持的控制字符，您可以通过指定encoding-type对返回结果中的Delimiter、Marker、Prefix、NextMarker以及Key进行编码。

响应元素

名称	类型	示例值	描述
ListVersionsResult	容器	不涉及	保存GetBucketVersions请求结果的容器。 子节点：Name、Prefix、Marker、MaxKeys、Delimiter、IsTruncated、Nextmarker、Version、DeleteMarker 父节点：None
CommonPrefixes	字符串	不涉及	如果请求中指定了delimiter参数，则OSS返回的响应中包含CommonPrefixes元素。该元素标明以delimiter结尾，并有共同前缀的Object名称的集合。 父节点：ListVersionsResult
Delimiter	字符串	/	用于对Object名字进行分组的字符。所有名字包含指定的前缀且第一次出现Delimiter字符之间的Object作为一组元素CommonPrefixes。 父节点：ListVersionsResult
EncodingType	字符串	URL	指明返回结果中编码使用的类型。如果请求的参数中指定了encoding-type，则表示对返回结果中的Delimiter、Marker、Prefix、NextMarker和Key这些元素进行编码。 父节点：ListVersionsResult
IsTruncated	字符串	true	指明是否已返回所有结果。 - true：表示本次没有返回全部结果。 - false：表示本次已返回全部结果。 有效值：true、false 父节点：ListVersionsResult
KeyMarker	字符串	example	标识此次GetBucketVersions的起点Object。 父节点：ListVersionsResult
VersionIdMarker	字符串	CAEQMxiBgICbof2D0BYilGRhZjgwMzJiMjA3MjQ0ODE5MWYxZDYwMzJlZjU1****	与KeyMarker参数一同使用，以指定GetBucketVersions的起点。 父节点：ListVersionsResult
NextKeyMarker	字符串	test	如果本次没有返回全部结果，响应请求中将包含NextKeyMarker元素，用于标明接下来请求的key-marker。 父节点：ListVersionsResult

名称	类型	示例值	描述
NextVersionIdMarker	字符串	CAEQGBiBgIC_jq7P9xYiIDRiZWJkNjY2Y2Q4NDQ5ZTI5ZGE5ODIxMTlyZThl****	如果本次没有返回全部结果，响应请求中将包含NextVersionIdMarker元素，用于标明接下来请求的version-id-marker。 父节点：ListVersionsResult
MaxKeys	字符串	1000	响应请求内返回结果的最大数目。 父节点：ListVersionsResult
Name	字符串	examplebucket-1250000000	Bucket名称。 父节点：ListVersionsResult
Owner	容器	不涉及	保存Bucket拥有者信息的容器。 父节点：ListVersionsResult
Prefix	字符串	fun	本次查询结果的前缀。 父节点：ListVersionsResult
Version	容器	不涉及	保存除删除标记以外的Object版本的容器。 父节点：ListVersionsResult
DeleteMarker	容器	不涉及	保存删除标记的容器。 父节点：ListVersionsResult
ETag	字符串	250F8A0AE989679A22926A875F0A2****	每个Object生成时创建的ETag，用于标识Object的内容。 - 对于PutObject请求创建的Object，ETag值是其内容的MD5值。 - 对于其他方式创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。  说明 ETag值仅用于检查Object内容是否发生变化。不建议使用ETag值作为Object内容的MD5数据完整性校验的依据。 父节点：ListVersionsResult.Version
Key	字符串	example	Object的名称。 父节点：ListVersionsResult.Version、ListVersionsResult.DeleteMarker

名称	类型	示例值	描述
LastModified	时间	2019-04-09T07:27:28.000Z	Object最后被修改的时间。 父节点：ListVersionsResult.Version、ListVersionsResult.DeleteMarker
VersionId	字符串	CAEQMxiBgMDNoP2D0BYiIDE3MWUxNzgxDQxNT RiODI5OGYwZGMwNGY3MzZjN***	Object的版本ID。 父节点：ListVersionsResult.Version、ListVersionsResult.DeleteMarker
IsLatest	字符串	true	Object是否为当前版本。 取值： - true：Object为当前版本。 - false：Object为非当前版本。 父节点：ListVersionsResult.Version、ListVersionsResult.DeleteMarker
Size	字符串	93731	Object的字节数。 父节点：ListVersionsResult.Version、ListVersionsResult.DeleteMarker
StorageClass	字符串	Standard	Object的存储类型。 父节点：ListVersionsResult.Version、ListVersionsResult.DeleteMarker
DisplayName	字符串	12345125285864390	Object拥有者的名称。 父节点：ListVersionsResult.Version.Owner、ListVersionsResult.DeleteMarker.Owner
ID	字符串	1234512528586****	Bucket拥有者的用户ID。 父节点：ListVersionsResult.Version.Owner、ListVersionsResult.DeleteMarker.Owner

此接口涉及的其他公共响应头，例如 `x-oss-request-id`、`Content-Type` 等的更多信息，请参见公共响应头（Common Response Headers）。

示例

- 未启用版本控制

请求示例

```
GET /?versions HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 07:27:48 GMT
Authorization: OSS ami4tq0x76ov9cu:WfX4****+e7Rc0jawCsh7hIk****
```

返回示例

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1262
Connection: keep-alive
Date: Thu, Tue, 09 Apr 2019 07:27:48 GMT
Server: AliyunOSS
x-cos-request-id: 534B371674E88A4D8906****
<ListVersionsResult>
  <Name>examplebucket-1250000000</Name>
  <Prefix/>
  <KeyMarker/>
  <VersionIdMarker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Version>
    <Key>example-object-1.jpg</Key>
    <VersionId/>
    <IsLatest>true</IsLatest>
    <LastModified>2019-08-5T12:03:10.000Z</LastModified>
    <ETag>5B3C1A2E053D763E1B669CC607C5A0FE1****</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
  </Version>
  <Version>
    <Key>example-object-2.jpg</Key>
    <VersionId/>
    <IsLatest>true</IsLatest>
    <LastModified>2019-08-9T12:03:09.000Z</LastModified>
    <ETag>5B3C1A2E053D763E1B002CC607C5A0FE1****</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
  </Version>
  <Version>
    <Key>example-object-3.jpg</Key>
    <VersionId/>
    <IsLatest>true</IsLatest>
    <LastModified>2019-08-10T12:03:08.000Z</LastModified>
    <ETag>4B3F1A2E053D763E1B002CC607C5AGTRF****</ETag>
    <Size>20</Size>
    <StorageClass>STANDARD</StorageClass>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
  </Version>
```

```
</Version>
</ListVersionsResult>
```

- 已开启版本控制

假设在名为oss example的Bucket中有example和pic.jpg 2个Object，其中example这个Object有3个版本，从新到旧排序的版本ID依次为111222、000123（删除标记）、222333。pic.jpg这个Object仅有1个版本，其版本ID为232323。

列举文件时将key-marker设置为example，version-id-marker指定为111222，则按example 000123、example 222333、pic.jpg 232323的顺序返回3个版本。

请求示例：

```
GET /?versions&key-marker=example&version-id-marker=CAEQMxiBgICbof2D0BYilGRhZjgwMzJiMjA3MjQ0ODE5MWYxZDYwMzJlZjU1**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 07:27:48 GMT
Authorization: OSS ami4tq0x76o****:WfX4kLpx+e7Rc0jawCsh7hIk****
```

返回示例：

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC4974B7AEADE01700****
Date: Tue, 09 Apr 2019 07:27:48 GMT
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<ListVersionsResult xmlns="http://doc.oss-cn-hangzhou.aliyuncs.com" >
  <Name>oss-example</Name>
  <Prefix></Prefix>
  <KeyMarker>example</KeyMarker>
  <VersionIdMarker>CAEQMxiBgICbof2D0BYilGRhZjgwMzJiMjA3MjQ0ODE5MWYxZDYwMzJlZjU1****</VersionIdMarker>
  <MaxKeys>100</MaxKeys>
  <Delimiter></Delimiter>
  <IsTruncated>>false</IsTruncated>
  <DeleteMarker>
    <Key>example</Key>
    <VersionId>CAEQMxiBgICaof2D0BYilDjHmGE3N2M1YTl1NDQzOGY5NTkyNTl3MGYyMzJm****</VersionId>
    <IsLatest>>false</IsLatest>
    <LastModified>2019-04-09T07:27:28.000Z</LastModified>
    <Owner>
      <ID>1234512528586****</ID>
      <DisplayName>12345125285864390</DisplayName>
    </Owner>
  </DeleteMarker>
  <Version>
    <Key>example</Key>
    <VersionId>CAEQMxiBgMDNoP2D0BYilDE3MWUxNzgxZDQxNTRiODI5OGYwZGMwNGY3MzZj****</VersionId>
    <IsLatest>>false</IsLatest>
    <LastModified>2019-04-09T07:27:28.000Z</LastModified>
    <ETag>"250F8A0AE989679A22926A875F0A2****"</ETag>
```

```

<Type>Normal</Type>
<Size>93731</Size>
<StorageClass>Standard</StorageClass>
<Owner>
  <ID>1234512528586****</ID>
  <DisplayName>12345125285864390</DisplayName>
</Owner>
</Version>
<Version>
  <Key>pic.jpg</Key>
  <VersionId>CAEQMxiBgMCZov2D0BYiIDY4MDlIOTc2YmY5MjQxMzdiOGI3OTlhNTU0ODIx****</VersionId>
>
  <IsLatest>true</IsLatest>
  <LastModified>2019-04-09T07:27:28.000Z</LastModified>
  <ETag>"3663F7B0B9D3153F884C821E7CF4****"</ETag>
  <Type>Normal</Type>
  <Size>574768</Size>
  <StorageClass>Standard</StorageClass>
  <Owner>
    <ID>1234512528586****</ID>
    <DisplayName>12345125285864390</DisplayName>
  </Owner>
</Version>
</ListVersionsResult>

```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404	目标Bucket不存在，包括试图访问因命名不规范而无法创建的Bucket。
AccessDenied	403	无访问此Bucket的权限。
InvalidArgument	400	• max-keys小于0或者大于1000。 • prefix、marker、delimiter参数长度不符合要求。 • version-id-marker为无效的版本ID。

6.7. 数据复制（Replication）

6.7.1. PutBucketReplication

PutBucketReplication接口用于为存储空间（Bucket）指定数据复制规则。OSS支持跨区域复制（Cross-Region Replication）和同区域复制（Same-Region Replication）。

注意事项

数据复制是以异步（近实时）方式将源Bucket中的文件（Object）以及对Object的创建、更新和删除等操作自动复制到目标Bucket。使用数据复制时，有如下注意事项：

- 数据复制采用异步复制，数据复制到目标Bucket需要一定的时间，通常几分钟到几小时不等，具体取决于数据的大小。
- 源Bucket与目标Bucket的名称不能相同。
- 使用跨区域复制时，源Bucket与目标Bucket必须处于不同的数据中心；使用同地域复制时，源Bucket与目标Bucket必须处于相同的数据中心。

关于数据复制的更多信息，请分别参见[跨区域复制介绍](#)和[同区域复制介绍](#)。

请求语法

```
POST /?replication&comp=add 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"?>
<ReplicationConfiguration>
  <Rule>
    <PrefixSet>
      <Prefix>prefix_1</Prefix>
      <Prefix>prefix_2</Prefix>
    </PrefixSet>
    <Action>ALL,PUT</Action>
    <Destination>
      <Bucket>Target Bucket Name</Bucket>
      <Location>oss-cn-hangzhou</Location>
      <TransferType>oss_acc</TransferType>
    </Destination>
    <HistoricalObjectReplication>enabled or disabled</HistoricalObjectReplication>
  </Rule>
</ReplicationConfiguration>
```

请求元素

名称	类型	是否必须	描述
ReplicationConfiguration	容器	是	配置Bucket数据复制规则的容器。 父节点：无 子节点：Rule
Rule	容器	是	保存数据复制规则的容器。 父节点：ReplicationConfiguration 子节点：Destination、HistoricalObjectReplication和ID
ID	字符串	否	数据复制规则对应的ID。 此ID最大长度为255个字符，且在单个Bucket中全局唯一。 如果没有指定ID或者ID值为空时，OSS会为该数据复制规则生成唯一值。 父节点：Rule 子节点：无
PrefixSet	容器	否	保存前缀（Prefix）的容器。每条数据复制规则中，最多可指定10条Prefix。 父节点：Rule 子节点：Prefix
Prefix	字符串	否	设置待复制Object的Prefix。只有匹配该Prefix的Object才被复制到目标Bucket。 - Prefix最大长度为1023个字符。 - 如果配置了Prefix，则新写入的数据和历史数据的同步都会遵循Prefix指定的规则。 父节点：PrefixSet 子节点：无

名称	类型	是否必须	描述
Action	字符串	否	指定可以被复制到目标Bucket的操作。如果配置了Action，则新写入的数据和历史数据的同步都会遵循Action指定的复制操作。 Action允许以下操作类型，您可以指定一项或多项。 取值： - ALL（默认值）：表示PUT、DELETE、ABORT操作均会被同步到目标Bucket。 - PUT：表示被同步到目标Bucket的写入操作，包括PutObject、PostObject、AppendObject、CopyObject、PutObjectACL、InitiateMultipartUpload、UploadPart、UploadPartCopy、CompleteMultipartUpload。 父节点：Rule 子节点：无
Destination	容器	是	保存目标Bucket信息的容器。 父节点：Rule 子节点：Bucket和Location
Bucket	字符串	是	指定数据要复制到的目标Bucket。 父节点：Destination 子节点：无
Location	字符串	是	目标Bucket所处的地域。 父节点：Destination 子节点：无
TransferType	字符串	是	指定数据复制时使用的数据传输链路。 取值： - internal（默认值）：OSS默认传输链路。 - oss_acc：传输加速链路。只有创建跨区域复制规则时才能使用传输加速链路。 父节点：Destination 子节点：无

名称	类型	是否必须	描述
HistoricalObjectReplication	字符串	否	指定是否复制历史数据。即开启数据复制前，是否将源Bucket中的数据复制到目标Bucket。 取值： - enabled（默认值）：表示复制历史数据。 - disabled：表示不复制历史数据。即仅复制创建数据复制规则后新写入的数据。 父节点：Rule 子节点：无
SyncRole	字符串	否	授权OSS使用哪个角色来进行数据复制。如果指定使用SSE-KMS加密目标对象，则必须指定该元素。
SourceSelectionCriteria	容器	否	用于标识要复制的源对象的其他筛选条件的容器。当前OSS仅支持针对SSE-KMS加密的源对象指定筛选条件。
SseKmsEncryptedObjects	容器	否	用于筛选使用SSE-KMS加密对象的容器。如果在数据复制规则中指定了SourceSelectionCriteria，则必须指定该元素。
Status	字符串	否	指定OSS是否复制通过SSE-KMS加密创建的对象。取值范围如下： - Enabled：表示复制通过SSE-KMS加密创建的对象。 - Disabled：表示不复制通过SSE-KMS加密创建的对象。
EncryptionConfiguration	容器	否	目标对象加密配置。如果指定Status为Enabled，则必须指定该元素。
ReplicaKmsKeyID	字符串	否	指定SSE-KMS密钥ID。如果指定Status为Enabled，则必须指定该元素。

示例

- 请求示例

```
POST /?replication&comp=add HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Type: application/xml
Content-Length: 186
Date: Thu, 24 Sep 2015 15:39:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrTZ****

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
  <Rule>
    <ID>test_replication_1</ID>
    <PrefixSet>
      <Prefix>source_image</Prefix>
      <Prefix>video</Prefix>
    </PrefixSet>
    <Action>PUT</Action>
    <Destination>
      <Bucket>target-bucket</Bucket>
      <Location>oss-cn-beijing</Location>
      <TransferType>oss_acc</TransferType>
    </Destination>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
    <SyncRole>aliyunramrole</SyncRole>
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <EncryptionConfiguration>
      <ReplicaKmsKeyID>c4d49f85-ee30-426b-a5ed-95e9139d****</ReplicaKmsKeyID>
    </EncryptionConfiguration>
  </Rule>
</ReplicationConfiguration>
```

● 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Thu, 24 Sep 2015 15:39:12 GMT
Content-Length: 0
Connection: close
Server: AliyunOSS
```

错误码

错误码	状态码	描述
-----	-----	----

错误码	状态码	描述
InvalidTargetBucket	400 BadRequest	返回此错误的可能原因如下： - 目标Bucket名称与源Bucket名称相同。 - 目标Bucket不存在。 - 目标Bucket与源Bucket不属于同一个用户。
InvalidTargetLocation	400 BadRequest	目标Bucket所在的Location不是请求XML中指定的Location。
BucketReplicationAlreadyExist	400 BadRequest	源Bucket到目标Bucket已存在复制关系。 如果需要配置新的数据复制规则，请先删除已有的数据复制规则再配置。
BadReplicationLocation	400 BadRequest	选择的目的地数据中心不合法。 您可通过GetBucketReplicationLocation来获得合法的可复制到的数据中心。
NoReplicationLocation	400 BadRequest	使用跨区域复制时，源Bucket所在的数据中心没有与之配对的可实施跨区域复制的数据中心。 跨区域复制数据中心的配对关系可通过访问域名和数据中心查看。
TooManyReplicationRules	400 BadRequest	请求中配置的数据复制规则超过一条。 单次请求中只能配置一条数据复制规则。
TooManyIncomingReplication	400 BadRequest	为Bucket配置的数据复制规则个数已达到100条。请删除不再使用的数据复制规则后重试。 单个Bucket支持配置100条数据复制规则。如果业务需求超出数量限制，请提交工单联系我们。
TooManyOutgoingReplication	400 BadRequest	为Bucket配置的数据复制规则个数已达到100条。请删除不再使用的数据复制规则后重试。 单个Bucket支持配置100条数据复制规则。如果业务需求超出数量限制，请提交工单联系我们。

错误码	状态码	描述
-----	-----	----

MissingArgument	400 BadRequest	未指定数据传输链路。
InvalidArgument	400 BadRequest	不支持指定的数据传输链路。

6.7.2. GetBucketReplication

调用GetBucketReplication接口获取某个存储空间（Bucket）已设置的数据复制规则。

请求语法

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

响应元素

名称	类型	描述
ReplicationConfiguration	容器	Bucket复制规则的容器。 父节点：无 子节点：Rule
Rule	容器	保存复制规则的容器。 父节点：ReplicationConfiguration 子节点：Destination、HistoricalObjectReplication、Status和ID
ID	字符串	复制规则对应的ID。 父节点：Rule 子节点：无

名称	类型	描述
PrefixSet	容器	保存前缀（Prefix）的容器。每条复制规则中，最多可指定10个Prefix。 父节点：Rule 子节点：Prefix
Prefix	字符串	被复制到目标Bucket中Object的前缀（Prefix）。 父节点：PrefixSet 子节点：无
Action	字符串	表示被同步到目标Bucket的操作。 Action允许以下操作类型，您可以指定一项或多项。 - ALL（默认值）：表示PUT、DELETE、ABORT操作均会被同步到目标Bucket。 - PUT：表示被同步到目标Bucket的写入操作，包括PutObject、PostObject、AppendObject、CopyObject、PutObjectACL、InitiateMultipartUpload、UploadPart、UploadPartCopy、CompleteMultipartUpload。 父节点：Rule 子节点：无
Status	字符串	表示复制状态。 取值： - starting：设置数据复制规则后，OSS会为Bucket准备复制任务，此时的复制状态为starting。 - doing：当数据复制规则生效后，即数据处于同步状态时，此时的复制状态为doing。 - closing：删除数据复制规则后，OSS会自动完成清理工作，此时的复制状态为closing。 父节点：Rule 子节点：无
Destination	容器	保存目标Bucket信息的容器。 父节点：Rule 子节点：Bucket和Location

名称	类型	描述
Bucket	字符串	数据要复制到的目标Bucket。 父节点：Destination 子节点：无
Location	字符串	目标Bucket所处的Location。 父节点：Destination 子节点：无
TransferType	字符串	数据复制时使用的数据传输类型。仅当传输类型为oss_acc时，返回示例中才会包含此元素。 取值： - internal（默认值）：OSS默认传输链路。 - oss_acc：传输加速链路。只有创建跨区域复制规则时才能使用传输加速链路。
HistoricalObjectReplication	字符串	是否复制历史数据。即开启数据复制前，是否将源Bucket中的已有的数据复制到目标Bucket。 取值： - enabled（默认）：表示复制历史数据。 - disabled：表示不复制历史数据，即仅复制数据复制规则生效后新写入的数据。 父节点：Rule 子节点：无
SyncRole	字符串	跨区域复制时使用的角色。仅当使用SSE-KMS加密目标对象时，返回示例中才会包含此元素。

示例

● 请求示例

```
GET /?replication HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 24 Sep 2015 15:39:15 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAI4XZ+WwIfNm0Fmgb****
```

● 返回示例

```

HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Thu, 24 Sep 2015 15:39:15 GMT
Content-Length: 186
Content-Type: application/xml
Connection: close
Server: AliyunOSS

<?xml version="1.0" ?>
<ReplicationConfiguration>
  <Rule>
    <ID>test_replication_1</ID>
    <PrefixSet>
      <Prefix>source_image</Prefix>
      <Prefix>video</Prefix>
    </PrefixSet>
    <Action>PUT</Action>
    <Destination>
      <Bucket>target-bucket</Bucket>
      <Location>oss-cn-beijing</Location>
      <TransferType>oss_acc</TransferType>
    </Destination>
    <Status>doing</Status>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
    <SyncRole>aliyunramrole</SyncRole>
  </Rule>
</ReplicationConfiguration>

```

错误码

错误码	状态码	说明
NoSuchBucket	404 Not Found	请求的Bucket不存在。
NoSuchReplicationConfiguration	404 Not Found	请求的Bucket没有配置数据复制规则。

6.7.3. GetBucketReplicationLocation

GetBucketReplicationLocation接口用于获取可复制到的目标存储空间（Bucket）所在的地域。您可以根据返回结果决定将源Bucket的数据复制到哪个地域。

请求语法

```

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

```

响应元素

名称	类型	描述
ReplicationLocation	容器	可复制地域的容器。
Location	字符串	可复制到的目标Bucket所在的地域，例如 oss-cn-beijing。 父节点：ReplicationLocation 子节点：无  说明 如果有多个可复制到的目标地域，那么返回的结果中包含多个Location。如果没有可复制到的目标地域，则返回的Location为空。
LocationTransferTypeConstraint	容器	包含TransferType约束的Location信息容器。
LocationTransferType	容器	包含TransferType的Location信息容器。
TransferTypes	容器	传输类型容器。
Type	字符串	数据复制时使用的数据传输类型。 取值： • internal（默认值）：OSS默认传输链路。 • oss_acc：传输加速链路。只有创建跨区域复制规则时才能使用传输加速链路。

示例

● 请求示例

```
GET /?replicationLocation HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 24 Sep 2015 15:39:15 GMT
Authorization: OSS qn6qrrqx02oawuk53otf****:CTkuxpLai4XZ+Wwlfnm0Fmgb****
```

● 返回示例

 说明

仅当传输类型为oss_acc时，返回的XML示例中才会包含<TransferType>元素。

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Thu, 24 Sep 2015 15:39:15 GMT
Content-Length: 84
Content-Type: application/xml Connection: close
Server: AliyunOSS
```

```
<?xml version="1.0" ?>
<ReplicationLocation>
  <Location>oss-cn-beijing</Location>
  <Location>oss-cn-qingdao</Location>
  <Location>oss-cn-shenzhen</Location>
  <Location>oss-cn-hongkong</Location>
  <Location>oss-us-west-1</Location>
  <LocationTransferTypeConstraint>
    <LocationTransferType>
      <Location>oss-cn-hongkong</Location>
      <TransferTypes>
        <Type>oss_acc</Type>
      </TransferTypes>
    </LocationTransferType>
    <LocationTransferType>
      <Location>oss-us-west-1</Location>
      <TransferTypes>
        <Type>oss_acc</Type>
      </TransferTypes>
    </LocationTransferType>
  </LocationTransferTypeConstraint>
</ReplicationLocation>
```

错误码

错误码	状态码	描述
NoSuchBucket	404 Not Found	请求的Bucket不存在。

6.7.4. GetBucketReplicationProgress

GetBucketReplicationProgress用于获取某个存储空间（Bucket）的数据复制进度。

请求语法

GET /?replicationProgress&rule-id=RuleId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue

请求参数

名称	类型	是否必须	描述
rule-id	字符串	否	复制规则对应的ID。此ID可从GetBucketReplication中获取。

响应元素

名称	类型	描述
ReplicationProgress	容器	保存复制进度的容器。 父节点：无 子节点：Rule
Rule	容器	保存每个复制规则进度条目的容器。 父节点：ReplicationConfiguration 子节点：ID、Destination、Status和Progress
ID	字符串	复制规则对应的ID。 父节点：Rule 子节点：无
PrefixSet	容器	保存Prefix 的容器，每个复制规则中，最多能指定10个Prefix。 父节点：Rule 子节点：Prefix
Prefix	字符串	待复制Object的Prefix，只有匹配该Prefix的Object才会被复制到目标Bucket。 父节点：PrefixSet 子节点：无

名称	类型	描述
Action	字符串	表示被同步到目标Bucket的操作。 Action允许以下操作类型，您可以指定一项或者多项。 - ALL（默认操作）：表示PUT、DELETE、ABORT操作均会被同步到目标Bucket。 - PUT：表示被同步到目标Bucket的写入操作，包括PutObject、PostObject、AppendObject、CopyObject、PutObjectACL、InitiateMultipartUpload、UploadPart、UploadPartCopy和CompleteMultipartUpload。 父节点：Rule 子节点：无
Destination	容器	保存目标Bucket信息的容器。 父节点：Rule 子节点：Bucket和Location
Bucket	字符串	数据被复制到的目标Bucket。 父节点：Destination 子节点：无
Location	字符串	目标Bucket所在的Location。 父节点：Destination 子节点：无
TransferType	字符串	数据复制时使用的数据传输类型。 - internal（默认值）：OSS默认传输链路。 - oss_acc：传输加速链路。只有创建跨区域复制规则时才能使用传输加速链路。
HistoricalObjectReplication	字符串	是否复制历史数据。即开启数据复制前，是否将源Bucket中的数据复制到目标Bucket。 取值： - enabled（默认值）：表示复制历史数据。 - disabled：表示不复制历史数据，仅复制跨区域复制规则生效后新写入的数据。
Progress	容器	保存复制进度的容器，仅当数据处于同步状态（doing）时才返回此元素。 父节点：Rule 子节点：HistoricalObject、NewObject

名称	类型	描述
HistoricalObject	字符串	显示已复制历史数据的百分比。仅对开启了历史数据复制的Bucket有效。 父节点：Progress 子节点：无
NewObject	字符串	显示数据复制到目标Bucket的时间点（GMT 格式）。 例如Thu, 24 Sep 2015 15:39:18 GMT，表示早于该时间点写入的数据都已复制到目标Bucket。 父节点：Progress 子节点：无

示例

● 请求示例

```
GET /?replicationProgress&rule-id=test_replication_1 HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 24 Sep 2015 15:39:15 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:CTkuxpLAI4XZ+WwIfNm0Fmgb****
```

● 返回示例

 说明 仅当传输类型为oss_acc时，返回的XML消息体中才会包含<TransferType>元素。

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Thu, 24 Sep 2015 15:39:15 GMT
Content-Length: 234
Content-Type: application/xml
Connection: close
Server: AliyunOSS
<?xml version="1.0" ?>
<ReplicationProgress>
  <Rule>
    <ID>test_replication_1</ID>
    <PrefixSet>
      <Prefix>source_image</Prefix>
      <Prefix>video</Prefix>
    </PrefixSet>
    <Action>PUT</Action>
    <Destination>
      <Bucket>target-bucket</Bucket>
      <Location>oss-cn-beijing</Location>
      <TransferType>oss_acc</TransferType>
    </Destination>
    <Status>doing</Status>
    <HistoricalObjectReplication>enabled</HistoricalObjectReplication>
    <Progress>
      <HistoricalObject>0.85</HistoricalObject>
      <NewObject>2015-09-24T15:28:14.000Z </NewObject>
    </Progress>
  </Rule>
</ReplicationProgress>
```

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404 NotFound	请求的Bucket不存在。
NoSuchReplicationRule	404 NotFound	指定的RuleId不存在。
NoSuchReplicationConfiguration	404 NotFound	请求的Bucket没有配置数据复制。
TooManyReplicationRules	400 BadRequest	请求中配置的数据复制规则超过一条。 单次请求中只能配置一条数据复制规则。

6.7.5. DeleteBucketReplication

DeleteBucketReplication接口用来停止某个存储空间（Bucket）的数据复制并删除Bucket的复制配置，此时源Bucket中的任何操作都不会被同步到目标Bucket。

注意事项

- 当请求的Bucket没有配置数据复制规则时，调用此接口将返回200 HTTP OK。
- 调用此接口删除某个数据复制规则时，该复制规则不会立刻被删除。OSS需要一定的时间来执行清理操作，此时复制规则的状态为closing。当清理工作完成后，该复制规则才被删除。
- 当请求的Bucket的数据复制规则处于closing状态时，调用此接口将返回204 NoContent。

请求语法

```
POST /?replication&comp=delete HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: ContentLength
Content-Type: application/xml
Authorization: SignatureValue

<?xml version="1.0" encoding="UTF-8"?>
<ReplicationRules>
  <ID>rule id</ID>
</ReplicationRules>
```

请求元素

名称	类型	是否必选	描述
ReplicationRules	容器	是	保存需要删除的数据复制规则的容器。 父节点：无 子节点：ID
ID	字符串	是	需要删除的复制规则对应的ID。规则ID可从GetBucketReplication中获取。 父节点：ReplicationRules 子节点：无

示例

- 请求示例

```
POST /?replication&comp=delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 24 Sep 2015 15:39:18 GMT
Content-Length: 46
Content-Type: application/xml
Authorization: OSS qn6qrrqxo2oawuk53otf****:CTkuxpLAI4XZ+WwIfNm0Fmgb****
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationRules>
  <ID>test_replication_1</ID>
</ReplicationRules>
```

● 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Thu, 24 Sep 2015 15:39:18 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

错误码

错误码	状态码	描述
NoSuchBucket	404 NotFound	请求的Bucket不存在。
TooManyReplicationRules	400 BadRequest	请求中配置的数据复制规则超过一条。 单次请求中只能配置一条数据复制规则。

错误码	状态码	描述
TransferAccAlreadyInUse	409Conflict	对跨区域复制指定的目标Bucket关闭了传输加速，此时错误XML中返回跨区域复制的源Bucket和目标Bucket信息如下： <pre><?xml version="1.0" encoding="UTF-8"?> <Error> <Code>TransferAccAlreadyInUse</Code> <Message>The transfer acceleration is already used by cross-region replication.</Message> <SourceBucket>srcBucket</SourceBucket> <DestinationBucket>destBucket</DestinationBucket> <RequestId>5F1E76142A535D373683****</RequestId> <HostId>oss-cn-hangzhou.aliyuncs.com</HostId> </Error></pre>

6.8. 授权策略（Policy）

6.8.1. PutBucketPolicy

PutBucketPolicy接口用于为指定的存储空间（Bucket）设置授权策略（Policy）。

背景信息

Bucket Policy是基于资源的授权策略。Bucket Policy常见的应用场景如下：

有关Bucket Policy的配置详情及场景案例，请参见[通过Bucket Policy授权用户访问指定资源](#)。有关Policy语法，请参见[权限策略语法和结构](#)。

请求语法

```
PUT /?policy
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 13 Nov 2018 09:09:13 GMT
Authorization: OSS xxxxxxxx:xxxxxxxxxx
Policy written in JSON
```

示例

- 请求示例

```
PUT /?policy
Content-Length: 230
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 21 Feb 2019 12:51:09 GMT
Authorization: OSS xxxxxxxx:xxxxxxxxxxxxxx
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "oss:PutObject",
        "oss:GetObject"
      ],
      "Effect": "Deny",
      "Principal": ["1234567890"],
      "Resource": ["acs:oss:*:1234567890:*/*"]
    }
  ]
}
```

- 返回示例

```
200 (OK)
content-length: 0
server: AliyunOSS
x-oss-server-time: 87
connection: keep-alive
x-oss-request-id: 5C6E9EBD5CC26B28EE41****
date: Thu, 21 Feb 2019 12:51:09 GMT
```

6.8.2. GetBucketPolicy

GetBucketPolicy用于获取指定存储空间（Bucket）的权限策略（Policy）。

背景信息

Bucket Policy是基于资源的授权策略。Bucket Policy常见的应用场景如下：

有关Bucket Policy的配置详情及场景案例，请参见[通过Bucket Policy授权用户访问指定资源](#)。有关Policy语法，请参见[权限策略语法和结构](#)。

请求语法

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

示例

- 请求示例

```
GET /?policy
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 13 Nov 2018 09:09:13 GMT
Authorization: OSS xxxxxxxx:xxxxxxxxxx
```

- 返回示例

```
200 (OK)
server: AliyunOSS
x-oss-server-time: 24
connection: keep-alive
x-oss-request-id: 5C6E9847BE0EBCD13DA90C11
date: Thu, 21 Feb 2019 12:23:35 GMT
content-type: application/json
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "oss:PutObject",
        "oss:GetObject"
      ],
      "Effect": "Deny",
      "Principal": ["1234567890"],
      "Resource": ["acs:oss:*:1234567890:*/*"]
    }
  ]
}
```

错误码

错误码	HTTP状态码	描述
NoSuchBucketPolicy	404	请求的Bucket没有配置权限策略（Policy）。

6.8.3. DeleteBucketPolicy

DeleteBucketPolicy用于删除指定存储空间（Bucket）的权限策略（Policy）。

背景信息

Bucket Policy是基于资源的授权策略。Bucket Policy常见的应用场景如下：

有关Bucket Policy的配置详情及场景案例，请参见[通过Bucket Policy授权用户访问指定资源](#)。有关Policy语法，请参见[权限策略语法和结构](#)。

示例

- 请求示例

```
DELETE /?policy
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 21 Feb 2019 12:55:10 GMT
Authorization: OSS xxxxxxxx:xxxxxxxxxxxxxxxx
```

- 返回示例

```
204 (No Content)
content-length: 0
server: AliyunOSS
x-oss-server-time: 31
connection: keep-alive
x-oss-request-id: 5C6E9FAF31A13327124B****
date: Thu, 21 Feb 2019 12:55:11 GMT
```

6.9. 清单 (Inventory)

6.9.1. PutBucketInventory

调用PutBucketInventory接口为指定存储空间（Bucket）配置清单（Inventory）规则。

注意事项

您可以通过清单获取Bucket中Object的各类信息，包括Object的数量、大小、存储类型、加密状态等。配置清单规则时，有如下注意事项：

- 要为Bucket配置清单规则，您必须 `oss:PutBucketInventory` 权限。
- 配置清单规则前需生成一个RAM角色，该角色需要拥有读取源Bucket所有文件和向目标Bucket写入文件的权限。首次使用清单功能时，建议您通过OSS控制台进行配置。清单规则配置完成后，您可以获取拥有所有权限的RAM角色。有关配置清单规则中RAM角色的权限说明，请参见[存储空间清单](#)。
- 单个Bucket最多只能配置1000条清单规则。
- 配置清单的源Bucket与存放清单文件的目标Bucket必须位于同一个Region。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求语法

```
PUT /?inventory&inventoryId HTTP/1.1
Host: bucket.oss.aliyuncs.com
GMT Date
Authorization: SignatureValue
Content-Length: length
<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration>
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Filter>
    <Prefix>filterPrefix</Prefix>
  </Filter>
  <Destination>
    <OSSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1000000000000000</AccountId>
      <RoleArn>acs:ram::1000000000000000:role/AliyunOSSRole</RoleArn>
      <Bucket>acs:oss:::destination-bucket</Bucket>
      <Prefix>prefix1</Prefix>
      <Encryption>
        <SSE-KMS>
          <KeyId>keyId</KeyId>
        </SSE-KMS>
      </Encryption>
    </OSSBucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>EncryptionStatus</Field>
  </OptionalFields>
</InventoryConfiguration>
```

请求参数

请求Query参数

参数名称	类型	是否必选	示例	说明
inventoryId	String	是	report1	清单名称，清单名称在当前Bucket下必须唯一。

请求Body参数

参数名称	类型	是否必选	示例	说明
bucket	String	是	oss-example	Bucket名称。
inventoryConfiguration	InventoryConfiguration	是		清单配置信息。

响应体语法

响应参数

无响应参数

示例

请求示例


```
PUT /?inventory&inventoryId=report1 HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Mon, 31 Oct 2016 12:00:00 GMT
Authorization: authorization string
Content-Length: length
<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration>
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Filter>
    <Prefix>Pics</Prefix>
  </Filter>
  <Destination>
    <OSSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1000000000000000</AccountId>
      <RoleArn>acs:ram::1000000000000000:role/AliyunOSSRole</RoleArn>
      <Bucket>acs:oss::bucket_0001</Bucket>
      <Prefix>prefix1</Prefix>
      <Encryption>
        <SSE-KMS>
          <KeyId>keyId</KeyId>
        </SSE-KMS>
      </Encryption>
    </OSSBucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>EncryptionStatus</Field>
  </OptionalFields>
</InventoryConfiguration>
```

正常返回示例

JSON 格式

HTTP/1.1 200 OK

返回示例补充说明

```
...
HTTP/1.1 200 OK
x-oss-request-id: 56594298207FB3044385****
Date: Mon, 31 Oct 2016 12:00:00 GMT
Content-Length: 0
Server: AliyunOSS
...
```

错误码

访问[错误中心](#)查看更多错误码。

开发者资源

- [SDK](#)

阿里云为您提供多种语言的SDK，帮助您快速通过API集成阿里云的产品和服务，推荐您使用SDK调用API，已免除您手动签名验证。

- [OpenAPI Explorer](#)

快速检索，可视化调试API，在线命令行工具，同步动态生成可执行的SDK代码示例。

- [阿里云CLI](#)

阿里云资产管理和配置工具，可通过命令方式同时管理多个阿里云产品和服务，简单快捷，是您上云好帮手。

SDK

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
InvalidArgument	400	传入非法参数。
InventoryExceedLimit	400	超出清单配置规则的数量限制。
AccessDenied	403	·发起PutBucketInventory请求时，没有传入用户验证信息。 ·用户无操作权限。

错误码	HTTP状态码	描述
InventoryAlreadyExist	409	请求的清单规则已存在。

6.9.2. GetBucketInventory

GetBucketInventory用于查看某个存储空间（Bucket）中指定的清单（Inventory）任务。

 **说明** 调用该请求时，请确保您有足够的权限对存储空间的清单任务进行操作。存储空间所有者默认拥有该权限，若您无该项权限，请先向存储空间所有者申请该项操作的权限。

请求语法

```
GET /?inventory&inventoryId=inventoryId HTTP/1.1
```

请求元素

名称	类型	是否必选	描述
inventoryId	字符串	是	查询的清单规则Id。

响应元素

名称	类型	是否必选	描述
Id	字符串	是	由用户指定的清单名称，清单名称在当前存储空间下必须全局唯一。
IsEnabled	布尔	是	清单是否启用的标识。 有效值：true或false - 如果设置为true，清单功能将生效。 - 如果设置为false，将不生成任何清单。
Filter	容器	否	清单筛选的前缀。指定前缀后，清单将筛选出符合前缀设置的对象。
Prefix	字符串	否	筛选规则的匹配前缀。 父节点：Filter
Destination	容器	是	存放清单结果的信息。
OSSBucketDestination	容器	是	清单结果导出后存放的存储空间信息。 父节点：Destination

名称	类型	是否必选	描述
Format	字符串	是	导出清单文件的文件格式。 有效值：CSV 父节点：OSSBucketDestination
AccountId	字符串	是	存储空间所有者授予的账户ID。 父节点：OSSBucketDestination
RoleArn	字符串	是	存储空间所有者授予操作权限的角色名。 格式为：acs:ram::uid:role/rolename 父节点：OSSBucketDestination
Bucket	字符串	是	存放导出的清单文件的存储空间。 父节点：OSSBucketDestination
Prefix	字符串	否	清单文件的存储路径前缀。 父节点：OSSBucketDestination
Encryption	容器	否	清单文件的加密方式。 有效值：SSE-OSS、SSE-KMS或空 父节点：OSSBucketDestination
SSE-OSS	容器	否	SSE-OSS加密方式的容器。 父节点：Encryption
SSE-KMS	容器	否	用于保存SSE-KMS加密方式下的密钥的容器。 父节点：Encryption
KeyId	字符串	否	KMS密钥id。 父节点：SSE-KMS
Schedule	容器	是	存放清单导出周期信息的容器。
Frequency	字符串	是	清单文件导出的周期。 有效值：Daily或Weekly 父节点：Schedule

名称	类型	是否必选	描述
IncludedObjectVersions	字符串	是	是否在清单中包含Object版本信息。 有效值：All或Current - 如果为All，则导出Object的所有版本信息。 - 如果为Current，则导出Object的当前版本信息。
OptionalFields	容器	否	设置清单结果中应包含的配置项。
Field	容器	否	清单结果中包含的配置项。 可选的配置项：Size、LastModifiedDate、ETag、StorageClass、IsMultipartUploaded、EncryptionStatus 父节点：OptionalFields

示例

- 请求示例

```
GET /?inventory&inventoryId=list1 HTTP/1.1
```

- 返回示例

```

HTTP/1.1 200 OK
x-oss-request-id: 56594298207FB304438516F9
Date: Mon, 31 Oct 2016 12:00:00 GMT
Server: AliyunOSS
Content-Length: length
<?xml version="1.0" encoding="UTF-8"?>
<InventoryConfiguration>
  <Id>report1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <OSSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1000000000000000</AccountId>
      <RoleArn>acs:ram::1000000000000000:role/AliyunOSSRole</RoleArn>
      <Bucket>acs:oss::bucket_0001</Bucket>
      <Prefix>prefix1</Prefix>
      <Encryption>
        <SSE-OSS/>
      </Encryption>
    </OSSBucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>myprefix</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>EncryptionStatus</Field>
  </OptionalFields>
</InventoryConfiguration>

```

6.9.3. ListBucketInventory

ListBucketInventory用于批量获取某个存储空间（Bucket）中的所有清单（Inventory）任务。

② 说明

- 单次请求最多可获取100条清单配置项内容。若需获取超过100条清单配置项，则需发送多次请求，并保留相应的token，作为下一次请求的参数。
- 调用该请求时，请确保您有足够的权限对存储空间的清单任务进行操作。存储空间所有者默认拥有该权限，若您无该项权限，请先向存储空间所有者申请该项操作的权限。

请求语法

- 带continuation-token的形式

```
GET /?inventory&continuation-token=xxx HTTP/1.1
```

- 不带continuation-token的形式

```
GET /?inventory HTTP/1.1
```

响应元素

名称	类型	描述
InventoryConfiguration	容器	存放清单配置参数的容器。
IsTruncated	布尔	是否列举全部的清单任务。 有效值：true或false - 如果值为false，则表明存储空间中的所有清单任务已全部列出。 - 如果值为true，表示还未完整列出存储空间中的所有清单任务，您可以将NextContinuationToken字段的值作为下一次list请求的continuation-token参数，以获取下一页的清单配置列表。
NextContinuationToken	字符串	当响应中的IsTruncated为true且NextContinuationToken非空时，使用该字段作为下一次list请求的continuation-token参数。
Id	字符串	由用户指定的清单名称，清单名称在当前存储空间下必须全局唯一。
IsEnabled	布尔	清单是否启用的标识。 有效值：true或false - 如果设置为true，清单功能将生效。 - 如果设置为false，将不生成任何清单。
Filter	容器	清单筛选的前缀。指定前缀后，清单将筛选出符合前缀设置的对象。
Prefix	字符串	筛选规则的匹配前缀。 父节点：Filter
Destination	容器	存放清单结果的信息。
OSSBucketDestination	容器	清单结果导出后存放的存储空间信息。 父节点：Destination

名称	类型	描述
Format	字符串	导出清单文件的文件格式。 有效值：CSV 父节点：OSSBucketDestination
AccountId	字符串	存储空间所有者授予的账户ID。 父节点：OSSBucketDestination
RoleArn	字符串	存储空间所有者授予操作权限的角色名。 格式为：acs:ram::uid:role/rolename 父节点：OSSBucketDestination
Bucket	字符串	存放导出的清单文件的存储空间。 父节点：OSSBucketDestination
Prefix	字符串	清单文件的存储路径前缀。 父节点：OSSBucketDestination
Encryption	容器	清单文件的加密方式。 有效值：SSE-OSS、SSE-KMS或Null 父节点：OSSBucketDestination
SSE-OSS	容器	SSE-OSS加密方式的容器。 父节点：Encryption
SSE-KMS	容器	用于保存SSE-KMS加密方式下的密钥的容器。 父节点：Encryption
KeyId	字符串	KMS密钥id。 父节点：SSE-KMS
Schedule	容器	存放清单导出周期信息的容器。
Frequency	字符串	清单文件导出的周期。 有效值：Daily或Weekly 父节点：Schedule

名称	类型	描述
IncludedObjectVersions	字符串	是否在清单中包含Object版本信息。 有效值：All或Current - 如果为All，则导出Object的所有版本信息。 - 如果为Current，则导出Object的当前版本信息。
OptionalFields	容器	设置清单结果中应包含的配置项。
Field	容器	清单结果中包含的配置项。 可选的配置项：Size、LastModifiedDate、ETag、StorageClass、IsMultipartUploaded、EncryptionStatus 父节点：OptionalFields

示例

● 请求示例

```
GET /?inventory HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: Fri, 24 Feb 2012 03:55:00 GMT
Authorization: authorization string
Content-Type: text/plain
```

● 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 56594298207FB304438516F9
Date: Sat, 30 Apr 2016 23:29:37 GMT
Content-Type: application/xml
Content-Length: length
Connection: close
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<ListInventoryConfigurationsResult>
  <InventoryConfiguration>
    <Id>report1</Id>
    <IsEnabled>true</IsEnabled>
    <Destination>
      <OSSBucketDestination>
        <Format>CSV</Format>
        <AccountId>1000000000000000</AccountId>
        <RoleArn>acs:ram::1000000000000000:role/AllyunOSSRole</RoleArn>
        <Bucket>acs:oss:::destination-bucket</Bucket>
        <Prefix>prefix1</Prefix>
      </OSSBucketDestination>
    </Destination>
    <Schedule>
      <Frequency>Daily</Frequency>
    </Schedule>
  </InventoryConfiguration>
</ListInventoryConfigurationsResult>
```

```

<Filter>
  <Prefix>prefix/One</Prefix>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
<OptionalFields>
  <Field>Size</Field>
  <Field>LastModifiedDate</Field>
  <Field>ETag</Field>
  <Field>StorageClass</Field>
  <Field>IsMultipartUploaded</Field>
  <Field>EncryptionStatus</Field>
</OptionalFields>
</InventoryConfiguration>
<InventoryConfiguration>
  <Id>report2</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <OSSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1000000000000000</AccountId>
      <RoleArn>acs:ram::1000000000000000:role/AliyunOSSRole</RoleArn>
      <Bucket>acs:oss:::destination-bucket</Bucket>
      <Prefix>prefix2</Prefix>
    </OSSBucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>prefix/Two</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>EncryptionStatus</Field>
  </OptionalFields>
</InventoryConfiguration>
<InventoryConfiguration>
  <Id>report3</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <OSSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1000000000000000</AccountId>
      <RoleArn>acs:ram::1000000000000000:role/AliyunOSSRole</RoleArn>
      <Bucket>acs:oss:::destination-bucket</Bucket>
      <Prefix>prefix3</Prefix>
    </OSSBucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>

```

```
</Schedule>
<Filter>
  <Prefix>prefix/Three</Prefix>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
<OptionalFields>
  <Field>Size</Field>
  <Field>LastModifiedDate</Field>
  <Field>ETag</Field>
  <Field>StorageClass</Field>
  <Field>IsMultipartUploaded</Field>
  <Field>EncryptionStatus</Field>
</OptionalFields>
</InventoryConfiguration>
...
<IsTruncated>true</IsTruncated>
<NextContinuationToken>...</NextContinuationToken>
</ListInventoryConfigurationsResult>
```

6.9.4. DeleteBucketInventory

DeleteBucketInventory用于删除某个存储空间（Bucket）中指定的清单（Inventory）任务。

 说明

- 调用该请求时，请确保您有足够的权限对存储空间的清单任务进行操作。存储空间所有者默认拥有该权限，若您无该项权限，请先向存储空间所有者申请该项操作的权限。
- 请求成功将返回HTTP状态码204。

请求语法

```
DELETE ?/inventory&inventoryId=list1 HTTP/1.1
```

请求元素

名称	类型	是否必选	描述
inventoryId	字符串	是	删除的清单任务Id。

示例

● 请求示例

```
DELETE ?/inventory&inventoryId=list1 HTTP/1.1
Host: BucketName.oss.aliyuncs.com
Date: Wed, 14 May 2014 02:11:22 GMT
Authorization: signatureValue
```

● 返回示例

```
HTTP/1.1 204 No Content
x-oss-request-id: 56594298207FB3044385****
Date: Wed, 14 May 2014 02:11:22 GMT
Server: AliyunOSS
```

错误码

错误码	HTTP状态码	描述
InvalidRequest	400	非法请求。
AccessDenied	403	- 发起DeleteBucketInventory请求时没有传入用户验证信息。 - 没有操作权限。

6.10. 日志管理（Logging）

6.10.1. PutBucketLogging

PutBucketLogging接口用于为存储空间（Bucket）开启日志转存功能，可将OSS的访问日志按照固定命名规则，以小时为单位生成日志文件写入您指定的Bucket。

注意事项

- 生成日志的源Bucket和存储日志的目标Bucket可以相同也可以不同，但是必须属于同一账号下的相同地域。
- 日志文件以小时为单位生成，但并不表示某个时段的日志文件记录了该时段的所有请求，部分请求可能会出现在上一时段或下一时段的日志文件中。

日志文件命名规则及日志格式说明，请参见[日志转存](#)。

- 在您关闭日志转存功能前，OSS的日志文件会一直生成。请及时清理不再需要的日志文件，以减少您的存储费用。

您可以通过生命周期规则定期删除日志文件。更多信息，请参见[基于最后一次修改时间的生命周期规则介绍](#)。

- OSS会根据需求在日志的尾部添加一些字段，请您在开发日志处理工具时考虑兼容性的问题。

请求语法

```
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>
```

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求元素

名称	类型	是否必需	示例值	描述
BucketLoggingStatus	容器	是	不涉及	存储访问日志状态信息的容器。 子元素：LoggingEnabled 父元素：无
LoggingEnabled	容器	开启日志转存时必选	不涉及	访问日志信息的容器。 子元素：TargetBucket, TargetPrefix 父元素：BucketLoggingStatus
TargetBucket	字符串	开启日志转存时必选	examplebucket	指定存储访问日志的Bucket。 子元素：无 父元素：BucketLoggingStatus.LoggingEnabled
TargetPrefix	字符串	否	MyLog-	指定保存的日志文件前缀，可以为空。 子元素：无 父元素：BucketLoggingStatus.LoggingEnabled

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 开启Bucket日志转存的请求示例

```
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 qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrTZ****
<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus>
<LoggingEnabled>
<TargetBucket>examplebucket</TargetBucket>
<TargetPrefix>MyLog-</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E888648906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- 关闭Bucket日志转存的请求示例

关闭Bucket的日志转存功能时，只需发送一个空的BucketLoggingStatus即可。示例如下：

```
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 qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrTZ****
<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus>
</BucketLoggingStatus>
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674125A4D8906008B
Date: Fri, 04 May 2012 04:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)

- .NET
- Node.js
- Ruby

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	源Bucket不存在。
InvalidTargetBucketForLogging	400	源Bucket和目标Bucket不属于同一个数据中心。
InvalidDigest	400	上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性，如果不一致会返回此错误码。
MalformedXML	400	请求中的XML不合法。
InvalidTargetBucketForLogging	403	请求发起者不是目标Bucket的拥有者。
AccessDenied	403	请求发起者不是源Bucket的拥有者。

6.10.2. GetBucketLogging

GetBucketLogging接口用于查看存储空间（Bucket）的访问日志配置。只有Bucket的拥有者才能查看Bucket的访问日志配置。

请求语法

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

响应元素

名称	类型	描述
BucketLoggingStatus	容器	访问日志状态信息的容器。 子元素：LoggingEnabled 父元素：无  说明 如果源Bucket未设置日志规则，OSS仍然返回一个XML消息体，但其中的BucketLoggingStatus元素为空。

名称	类型	描述
LoggingEnabled	容器	访问日志信息的容器。此元素在开启时返回，关闭时不返回。 子元素：TargetBucket, TargetPrefix 父元素：BucketLoggingStatus
TargetBucket	字符	指定存放访问日志的Bucket。 子元素：无 父元素：BucketLoggingStatus.LoggingEnabled
TargetPrefix	字符	指定最终被保存的访问日志文件前缀。 子元素：无 父元素：BucketLoggingStatus.LoggingEnabled

示例

请求示例

```
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:ceOEyZavKY4QcjoUWYSpYbJ3****
```

返回示例

- 已设置日志规则的返回示例

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>
```

- 未设置日志规则的返回示例


```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有查看Bucket访问日志配置的权限。只有Bucket的拥有者才能查看Bucket的访问日志配置。

6.10.3. DeleteBucketLogging

DeleteBucketLogging用于关闭存储空间（Bucket）的访问日志记录功能。只有Bucket的拥有者才有权限关闭Bucket访问日志记录功能。

请求语法

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

示例

请求示例

```
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/Cn****
```

返回示例

 **说明** 如果目标Bucket没有开启访问日志记录功能，则返回204状态码。

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 24 Feb 2012 05:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有关闭Bucket访问日志记录功能的权限。只有Bucket的拥有者才可以关闭Bucket访问日志记录功能。

6.11. 静态网站（Website）

6.11.1. PutBucketWebsite

调用PutBucketWebsite接口将存储空间（Bucket）设置为静态网站托管模式并设置跳转规则（RoutingRule）。

注意事项

静态网站是指所有的网页都由静态内容构成，包括客户端执行的脚本，例如JavaScript。OSS不支持需要服务器端处理的内容，例如PHP、JSP、ASP.NET等。

- 功能支持
此接口主要用于设置默认主页、默认404页和RoutingRule。RoutingRule用来指定3xx跳转规则以及镜像回源规则。其中镜像回源支持公共云和金融云。
- 使用自有域名访问静态网站
如果要使用自有域名来访问基于Bucket的静态网站，您可以通过域名CNAME来实现。具体操作，请参见[绑定自定义域名](#)。
- 索引页面和错误页面
将一个Bucket设置为静态网站托管模式时，如果指定了索引页面或错误页面，则指定的索引页面和错误页面为该Bucket内的某个Object。
- 对静态网站根域名的匿名访问
将一个Bucket设置为静态网站托管模式后，对静态网站根域名的匿名访问，OSS将返回索引页面。对静态网站根域名的签名访问，OSS将返回Get Bucket(List Objects)的结果。

请求语法

```
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>
    <HttpStatus>404</HttpStatus>
  </ErrorDocument>
</WebsiteConfiguration>
```

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求参数

- WebsiteConfiguration的内容

名称	类型	是否必选	描述
WebsiteConfigura tion	容器	是	根节点。 父节点：无

- IndexDocument的内容

名称	类型	是否必选	描述
IndexDocument	容器	有条件 至少指定 IndexDocument、ErrorDocument、RoutingRules 三个容器中的一个。	默认主页的容器。 父节点：WebsiteConfiguration
Suffix	字符串	有条件 如果指定了父节点 IndexDocument，则必须指定此项。	默认主页。 设置默认主页后，如果访问以正斜线 (/) 结尾的 Object，则 OSS 都会返回此默认主页。 父节点：IndexDocument
SupportSubDir	字符串	否	访问子目录时，是否支持转到子目录下的默认主页。取值范围如下： ◦ <i>true</i>：转到子目录下的默认主页。 ◦ <i>false</i>（默认）：不转到子目录下的默认主页，而是转到根目录下的默认主页。 假设默认主页为 index.html，要访问 bucket.oss-cn-hangzhou.aliyuncs.com/subdir/，如果设置 SupportSubDir 为 false，则转到 bucket.oss-cn-hangzhou.aliyuncs.com/index.html；如果设置 SupportSubDir 为 true，则转到 bucket.oss-cn-hangzhou.aliyuncs.com/subdir/index.html。 父节点：IndexDocument

名称	类型	是否必选	描述
Type	枚举值	否	设置默认主页后，访问以非正斜线 (/) 结尾的Object，且该Object不存在时的行为。只有设置SupportSubDir为true时才生效，且生效的顺序在RoutingRule之后、ErrorFile之前。 假设默认主页为index.html。要访问的文件路径为 bucket.oss-cn-hangzhou.aliyuncs.com/abc，且abc这个Object不存在，此时Type的不同取值对应的行为如下： 0（默认）：检查abc/index.html是否存在（即 Object + 正斜线 (/) + 主页的形式），如果存在则返回302，Location头为 /abc/ 的URL编码（即 正斜线 (/) + Object + 正斜线 (/) 的形式），如果不存在则返回404，继续检查ErrorFile。 1：直接返回404，报错NoSuchKey，继续检查ErrorFile。 2：检查abc/index.html是否存在，如果存在则返回该Object的内容；如果不存在则返回404，继续检查ErrorFile。 父节点：IndexDocument

• ErrorDocument的内容

名称	类型	是否必选	描述
ErrorDocument	容器	有条件 至少指定IndexDocument、ErrorDocument、RoutingRules三个容器中的一个。	404页面的容器。 父节点：WebsiteConfiguration
Key	字符串	有条件 如果指定了父节点ErrorDocument，则必须指定此项。	错误页面。 指定错误页面后，如果访问的Object不存在，则返回此错误页面。 父节点：ErrorDocument
HttpStatus	字符串	否	返回错误页面时的HTTP状态码。 取值：200、404（默认） 父节点：ErrorDocument

• RoutingRules|RoutingRule|RuleNumber的内容

名称	类型	是否必选	描述
RoutingRules	容器	有条件 至少指定 IndexDocument、ErrorDocument、RoutingRules 三个容器中的一个。	RoutingRule 的容器。 父节点：WebsiteConfiguration
RoutingRule	容器	否	指定跳转规则或者镜像回源规则，最多指定 20 个 RoutingRule。 父节点：RoutingRules
RuleNumber	正整数	有条件 如果指定了父节点 RoutingRule，则必须指定此项。	匹配和执行 RoutingRule 的序号，OSS 将按照此序号依次匹配规则。如果匹配成功，则执行此规则，后续的规则不再执行。 父节点：RoutingRule

● RoutingRules|RoutingRule|Condition 的内容

名称	类型	是否必选	描述
Condition	容器	有条件 如果指定了父节点 RoutingRule，则必须指定此项。	匹配的条件。 如果指定的项都满足，则执行此规则。只有满足此容器下的各个节点的所有条件才算匹配。 父节点：RoutingRule
KeyPrefixEquals	字符串	否	只有匹配此前缀的 Object 才能匹配此规则。 父节点：Condition
HttpErrorCodeReturnedEquals	HTTP 状态码	否	访问指定 Object 时，返回此 status 才能匹配此规则。当跳转规则是镜像回源类型时，此字段必须为 404。 父节点：Condition
IncludeHeader	容器	否	只有请求中包含了指定 Header 且值为指定值时，才能匹配此规则。该容器最多可指定 10 个。 父节点：Condition
Key	字符串	是	只有请求中包含了此 Header 且值为 Equals 的指定值时，才能匹配此规则。 父节点：IncludeHeader

名称	类型	是否必选	描述
Equals	字符串	否	只有请求中包含了Key指定的Header且值为指定值时，才能匹配此规则。 父节点：IncludeHeader
KeySuffixEquals	字符串	否	只有匹配此字段指定的后缀才能匹配此规则。 默认值为空，表示不匹配指定的后缀。 父节点：Condition

● RoutingRules|RoutingRule|Redirect的内容

名称	类型	是否必选	描述
Redirect	容器	有条件 如果指定了父节点RoutingRule，则必须指定此项。	指定匹配此规则后执行的动作。 父节点：RoutingRule
RedirectType	字符串	有条件 如果指定了父节点Redirect，则必须指定此项。	指定跳转的类型。取值范围如下： ◦ <i>Mirror</i>: 镜像回源。 ◦ <i>External</i>: 外部跳转，即OSS会返回一个3xx请求，指定跳转到另外一个地址。 ◦ <i>AliCDN</i>: 阿里云CDN跳转，主要用于阿里云的CDN。与External不同的是，OSS会额外添加一个Header。阿里云CDN识别到此Header后会主动跳转到指定的地址，返回给用户获取到的数据，而不是将3xx跳转请求返回给用户。 父节点：Redirect
PassQueryString	布尔	否	执行跳转或者镜像回源规则时，是否携带请求参数。 用户请求OSS时携带了请求参数 <code>?a=b&c=d</code> ，并且设置PassQueryString为true，如果规则为302跳转，则跳转的Location头中会添加此请求参数。例如 <code>Location:example.com?a=b&c=d</code> ，跳转类型为镜像回源，则在发起的回源请求中也会携带此请求参数。 取值：true、false（默认） 父节点：Redirect

名称	类型	是否必选	描述
MirrorURL	字符串	有条件 如果RedirectType指定为Mirror，则必须指定此项。	镜像回源的源站地址。只有设置RedirectType为Mirror时才生效。 源站地址必须以http://或者https://开头，并且以正斜线 (/) 结尾，OSS会在此地址后带上Object名称组成回源URL。 例如要访问的Object名称为mvobject，如果指定此项为 <code>http://example.com/</code>，则回源URL为 <code>http://example.com/mvobject</code>，如果指定此项为 <code>http://example.com/dir1/</code>，则回源URL为 <code>http://example.com/dir1/myobject</code>。 父节点：Redirect
TransparentMirrorResponseCodes	字符串	否	指定当源站返回哪些状态码时需要透传该状态码以及body到客户端，取值为4xx、5xx等HTTP状态码，多个HTTP状态码之间用英文逗号 (,) 分隔，例如 <code>400,404</code>。只有设置RedirectType为Mirror时才生效。 当OSS向源站请求内容时，如果源站返回了此参数中的某个状态码，则OSS将透传源站返回的该状态码以及body到客户端。  注意 如果在此参数中设置了404状态码，则设置的ErrorDocument会失效。 父节点：Redirect
MirrorPassQueryString	布尔	否	与PassQueryString作用相同，优先级高于PassQueryString。只有设置RedirectType为Mirror时才生效。 默认值：false 父节点：Redirect

名称	类型	是否必选	描述
MirrorFollowRedirect	布尔	否	如果镜像回源获取的结果为3xx，是否继续跳转到指定的Location获取数据。只有设置RedirectType为Mirror时才生效。 例如发起镜像回源请求时，源站返回了302，并且指定了Location。 - 如果设置此项为true，则OSS会继续请求Location对应的地址。 - 最多能跳转10次，如果跳转超过10次，则返回镜像回源失败。 - 如果设置此项为false，则OSS会返回302，并透传Location。 默认值：true 父节点：Redirect
MirrorCheckMd5	布尔	否	是否检查回源body的MD5。只有设置RedirectType为Mirror时才生效。 当设置MirrorCheckMd5为true，并且源站返回的response中含有Content-Md5头时，OSS检查拉取的数据MD5是否与此Header匹配，如果不匹配，则不保存在OSS上。 默认值：false 父节点：Redirect
MirrorHeaders	容器	否	指定镜像回源时携带的Header。只有设置RedirectType为Mirror时才生效。 父节点：Redirect
PassAll	布尔	否	是否透传除以下Header之外的其他Header到源站。只有设置RedirectType为Mirror时才生效。 - content-length、authorization2、authorization、range、date等Header - 以oss-/x-oss-/x-drs-开头的Header 默认值：false 父节点：MirrorHeaders

名称	类型	是否必选	描述
Pass	字符串	否	透传指定的Header到源站。只有设置RedirectType为Mirror时才生效。 每个Header长度最多为1024个字节，字符集为0~9、A~Z、a~z以及短划线（-）。 此字段最多可指定10个。 父节点：MirrorHeaders
Remove	字符串	否	禁止透传指定的Header到源站。只有设置RedirectType为Mirror时才生效。 每个Header长度最多为1024个字节，字符集与Pass相同。 此字段最多可指定10个，通常与PassAll一起使用。 父节点：MirrorHeaders
Set	容器	否	设置一个Header传到源站，不管请求中是否携带这些指定的Header，回源时都会设置这些Header。只有设置RedirectType为Mirror时才生效。 此容器最多可指定10组。 父节点：MirrorHeaders
Key	字符串	有条件 若指定了父节点Set，则必须指定此项。	设置Header的key，最多1024个字节，字符集与Pass相同。只有设置RedirectType为Mirror时才生效。 父节点：Set
Value	字符串	有条件 若指定了父节点Set，则必须指定此项。	设置Header的value，最多1024个字节，不能出现 \r\n 。只有设置RedirectType为Mirror时才生效。 父节点：Set
Protocol	字符串	否	跳转时的协议。只有设置RedirectType为External或者AliCDN时才生效。 如果要访问的文件为test，设置跳转到 <code>example.com</code>，并且设置Protocol为https，则Location头为 <code>https://example.com/test</code>。 取值：<code>http</code>、<code>https</code>。 父节点：Redirect

名称	类型	是否必选	描述
HostName	字符串	否	跳转时的域名，域名需符合域名规范。 如果要访问的文件为test，设置Protocol为https，并且设置Hostname为 example.com，则Location头为 https://example.com/test。 父节点：Redirect
ReplaceKeyPrefixWith	字符串	否	Redirect时Object名称的前缀将替换成该值。如果前缀为空，则将这个字符串插入Object名称的前面。 ❓ 说明 仅允许存在 ReplaceKeyWith或 ReplaceKeyPrefixWith节点。 假设要访问的Object为abc/test.txt，如果设置KeyPrefixEquals为abc/，ReplaceKeyPrefixWith为def/，则Location头为 http://example.com/def/test.txt。 父节点：Redirect
EnableReplacePrefix	布尔	否	如果设置此字段为true，则Object的前缀将被替换为ReplaceKeyPrefixWith指定的值。如果未指定此字段或为空，则表示截断Object前缀。 ❓ 说明 当ReplaceKeyWith字段不为空时，不能设置此字段为true。 默认值：false 父节点：Redirect
ReplaceKeyWith	字符串	否	Redirect时Object名称将替换成 ReplaceKeyWith指定的值，ReplaceKeyWith支持设置变量。目前支持的变量为\${key}，表示该请求中的Object名称。 假设要访问的Object为test，如果设置 ReplaceKeyWith 为 prefix/\${key}.suffix，则Location头为 http://example.com/prefix/test.suffix。 父节点：Redirect

名称	类型	是否必选	描述
HttpRedirectCode	HTTP状态码	否	跳转时返回的状态码。只有设置RedirectType为External或者AliCDN时才生效。 取值：301（默认）、302、307。 父节点：Redirect

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 请求示例

```
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 nxj7dtwhcyl5hp****:sNKIHT6ci/z231yIT5vYnetD****
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
    <SupportSubDir>true</SupportSubDir>
    <Type>0</Type>
  </IndexDocument>
  <ErrorDocument>
    <Key>error.html</Key>
    <HttpStatus>404</HttpStatus>
  </ErrorDocument>
</WebsiteConfiguration>
```

- 返回示例

```
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
```

- 完整示例

```
PUT /?website HTTP/1.1
Date: Fri, 27 Jul 2018 09:03:18 GMT
Content-Length: 2064
Host: test.oss-cn-hangzhou-internal.aliyuncs.com
Authorization: OSS nxj7dtwhcyl5hp****:sNKIHT6ci/z231yIT5vYnetD****
User-Agent: aliyun-sdk-python-test/0.4.0
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
```

```
<SupportSubDir>true</SupportSubDir>
<Type>0</Type>
</IndexDocument>
<ErrorDocument>
  <Key>error.html</Key>
  <HttpStatus>404</HttpStatus>
</ErrorDocument>
<RoutingRules>
  <RoutingRule>
    <RuleNumber>1</RuleNumber>
    <Condition>
      <KeyPrefixEquals>abc</KeyPrefixEquals>
      <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
    </Condition>
    <Redirect>
      <RedirectType>Mirror</RedirectType>
      <PassQueryString>true</PassQueryString>
      <MirrorURL>http://example.com/</MirrorURL>
      <TransparentMirrorResponseCodes>400,404</TransparentMirrorResponseCodes>
      <MirrorPassQueryString>true</MirrorPassQueryString>
      <MirrorFollowRedirect>true</MirrorFollowRedirect>
      <MirrorCheckMd5>false</MirrorCheckMd5>
      <MirrorHeaders>
        <PassAll>true</PassAll>
        <Pass>myheader-key1</Pass>
        <Pass>myheader-key2</Pass>
        <Remove>myheader-key3</Remove>
        <Remove>myheader-key4</Remove>
        <Set>
          <Key>myheader-key5</Key>
          <Value>myheader-value5</Value>
        </Set>
      </MirrorHeaders>
    </Redirect>
  </RoutingRule>
  <RoutingRule>
    <RuleNumber>2</RuleNumber>
    <Condition>
      <KeyPrefixEquals>abc</KeyPrefixEquals>
      <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      <IncludeHeader>
        <Key>host</Key>
        <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
      </IncludeHeader>
    </Condition>
    <Redirect>
      <RedirectType>AliCDN</RedirectType>
      <Protocol>http</Protocol>
      <HostName>example.com</HostName>
      <PassQueryString>false</PassQueryString>
      <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
      <HttpRedirectCode>301</HttpRedirectCode>
    </Redirect>
  </RoutingRule>
```

```

<RoutingRule>
  <Condition>
    <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
  </Condition>
  <RuleNumber>3</RuleNumber>
  <Redirect>
    <ReplaceKeyWith>prefix/${key}</ReplaceKeyWith>
    <HttpRedirectCode>302</HttpRedirectCode>
    <EnableReplacePrefix>false</EnableReplacePrefix>
    <PassQueryString>false</PassQueryString>
    <Protocol>http</Protocol>
    <HostName>example.com</HostName>
    <RedirectType>External</RedirectType>
  </Redirect>
</RoutingRule>
</RoutingRules>
</WebsiteConfiguration>
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 27 Jul 2018 09:03:18 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5B5ADFD6ED3CC49176CBE29D
x-oss-server-time: 47

```

SDK

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
InvalidDigest	400	上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性，如果不一致则返回此错误码。

6.11.2. GetBucketWebsite

调用GetBucketWebsite接口查看存储空间（Bucket）的静态网站托管状态以及跳转规则。

请求语法

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

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

响应元素

• WebsiteConfiguration的内容

名称	类型	示例值	描述
WebsiteConfiguration	容器	不涉及	根节点。 父节点：无

• IndexDocument的内容

名称	类型	示例值	描述
IndexDocument	容器	不涉及	默认主页的容器。 父节点：WebsiteConfiguration
Suffix	字符串	index.html	默认主页。 父节点：IndexDocument

• ErrorDocument的内容

名称	类型	示例值	描述
ErrorDocument	容器	不涉及	错误页面的容器。 父节点：WebsiteConfiguration
Key	字符串	error.html	错误页面。 父节点：ErrorDocument
HttpStatus	字符串	404	返回错误页面时的HTTP状态码。 父节点：ErrorDocument

• RoutingRules|RoutingRule|RuleNumber的内容

名称	类型	示例值	描述
RoutingRules	容器	不涉及	RoutingRule的容器。 父节点：WebsiteConfiguration
RoutingRule	容器	不涉及	跳转规则或者镜像回源规则。 父节点：RoutingRules
RuleNumber	正整数	1	匹配并执行跳转规则或者镜像回源规则的序号。 按照此序号顺序进行规则匹配，如果匹配成功，则执行此规则，且后续的规则不再执行。 父节点：RoutingRule

- RoutingRules|RoutingRule|Condition的内容

名称	类型	示例值	描述
Condition	容器	不涉及	匹配的条件，需满足所有指定的项才执行此规则。 父节点：RoutingRule
KeyPrefixEquals	字符串	abc	只有匹配此前缀的Object才能匹配此规则。 父节点：Condition
HttpErrorCodeReturnedEquals	HTTP状态码	404	访问指定Object时返回此状态码才能匹配此规则。当跳转规则是镜像回源类型时，此字段必须为404。 父节点：Condition
IncludeHeader	容器	不涉及	只有请求中包含了指定Header且值为指定值时，才能匹配此规则。此容器可以最多重复5个。 父节点：IncludeHeader
Key	字符串	host	只有请求中包含了此头且值为Equals指定值时，才能匹配此规则。 父节点：IncludeHeader
Equals	字符串	test.oss-cn-beijing-internal.aliyuncs.com	只有请求中包含了Key指定的头且值为指定的值时，才能匹配此规则。 父节点：IncludeHeader

- RoutingRules|RoutingRule|Redirect的内容

名称	类型	示例值	描述
Redirect	容器	不涉及	指定匹配此规则后执行的动作。 父节点：RoutingRule
RedirectType	字符串	Mirror	跳转类型。 ◦ Mirror：镜像回源。 ◦ External：外部跳转，即OSS会返回一个3xx请求，指定跳转到另外一个地址。 ◦ AliCDN：阿里云CDN跳转，主要用于阿里云的CDN。与External不同的是，OSS会额外添加一个Header。阿里云CDN识别到此Header后会主动跳转到指定的地址，返回给用户获取到的数据，而不是将3xx跳转请求返回给用户。 父节点：Redirect
PassQueryString	布尔值	false	执行跳转或者镜像回源时，是否要携带发起请求的请求参数。 当用户请求OSS时携带了请求参数“a=b&c=d”且此项设置为true，如果规则是302跳转，则在跳转的Location头中会添加此请求参数。例如请求时携带了“Location:example.com?a=b&c=d”，跳转类型是镜像回源，则在发起的回源请求中也会携带此请求参数。 默认值：false 父节点：Redirect
MirrorURL	字符串	http://example.com	镜像回源的源站地址。只有当RedirectType为Mirror时才生效。 源站地址必须以http://或https://开头，且以正斜线（/）结尾，OSS会在此地址后带上Object组成回源URL。 例如要访问的Object名称为mvobject，如果指定此项为 http://example.com/，则回源URL为 http://example.com/mvobject，如果指定此项为 http://example.com/dir1/，则回源URL为 http://example.com/dir1/myobject。 父节点：Redirect
TransparentMirrorResponseCodes	字符串	400,404	当源站返回哪些状态码时需要透传该状态码以及body到客户端。只有当RedirectType为Mirror时才生效。 父节点：Redirect

名称	类型	示例值	描述
MirrorPassQueryString	布尔值	false	与PassQueryString作用相同，优先级比PassQueryString高。只有当RedirectType为Mirror时才生效。 默认值：false 父节点：Redirect
MirrorFollowRedirect	布尔值	true	如果镜像回源获取的结果为3xx，是否要继续跳转到指定的Location获取数据。只有当RedirectType为Mirror时才生效。 例如发起镜像回源请求时，源站返回了302，并且指定了Location。 ◦ <i>true</i>：OSS会继续请求Location对应的地址。最多能跳转10次，如果跳转超过10次，则报错。 ◦ <i>false</i>：OSS返回302，并透传Location。 默认值：true 父节点：Redirect
MirrorCheckMd5	布尔值	false	是否要检查回源主体的MD5。只有当RedirectType为Mirror时才生效。 当设置此项为true且源站返回的response中含有Content-MD5头时，OSS会检查拉取的数据MD5是否与此头匹配，如果不匹配，则不保存在OSS上。 默认值：false 父节点：Redirect
MirrorHeaders	容器	不涉及	用于指定镜像回源时携带的头。只有当RedirectType为Mirror时才生效。 父节点：Redirect
PassAll	布尔值	true	是否透传请求中所有的Header（除了保留的几个Header以及以 <code>oss-/x-oss-/x-drs-</code> 开头的Header）到源站。只有当RedirectType为Mirror时才生效。 默认值：false 父节点：MirrorHeaders

名称	类型	示例值	描述
Pass	字符串	myheader-key1	透传指定的Header到源站。只有当RedirectType为Mirror时才生效。 每个Header长度最多为1024个字节，字符集为0~9、A~Z、a~z以及短划线（-）。 此字段最多可指定10个。 父节点：MirrorHeaders
Remove	字符串	myheader-key3	禁止透传指定的Header到源站，此字段可以重复，最多10个，一般与PassAll一起使用。每个Header长度最多1024个字节，字符集与Pass相同。只有当RedirectType为Mirror时才生效。 父节点：MirrorHeaders
Set	容器	不涉及	设置一个Header传到源站，不管请求中是否携带这些指定的Header，回源时都会设置这些Header。该容器可以重复，最多10组。只有当RedirectType为Mirror时才生效。 父节点：MirrorHeaders
Key	字符串	myheader-key5	设置Header的key，最多1024个字节，字符集与Pass相同。只有当RedirectType为Mirror时才生效。 父节点：Set
Value	字符串	myheader-value5	设置Header的value，最多1024个字节，且不能出现“\r\n”。只有当RedirectType为Mirror时才生效。 父节点：Set
Protocol	字符串	http	跳转协议。只有当RedirectType为External或AliCDN时才生效。 例如访问的文件为test，设置跳转到 example.com，且设置Protocol为https，则Location的头为 https://example.com/test。 取值：http、https 父节点：Redirect

名称	类型	示例值	描述
HostName	字符串	example.com	跳转时的域名，域名需符合域名规范。只有当RedirectType为External或AliCDN时才生效。 例如访问的Object为test，Protocol为https，Hostname指定为example.com，则Location的头为https://example.com/test。 父节点：Redirect
HttpRedirectCode	HTTP状态码	301	跳转时返回的状态码。只有当RedirectType为External或者AliCDN时才生效。 取值：301、302、307 父节点：Redirect
ReplaceKeyPrefixWith	字符串	def/	执行Redirect时文件名前缀将替换成此值。只有当RedirectType为External或者AliCDN时才生效。 例如ReplaceKeyPrefixWith指定为def/，访问的文件名称为abc/test.txt，根据KeyPrefixEquals是否为空，Location头变化如下： - 当KeyPrefixEquals指定为abc/，则Location的头为http://example.com/def/test.txt。 - 当KeyPrefixEquals指定为空，则Location的头为http://example.com/def/abc/test.txt。 父节点：Redirect
ReplaceKeyWith	字符串	prefix/\${key}.suffix	执行Redirect时文件名将替换成此值。只有当RedirectType为External或者AliCDN时才生效。 此参数支持变量\${key}，\${key}表示该请求中的文件名称。例如ReplaceKeyWith为prefix/\${key}.suffix，访问的文件名称为test，则Location的头为http://example.com/prefix/test.suffix。 父节点：Redirect

示例

请求示例

```
Get /?website HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqx*****k53otfjbyc: BuG4rRK+zNh*****1NNHD39zXw=
```

返回示例

- 已设置静态网站规则返回示例

```
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>
<HttpStatus>404</HttpStatus>
</ErrorDocument>
</WebsiteConfiguration>
```

- 未设置静态网站规则的返回示例

```
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 configuration.</Message>
<BucketName>oss-example</BucketName>
<RequestId>505191BEC4689A033D00236F</RequestId>
<HostId>oss-example.oss-cn-hangzhou.aliyuncs.com</HostId>
</Error>
```

完整返回示例

```
GET /?website HTTP/1.1
Date: Fri, 27 Jul 2018 09:07:41 GMT
Host: test.oss-cn-hangzhou-internal.aliyuncs.com
Authorization: OSS a1nBN*****QMf8u:0Jzamofmy*****sU9HUWomxsus=
User-Agent: aliyun-sdk-python-test/0.4.0
<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
<IndexDocument>
<Suffix>index.html</Suffix>
</IndexDocument>
<ErrorDocument>
<Key>error.html</Key>
<HttpStatus>404</HttpStatus>
</ErrorDocument>
<RoutingRules>
<RoutingRule>
```

```

<RoutingRule>
  <RuleNumber>1</RuleNumber>
  <Condition>
    <KeyPrefixEquals>abc</KeyPrefixEquals>
    <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
  </Condition>
  <Redirect>
    <RedirectType>Mirror</RedirectType>
    <PassQueryString>true</PassQueryString>
    <MirrorURL>http://example.com/</MirrorURL>
    <TransparentMirrorResponseCodes>400,404</TransparentMirrorResponseCodes>
    <MirrorPassQueryString>true</MirrorPassQueryString>
    <MirrorFollowRedirect>true</MirrorFollowRedirect>
    <MirrorCheckMd5>false</MirrorCheckMd5>
    <MirrorHeaders>
      <PassAll>true</PassAll>
      <Pass>myheader-key1</Pass>
      <Pass>myheader-key2</Pass>
      <Remove>myheader-key3</Remove>
      <Remove>myheader-key4</Remove>
      <Set>
        <Key>myheader-key5</Key>
        <Value>myheader-value5</Value>
      </Set>
    </MirrorHeaders>
  </Redirect>
</RoutingRule>
<RoutingRule>
  <RuleNumber>2</RuleNumber>
  <Condition>
    <IncludeHeader>
      <Key>host</Key>
      <Equals>test.oss-cn-beijing-internal.aliyuncs.com</Equals>
    </IncludeHeader>
    <KeyPrefixEquals>abc</KeyPrefixEquals>
    <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
  </Condition>
  <Redirect>
    <RedirectType>AliCDN</RedirectType>
    <Protocol>http</Protocol>
    <HostName>example.com</HostName>
    <PassQueryString>false</PassQueryString>
    <ReplaceKeyWith>prefix/${key}.suffix</ReplaceKeyWith>
    <HttpRedirectCode>301</HttpRedirectCode>
  </Redirect>
</RoutingRule>
</RoutingRules>
</WebsiteConfiguration>
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 27 Jul 2018 09:07:41 GMT
Content-Type: application/xml
Content-Length: 2102
Connection: keep-alive
x-oss-request-id: 5B5AF0DD2F7938C45FCFD4BA

```

```
x-oss-request-id: 800120021-1000101-000101010
x-oss-server-time: 47
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C++](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有相应的操作权限。只有Bucket的拥有者才可以查看Bucket的静态网站托管状态。
NoSuchWebsiteConfiguration	404	目标Bucket未设置静态网站托管功能。

6.11.3. DeleteBucketWebsite

DeleteBucketWebsite接口用于关闭存储空间（Bucket）的静态网站托管模式以及跳转规则。只有Bucket的拥有者才能关闭Bucket的静态网站托管模式。

请求语法

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

示例

请求示例

```
DELETE /?website HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Authorization: OSS qn6qrrqx*****k53otfjbyc:LnM4AZ1O*****5vGFWicOMekVg=
```

返回示例

```
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
```

完整代码

```
DELETE /?website HTTP/1.1
Date: Fri, 27 Jul 2018 09:10:52 GMT
Host: test.oss-cn-hangzhou-internal.aliyuncs.com
Authorization: OSS a1nBN*****QMf8u:qPrKwuM*****fk1pqTCylFs1jY=
User-Agent: aliyun-sdk-python-test/0.4.0
HTTP/1.1 204 No Content
Server: AliyunOSS
Date: Fri, 27 Jul 2018 09:10:52 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5B5AE19C188DC1CE81DAD7C8
```

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404 Not Found	目标Bucket不存在。
AccessDenied	403 Forbidden	没有关闭Bucket静态网站托管模式的权限。只有Bucket的拥有者才能关闭Bucket的静态网站托管模式。

6.12. 防盗链（Referer）

6.12.1. PutBucketReferer

PutBucketReferer接口用于设置存储空间（Bucket）的防盗链（Referer）。您可以使用此接口设置Referer的访问白名单以及是否允许Referer字段为空。

请求语法


```
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>
```

请求元素

名称	类型	是否必需	描述
RefererConfiguration	容器	是	保存Referer配置内容的容器。 子节点：AllowEmptyReferer、RefererList 父节点：无
AllowEmptyReferer	枚举字符串	是	指定是否允许Referer字段为空的请求访问。 取值：true（默认值）、false 父节点：RefererConfiguration
RefererList	容器	是	保存Referer访问白名单的容器。 ② 说明 PutBucketReferer为覆盖语义，即RefererList中的新指定的白名单列表将覆盖已配置的白名单列表。当您上传的RefererList为空时（即不包含Referer请求元素），此操作将清空RefererList中已配置的白名单列表。 父节点：RefererConfiguration 子节点：Referer
Referer	字符串	否	指定一条Referer访问白名单。 父节点：RefererList

示例

请求示例

- 不包含Referer的请求示例

```
PUT /?referer HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrTZH****
<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer>
<RefererList/>
</RefererConfiguration>
```

- 包含Referer的请求示例

```
PUT /?referer HTTP/1.1
Host: BucketName.oss.example.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrTZ****
<?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>
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906****
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403	没有操作权限。只有Bucket的拥有者才能发起PutBucketReferer请求。
InvalidDigest	400	上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性，如果不一致则返回此错误码。

6.12.2. GetBucketReferer

GetBucketReferer接口用于查看存储空间（Bucket）的防盗链（Referer）相关配置。

请求语法

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

响应元素

名称	类型	描述
RefererConfiguration	容器	保存Referer配置内容的容器。 父节点：无 子节点：AllowEmptyReferer、RefererList
AllowEmptyReferer	枚举字符串	指定是否允许Referer字段为空的请求访问。 取值： <i>true</i> 、 <i>false</i> 父节点：RefererConfiguration
RefererList	容器	保存Referer访问白名单的容器。 父节点：RefererConfiguration 子节点：Referer
Referer	字符串	指定一条Referer的访问白名单。 父节点：RefererList

示例

请求示例

```
Get /?referer HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51NNHD39****
```

返回示例

- 已设置Referer规则的返回示例

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>
```

- 未设置Referer规则的返回示例

 **说明** 如果Bucket未进行Referer相关配置，OSS会返回默认的AllowEmptyReferer值和空的RefererList。

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906****
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>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [PHP](#)
- [Python](#)
- [Go](#)
- [C](#)

- .NET
- Node.js
- Ruby

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有查看Bucket的Referer配置信息的权限。只有Bucket的拥有者才能查看Bucket的Referer配置信息。

6.13. 标签（Tags）

6.13.1. PutBucketTags

PutBucketTags接口用来给某个存储空间（Bucket）添加或修改标签。

注意事项

使用PutBucketTags接口时，有如下注意事项：

- 只有Bucket的拥有者及授权RAM账户才能为Bucket设置用户标签，否则返回403 Forbidden错误，错误码为AccessDenied。
- 最多可设置20对Bucket用户标签（Key-Value对）。
- PutBucketTags是覆盖语义，即新添加的标签会完全覆盖已有的标签。

请求语法

```
PUT /?tagging HTTP/1.1
Date: GMT Date
Content-Length: ContentLength
Authorization: SignatureValue
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
<?xml version="1.0" encoding="UTF-8"?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>key1</Key>
      <Value>value1</Value>
    </Tag>
    <Tag>
      <Key>key2</Key>
      <Value>value2</Value>
    </Tag>
  </TagSet>
</Tagging>
```

请求头

此接口仅涉及公共请求头，详情请参见[公共请求头（Common Request Headers）](#)。

请求元素

名称	类型	是否必需	描述
Tagging	容器	是	设置Bucket TagSet的容器。 子元素：TagSet 父元素：无
TagSet	容器	是	包含一系列Bucket Tag的容器。 子元素：Tag 父元素：Tagging
Tag	容器	是	设置Bucket Tag的容器。 子元素：Key、Value 父元素：TagSet
Key	字符串	是	指定Bucket Tag的Key。 - 最大长度为64字节。 - 不能以 <code>http://</code>、<code>https://</code>、<code>Aliyun</code> 为前缀。 - 必须为UTF-8编码； - 不能为空。 子元素：无 父元素：Tag
Value	字符串	否	指定Bucket Tag的Value。 - 最大长度为128字节。 - 必须为UTF-8编码。 - 可以为空。 子元素：无 父元素：Tag

响应头

此接口仅涉及公共响应头，详情请参见[公共响应头（Common Response Headers）](#)。

示例

- 请求示例

```
PUT /?tagging
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 20 Dec 2018 11:49:13 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
<Tagging>
  <TagSet>
    <Tag>
      <Key>testa</Key>
      <Value>testv1</Value>
    </Tag>
    <Tag>
      <Key>testb</Key>
      <Value>testv2</Value>
    </Tag>
  </TagSet>
</Tagging>
```

● 返回示例

```
200 (OK)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5C1B138A109F4E405B2D****
date: Thu, 20 Dec 2018 11:59:06 GMT
x-oss-server-time: 148
connection: keep-alive
```

6.13.2. GetBucketTags

GetBucketTags用于获取存储空间（Bucket）的标签信息。

请求语法

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

响应元素

名称	类型	描述
Tagging	容器	保存Bucket Tag结果的容器。 父元素：无  说明 如果源Bucket未设置Tag，OSS仍返回一个XML消息体，但其中的Tagging元素为空。

名称	类型	描述
TagSet	容器	保存Bucket Tag结果的容器。 父元素：Tagging
Tag	容器	保存Bucket Tag结果的容器。 父元素：TagSet
Key	字符串	某个Tag的Key。 父元素：Tag
Value	字符串	某个Tag的Value。 父元素：Tag

示例

- 请求示例

```
GET /?tagging
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 20 Dec 2018 13:09:13 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
```

- 返回示例

```
200 (OK)
content-length: 237
server: AliyunOSS
x-oss-request-id: 5C1B2D24B90AD5490CFE368E
date: Thu, 20 Dec 2018 13:12:21 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>testa</Key>
      <Value>value1-test</Value>
    </Tag>
    <Tag>
      <Key>testb</Key>
      <Value>value2-test</Value>
    </Tag>
  </TagSet>
</Tagging>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404	目标Bucket不存在。
AccessDenied	403	没有查看Bucket标签的权限。只有Bucket的拥有者及授权子账户才能查看Bucket的标签信息。

6.13.3. DeleteBucketTags

DeleteBucketTags接口用于删除存储空间（Bucket）标签。

 **说明** 如果目标Bucket没有任何标签或指定标签的Key不存在，则返回HTTP状态码204。

请求语法

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

请求头

此接口仅涉及公共请求头，详情请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头，详情请参见[公共响应头（Common Response Headers）](#)。

示例

- 示例1：删除所有标签

```
DELETE /?tagging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 25 Dec 2018 17:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:6ZVHOehYzxoC1yRydPQs/Cn****
```

返回示例

```
HTTP/1.1 204 No Content
x-oss-request-id: 5C22E0EFD127F6810B1A****
Date: Tue, 25 Dec 2018 17:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

- 示例2：删除Key为k1和k2的标签

```
DELETE /?tagging=k1,k2 HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 25 Dec 2018 17:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:6ZVHOehYzxC1yRydPQs/Cn****
```

返回示例

```
HTTP/1.1 204 No Content
x-oss-request-id: 5C22E0EFD127F6810B1A92A8
Date: Tue, 25 Dec 2018 17:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

SDK

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [Node.js](#)
- [.NET](#)
- [PHP](#)

错误码

错误码	HTTP状态码	描述
NoSuchBucket	404 No Content	目标Bucket不存在。
AccessDenied	403 Forbidden	无权限删除该Bucket的标签信息。 只有Bucket的拥有者才能删除Bucket的标签信息。

6.14. 加密（Encryption）

6.14.1. PutBucketEncryption

PutBucketEncryption接口用于配置存储空间（Bucket）的加密规则。

 **说明** 只有Bucket的拥有者及授权的RAM用户才能为Bucket设置加密规则，否则返回403错误。有关Bucket加密的更多信息，请参见[服务器端加密](#)。

请求语法

```
PUT /?encryption 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"?>
<ServerSideEncryptionRule>
  <ApplyServerSideEncryptionByDefault>
    <SSEAlgorithm>AES256</SSEAlgorithm>
    <KMSTMasterKeyID></KMSTMasterKeyID>
  </ApplyServerSideEncryptionByDefault>
</ServerSideEncryptionRule>
```

请求头

此接口涉及的公共请求头，例如Date、Host等。更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求元素

名称	类型	是否必选	示例值	描述
ServerSideEncryptionRule	容器	是	不涉及	服务器端加密规则的容器。 子元素： ApplyServerSideEncryptionByDefault
ApplyServerSideEncryptionByDefault	容器	是	不涉及	服务器端默认加密方式的容器。 子元素： SSEAlgorithm, KMSTMasterKeyID

名称	类型	是否必选	示例值	描述
SSEAlgorithm	字符串	是	AES256	设置服务器端默认加密方式。 取值：KMS、AES256、SM4 使用KMS密钥功能时会产生少量的KMS密钥API调用费用，费用详情请参见KMS计费标准。 进行跨区域复制时，若目标Bucket启用了默认服务器端加密方式，且复制规则配置了ReplicaCMKID，有以下两种情况： - 若源Bucket中的对象未加密，则使用目标Bucket的默认加密方式对跨区域复制过来的明文对象进行加密。 - 若源Bucket中的对象使用了SSE-KMS或SSE-OSS的加密方式，则目标Bucket针对这些对象仍然使用原加密方式进行加密。 更多信息，请参见跨区域复制结合服务器端加密。
KMSDataEncryption	字符串	否	SM4	指定Object的加密算法。若未指定此选项，表明Object使用AES256加密算法。此选项仅当SSEAlgorithm取值为KMS有效。 取值：SM4
KMSMasterKeyID	字符串	否	9468da86-3509-4f8d-a61e-6eab1eac****	当SSEAlgorithm值为KMS，且使用指定的密钥加密时，需输入KMSMasterKeyID。其他情况下，必须为空。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 请求示例

- 设置SSE-KMS加密

以下示例用于对名为oss-example的Bucket设置SSE-KMS加密。

```
PUT /?encryption HTTP/1.1
Date: Thur, 5 Nov 2020 11:09:13 GMT
Content-Length: ContentLength
Content-Type: application/xml
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
<?xml version="1.0" encoding="UTF-8"?>
<ServerSideEncryptionRule>
  <ApplyServerSideEncryptionByDefault>
    <SSEAlgorithm>KMS</SSEAlgorithm>
    <KMSTMasterKeyID>9468da86-3509-4f8d-a61e-6eab1eac****</KMSTMasterKeyID>
  </ApplyServerSideEncryptionByDefault>
</ServerSideEncryptionRule>
```

- 设置SM4加密

以下示例用于对名为oss-example的Bucket设置SM4加密。

```
PUT /?encryption HTTP/1.1
Date: Thur, 5 Nov 2020 11:09:13 GMT
Content-Length: ContentLength
Content-Type: application/xml
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
<?xml version="1.0" encoding="UTF-8"?>
<ServerSideEncryptionRule>
  <ApplyServerSideEncryptionByDefault>
    <SSEAlgorithm>SM4</SSEAlgorithm>
    <KMSTMasterKeyID></KMSTMasterKeyID>
  </ApplyServerSideEncryptionByDefault>
</ServerSideEncryptionRule>
```

您还可以在上传Object时，通过在请求头中携带 `x-oss-server-side-encryption` 并指定其值为SM4，表明该Object使用SM4加密算法。

- 设置KMS-SM4加密

以下示例用于对名为oss-example的Bucket设置KMS-SM4加密。

```
PUT /?encryption HTTP/1.1
Date: Thur, 5 Nov 2020 11:09:13 GMT
Content-Length: ContentLength
Content-Type: application/xml
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
<?xml version="1.0" encoding="UTF-8"?>
<ServerSideEncryptionRule>
  <ApplyServerSideEncryptionByDefault>
    <SSEAlgorithm>KMS</SSEAlgorithm>
    <KMSEncryption>SM4</KMSEncryption>
    <KMSMasterKeyID></KMSMasterKeyID>
  </ApplyServerSideEncryptionByDefault>
</ServerSideEncryptionRule>
```

您还可以在上传Object时，通过在请求头中携带 `x-oss-server-side-encryption` 并指定其值为KMS，`x-oss-server-side-data-encryption` 指定其值为SM4，表明该Object使用KMS-SM4加密算法。

- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5C1B138A109F4E405B2D****
Date: Thur, 5 Nov 2020 11:09:13 GMT
```

SDK

PutBucketEncryption接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	说明
InvalidEncryptionAlgorithmError	400	当SSEAlgorithm取值不为KMS或者AES256时，则返回此错误。错误提示为 <code>The Encrvption request you specified is not valid. Supported value: AES256/KMS</code> 。
InvalidArgument	400	当SSEAlgorithm取值为AES256，但填写了KMSMasterKeyID，则返回此错误，错误提示为 <code>KMSMasterKeyID is not applicable if the default sse algorithm is not KMS</code> 。

6.14.2. GetBucketEncryption

GetBucketEncryption接口用于获取存储空间（Bucket）的加密规则。

 **说明** 只有Bucket的拥有者及授权的RAM用户才能获取Bucket的加密规则，否则返回403错误。有关Bucket加密的更多信息，请参见[服务器端加密](#)。

请求语法

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

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

响应元素

名称	类型	示例值	描述
ServerSideEncryptionRule	容器	不涉及	服务端加密规则的容器。 子元素： ApplyServerSideEncryptionByDefault
ApplyServerSideEncryptionByDefault	容器	不涉及	服务端默认加密方式的容器。 子元素：SSEAlgorithm, KMSMasterKeyID
SSEAlgorithm	字符串	KMS	显示服务端默认加密方式。 取值：KMS、AES256、SM4
KMSDataEncryption	字符串	SM4	Object的加密算法。若响应中未包含此元素，表明Object使用AES256加密算法。此选项仅当SSEAlgorithm取值为KMS有效。 取值：SM4
KMSMasterKeyID	字符串	9468da86-3509-4f8d-a61e-6eab1eac****	显示当前使用的KMS密钥ID。 仅当SSEAlgorithm为KMS且指定了密钥ID时返回，其他情况下，此项为空。

示例

- 请求示例

```
Get /?encryption HTTP/1.1
Date: Tue, 20 Dec 2018 11:20:10 GMT
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otf****:ceOEyZavKY4QcjoUWYSpYbJ3****
```

- 返回示例

以下返回示例表明Bucket设置了SSE-KMS加密。

```
HTTP/1.1 204 NoContent
x-oss-request-id: 5C1B138A109F4E405B2D8AEF
Date: Tue, 20 Dec 2018 11:22:05 GMT
<?xml version="1.0" encoding="UTF-8"?>
<ServerSideEncryptionRule>
  <ApplyServerSideEncryptionByDefault>
    <SSEAlgorithm>KMS</SSEAlgorithm>
    <KMSEncryptionContext>9468da86-3509-4f8d-a61e-6eab1eac****</KMSEncryptionContext>
  </ApplyServerSideEncryptionByDefault>
</ServerSideEncryptionRule>
```

以下返回示例表明Bucket设置了KMS-SM4加密。

```
HTTP/1.1 200 OK
x-oss-request-id: 5C1B138A109F4E405B2D8AEF
Date: Tue, 20 Dec 2018 11:22:05 GMT
<?xml version="1.0" encoding="UTF-8"?>
<ServerSideEncryptionRule>
  <ApplyServerSideEncryptionByDefault>
    <SSEAlgorithm>KMS</SSEAlgorithm>
    <KMSEncryptionContext>SM4</KMSEncryptionContext>
    <KMSEncryptionContext>9468da86-3509-4f8d-a61e-6eab1eac****</KMSEncryptionContext>
  </ApplyServerSideEncryptionByDefault>
</ServerSideEncryptionRule>
```

SDK

GetBucketEncryption接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	说明
AccessDenied	403	无获取Bucket加密规则的权限。

错误码	HTTP状态码	说明
NoSuchBucket	400	指定获取加密规则对应的Bucket不存在。
NoSuchServerSideEncryptionRule	400	Bucket未设置加密规则。

6.14.3. DeleteBucketEncryption

DeleteBucketEncryption接口用于删除Bucket加密规则。

 **说明** 只有Bucket的拥有者及授权的RAM用户才能删除Bucket的加密规则，否则返回403错误。有关Bucket加密的更多信息，请参见[服务器端加密](#)。

请求语法

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

示例

- 请求示例

```
DELETE /?encryption HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 20 Dec 2018 11:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:6ZVHOehYzxC1yxRydPQs/Cn****
```

- 响应示例

```
HTTP/1.1 204 OK
x-oss-request-id: 5C22E0EFD127F6810B1A****
Date: Tue, 20 Dec 2018 11:37:05 GMT
Connection: keep-alive
Content-Length: 0
```

SDK

DeleteBucketEncryption接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	说明
AccessDenied	403	无删除Bucket加密规则的权限。
NoSuchBucket	400	指定删除加密规则对应的Bucket不存在。

6.15. 请求者付费（RequestPayment）

6.15.1. PutBucketRequestPayment

PutBucketRequestPayment接口用于设置请求者付费模式。

注意事项

使用请求者付费模式时，有如下注意事项：

- 不允许匿名访问

如果您在Bucket上启用了请求者付费模式，则不允许匿名访问该Bucket。请求方必须提供身份验证信息，以便OSS能够识别请求方，从而对请求方而非Bucket拥有者收取请求所产生的费用。

当请求者是通过扮演阿里云RAM角色来请求数据时，该角色所属的账户将为此请求付费。

- 申请方需携带x-oss-request-payer信息

如果您在Bucket上启用了请求者付费模式，请求方必须在其请求中包含x-oss-request-payer:requester（在POST、GET和HEAD请求的Header信息中），以表明请求方已知悉请求和数据下载将产生费用。否则，请求方无法通过验证。

数据拥有者访问该Bucket时，可以不携带x-oss-request-payer请求头。数据拥有者作为请求者访问该Bucket时，请求产生的费用由数据拥有者（也是请求者）来支付。

请求语法

```
PUT /?requestPayment 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"?>
<RequestPaymentConfiguration>
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

请求头

此接口仅使用公共请求头部，详情请参见[公共请求头（Common Request Headers）](#)。

请求元素

名称	类型	是否必选	描述
RequestPaymentConfiguration	容器	是	请求付费配置的容器。 子节点：Payer
Payer	字符串	是	指定Bucket付费类型。 取值： <i>BucketOwner</i>: 由Bucket拥有者付费。 <i>Requester</i>: 由请求者付费。 父节点： RequestPaymentConfiguration

响应头

此接口仅返回公共响应头部，详情请参见[公共响应头（Common Response Headers）](#)。

示例

请求示例

```
PUT /?requestPayment
Content-Length: 83
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 23 Jul 2019 01:33:47 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:77Dvh5wQgljWjwO/KyRt8dOP****
<RequestPaymentConfiguration>
  <Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

返回示例

```
200 (OK)
content-length: 0
x-oss-request-id: 5D3663FBB007B79097FC****
date: Tue, 23 Jul 2019 01:33:47 GMT
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	访问的Bucket不存在。

6.15.2. GetBucketRequestPayment

GetBucketRequestPayment接口用于获取请求者付费模式配置信息。

请求语法

```
GET /?requestPayment HTTP/1.1
Date: GMT Date
Host: BucketName.oss.aliyuncs.com
Authorization: authorization string
```

响应元素

名称	类型	描述
RequestPaymentConfiguration	容器	请求付费配置的容器。 子节点: Payer
Payer	字符串	指定Bucket付费类型。 取值: <i>BucketOwner</i> 、 <i>Requester</i> 父节点: RequestPaymentConfiguration

示例

请求示例

```
GET /?requestPayment
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 23 Jul 2019 01:23:20 GMT
Authorization: OSS LTAICQmy1*****:Rk7tT30yhDO3D*****
```

返回示例

```
200 (OK)
content-length: 129
server: AliyunOSS
x-oss-request-id: 5D366188B007B79097EC****
date: Tue, 23 Jul 2019 01:23:20 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration>
  <Payer>BucketOwner</Payer>
</RequestPaymentConfiguration>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	访问的Bucket不存在。

6.16. 跨域资源共享（CORS）

6.16.1. 简介

跨域资源共享（CORS）允许Web端的应用程序访问不属于本域的资源。

OSS支持CORS标准，方便您更灵活地开发Web应用程序。OSS提供了如下接口方便您控制跨域访问的各种权限：

- [PutBucketCORS](#)
- [GetBucketCORS](#)
- [DeleteBucketCORS](#)
- [OptionObject](#)

6.16.2. PutBucketCors

PutBucketCors接口用于为指定的存储空间（Bucket）设置跨域资源共享CORS（Cross-Origin Resource Sharing）规则。

注意事项

使用跨域资源共享时，有如下注意事项：

- 默认不开启CORS
Bucket默认不开启CORS功能，所有跨域请求的Origin都不被允许。

- 覆盖语义

PutBucketCors为覆盖语义，即新配置的CORS规则将覆盖已有的CORS规则。

- 应用程序中使用CORS

在应用程序中使用CORS功能时，需通过PutBucketCors接口手动上传CORS规则来开启CORS功能。

例如，从 example.com 通过浏览器的 XMLHttpRequest 功能来访问OSS，需要通过本接口手动上传CORS规则，且CORS规则需由XML文档进行描述。

- CORS规则匹配

当OSS收到一个跨域请求或OPTIONS请求，会先读取Bucket对应的CORS规则，然后进行相应的权限检查。OSS会依次检查每一条规则，使用第一条匹配的规则来允许请求并返回对应的Header。如果所有规则都匹配失败，则不附加任何CORS相关的Header。

CORS规则匹配成功必须满足以下三个条件：

- 请求的Origin必须匹配一个 AllowedOrigin 项。
- 请求的方法（如GET、PUT等）或者OPTIONS请求的 Access-Control-Request-Method 头对应的方法必须匹配一个 AllowedMethod 项。
- OPTIONS请求的 Access-Control-Request-Headers 头包含的每个header都必须匹配一个 AllowedHeader 项。

请求语法

```
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-flight response</MaxAgeSeconds>
  </CORSRule>
  <CORSRule>
    ...
  </CORSRule>
  ...
</CORSConfiguration>
```

请求元素

名称	类型	是否必选	示例值	描述
CORSRule	容器	是	不涉及	CORS规则的容器。 每个Bucket最多允许10条CORS规则。上传的XML文档最大允许16 KB。 父节点：CORSConfiguration
AllowedOrigin	字符串	是	*	指定允许的跨域请求来源。 - OSS允许使用多个元素来指定多个允许的来源。 - OSS仅允许使用一个星号（*）通配符。如果指定为星号（*），则表示允许所有来源的跨域请求。 父节点：CORSRule
AllowedMethod	枚举值	是	GET	指定允许的跨域请求方法。 取值：GET、PUT、DELETE、POST、HEAD 父节点：CORSRule
AllowedHeader	字符串	否	Authorization	控制OPTIONS预取指令 Access-Control-Request-Headers 中指定的Header是否被允许。 在 Access-Control-Request-Headers 中指定的每个Header都必须在AllowedHeader中有对应的项。  说明 仅允许使用一个星号（*）通配符。 父节点：CORSRule
ExposeHeader	字符串	否	x-oss-test	指定允许用户从应用程序中访问的响应头。例如，一个JavaScript的XMLHttpRequest对象。  说明 不允许使用星号（*）通配符。 父节点：CORSRule
MaxAgeSeconds	整型	否	100	指定浏览器对特定资源的预取（OPTIONS）请求返回结果的缓存时间。单位为秒。 单条CORS规则仅允许一个MaxAgeSeconds。 父节点：CORSRule

名称	类型	是否必选	示例值	描述
CORSConfiguration	容器	是	不涉及	Bucket的CORS规则容器。 父节点：无
ResponseVary	布尔	否	false	是否返回Vary: Origin头。 取值： <i>true</i>：不管发送的是否是跨域请求或跨域请求是否成功，均会返回Vary: Origin头。 <i>false</i>（默认值）：任何情况下均不返回Vary: Origin头 ❓ 说明 此字段不能单独配置，必须至少配置一项跨域规则才能生效。

此接口涉及的Host、Authorization等其他更多公共请求头的更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅返回公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 请求示例

```
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 qn6qrrqxo2oawuk53otf****:KU5h8YMUC78M30dXqf3JxrT*****
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader>Authorization</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>http://example.com</AllowedOrigin>
    <AllowedOrigin>http://example.net</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <AllowedHeader> Authorization</AllowedHeader>
    <ExposeHeader>x-oss-test</ExposeHeader>
    <ExposeHeader>x-oss-test1</ExposeHeader>
    <MaxAgeSeconds>100</MaxAgeSeconds>
  </CORSRule>
  <ResponseVary>false</ResponseVary>
</CORSConfiguration>
```


- 返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 50519080C4689A033D0*****
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C++](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
InvalidDigest	400	上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性，如果不一致则返回此错误码。

6.16.3. GetBucketCors

GetBucketCors接口用于获取指定存储空间（Bucket）当前的跨域资源共享CORS（Cross-Origin Resource Sharing）规则。

请求语法

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

请求头

此接口仅使用公共请求头部，详情请参见[公共请求头（Common Request Headers）](#)。

响应元素

名称	类型	描述
CORSRule	容器	CORS规则的容器。每个Bucket最多允许10条规则。 父节点： CORSConfiguration
AllowedOrigin	字符串	允许的跨域请求来源。星号（*）表示允许所有的来源的跨域请求。 父节点： CORSRule
AllowedMethod	枚举 (GET,PUT,DELETE, POST,HEAD)	允许的跨域请求方法。 父节点： CORSRule
AllowedHeader	字符串	控制在OPTIONS预取指令中Access-Control-Request-Headers指定的header是否允许。在Access-Control-Request-Headers中指定的每个header都在AllowedHeader中有一条对应的项。 父节点： CORSRule
ExposeHeader	字符串	允许用户从应用程序中访问的响应头（例如一个JavaScript的XMLHttpRequest对象）。 父节点： CORSRule
MaxAgeSeconds	整型	浏览器对特定资源的预取（OPTIONS）请求返回结果的缓存时间。一个CORS规则里面最多允许出现一个MaxAgeSeconds。 单位：秒 父节点： CORSRule
CORSConfiguration	容器	Bucket的CORS规则容器。 父节点： 无

示例

请求示例

```
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+zNhH1AcF51NNHD39****
```

返回示例

```
HTTP/1.1 200
x-oss-request-id: 50519080C4689A033D00****
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>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C++](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。
NoSuchCORSConfiguration	404	目标CORS规则不存在。
AccessDenied	403	只有Bucket的拥有者才能获取CORS规则。

6.16.4. DeleteBucketCors

DeleteBucketCors用于关闭指定存储空间（Bucket）对应的跨域资源共享CORS（Cross-Origin Resource Sharing）功能并清空所有规则。

请求语法

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

请求头

此接口仅使用公共请求头部，详情请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅返回公共响应头部，详情请参见[公共响应头（Common Response Headers）](#)。

示例

请求示例

```
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:LnM4AZ1OelduZF5vGFWicOME****
```

返回示例

```
HTTP/1.1 204 No Content
x-oss-request-id: 5051845BC4689A033D00****
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C++](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
NoSuchBucket	404	目标Bucket不存在。

错误码	HTTP 状态码	描述
AccessDenied	403	没有删除权限。只有Bucket的拥有者才有权限删除Bucket对应的CORS规则。

6.16.5. Options

浏览器在发送跨域请求之前会发送一个preflight请求（Options）给OSS，并带上特定的来源域、HTTP方法和header等信息，以决定是否发送真正的请求。Options请求是由浏览器自动根据是否跨域来决定是否发送。

请求语法

```
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
```

请求头

名称	类型	是否必选	示例值	描述
Origin	字符串	是	http://www.example.com	请求来源域，用于标识跨域请求。 在实际请求中只能设置一个该请求头。 默认值：无
Access-Control-Request-Method	字符串	是	PUT	在实际请求中会用到的方法。 在实际请求中只能设置一个该请求头。 默认值：无
Access-Control-Request-Headers	字符串	否	x-oss-test1,x-oss-test2	在实际请求中会用到的除了简单头部之外的header。 在实际请求中可以为该请求头设置多个header，多个header之间使用英文逗号(,)隔开。 默认值：无

响应头

名称	类型	示例值	描述
Access-Control-Allow-Origin	字符串	http://www.example.com	请求中包含的Origin。如果不允许该请求，则响应头不包含该头部。

名称	类型	示例值	描述
Access-Control-Allow-Methods	字符串	PUT	允许请求的HTTP方法。如果不允许该请求，则响应头不包含该头部。
Access-Control-Allow-Headers	字符串	x-oss-test,x-oss-test1	允许请求携带的header的列表。如果请求中有不被允许的header，则不包含该头部，请求也将被拒绝。
Access-Control-Expose-Headers	字符串	x-oss-test1,x-oss-test2	允许在客户端JavaScript程序中访问的headers的列表。
Access-Control-Max-Age	整型	60	允许浏览器缓存preflight结果的时间，单位为秒。

示例

请求示例

```
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-test1,x-oss-test2
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5051845BC4689A033D00****
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-test1,x-oss-test2
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

错误码

错误码	HTTP 状态码	描述
AccessForbidden	403	OSS可以通过PutBucketCORS接口来开启Bucket的CORS功能。开启CORS功能后，OSS在收到浏览器preflight请求时会根据设定的规则评估是否允许本次请求，如果不允许或者CORS功能未开启，则返回此错误码。

7.关于Object操作

7.1. 基础操作

7.1.1. PutObject

调用PutObject接口上传文件（Object）。

注意事项

- 添加的Object大小不能超过5 GB。
- 默认情况下，如果已存在同名Object且对该Object有访问权限，则新添加的Object将覆盖原有的Object，并返回200 OK。
- OSS没有文件夹的概念，所有资源都是以文件来存储，但您可以通过创建一个以正斜线（/）结尾，大小为0的Object来创建模拟文件夹。

版本控制

在已开启版本控制的Bucket中，OSS会为新添加的Object自动生成唯一的版本ID，并在响应Header中通过x-oss-version-id形式返回。

在暂停了版本控制的Bucket中，新添加的Object的版本ID为null。OSS会保证同一个Object仅有一个null的版本ID。

请求语法

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

请求头

OSS支持HTTP协议规定的5个请求头Cache-Control、Expires、Content-Encoding、Content-Disposition、Content-Type。如果上传Object时设置了这些请求头，则下载该Object时，相应的请求头的值会自动使用上传Object时设置的值。

名称	类型	是否必选	描述
Authorization	字符串	否	表示请求本身已被授权。更多信息，请参见RFC2616。关于Authorization计算方法的更多信息，请参见在Header中包含签名。 通常情况下Authorization是必选请求头，但如果采用了URL包含签名，则不用携带该请求头。更多信息，请参见在URL中包含签名。 默认值：无

名称	类型	是否必选	描述
Cache-Control	字符串	否	指定该Object被下载时网页的缓存行为。更多信息，请参见 RFC2616 。 默认值：无
Content-Disposition	字符串	否	指定该Object被下载时的名称。更多信息，请参见 RFC2616 。 默认值：无
Content-Encoding	字符串	否	指定该Object被下载时的内容编码格式。更多信息，请参见 RFC2616 。 默认值：无
Content-MD5	字符串	否	用于检查消息内容是否与发送时一致。Content-MD5是由MD5算法生成的值。上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性。更多信息，请参见 Content-MD5的计算方法 。 为确保数据完整性，OSS提供多种方式对数据的MD5值进行校验。如果需要通过Content-MD5进行MD5验证，可将Content-MD5加入到请求头中。 默认值：无
Content-Length	字符串	否	用于描述HTTP消息体的传输大小。 如果请求头中的Content-Length值小于实际请求体中传输的数据大小，OSS仍将成功创建Object，但Object的大小只能等于Content-Length中定义的大小，其他数据将被丢弃。
ETag	字符串	否	Object生成时会创建相应的ETag，ETag用于标识一个Object的内容。 - 对于PutObject请求创建的Object，ETag值是其内容的MD5值。 - 对于其他方式创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。  说明 ETag值可以用于检查Object内容是否发生变化。不建议使用ETag作为Object内容的MD5来校验数据完整性。 默认值：无
Expires	字符串	否	过期时间。更多信息，请参见 RFC2616 。 默认值：无

名称	类型	是否必选	描述
x-oss-forbid-overwrite	字符串	否	指定PutObject操作时是否覆盖同名Object。当目标Bucket处于已开启或已暂停的版本控制状态时，x-oss-forbid-overwrite请求Header设置无效，即允许覆盖同名Object。 - 不指定x-oss-forbid-overwrite或者指定x-oss-forbid-overwrite为<code>false</code>时，表示允许覆盖同名Object。 - 指定x-oss-forbid-overwrite为<code>true</code>时，表示禁止覆盖同名Object。 设置x-oss-forbid-overwrite请求Header会导致QPS处理性能下降，如果您有大量的操作需要使用x-oss-forbid-overwrite请求Header（QPS>1000），请联系技术支持，避免影响您的业务。 默认值：<code>false</code>
x-oss-server-side-encryption	字符串	否	创建Object时，指定服务器端加密方式。 取值：<code>AES256</code>、<code>KMS</code>或<code>SM4</code> 指定此选项后，在响应头中会返回此选项，OSS会对上传的Object进行加密编码存储。当下载该Object时，响应头中会包含x-oss-server-side-encryption，且该值会被设置成此Object的加密算法。
x-oss-server-side-data-encryption	字符串	否	指定Object的加密算法。如果未指定此选项，表明Object使用AES256加密算法。此选项仅当x-oss-server-side-encryption为KMS时有效。 取值：<code>SM4</code>
x-oss-server-side-encryption-key-id	字符串	否	KMS托管的用户主密钥。 此选项仅在x-oss-server-side-encryption为KMS时有效。

名称	类型	是否必选	描述
x-oss-object-acl	字符串	否	指定OSS创建Object时的访问权限。 取值： <i>default</i>（默认）：Object遵循所在存储空间的访问权限。 <i>private</i>：Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户没有权限操作该Object。 <i>public-read</i>：Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户只有该Object的读权限。请谨慎使用该权限。 <i>public-read-write</i>：Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息，请参见基于读写权限ACL的权限控制。
x-oss-storage-class	字符串	否	指定Object的存储类型。 对于任意存储类型的Bucket，如果上传Object时指定此参数，则此次上传的Object将存储为指定的类型。例如在IA类型的Bucket中上传Object时，如果指定x-oss-storage-class为Standard，则该Object直接存储为Standard。 取值： <i>Standard</i>：标准存储 <i>IA</i>：低频访问 <i>Archive</i>：归档存储 <i>ColdArchive</i>：冷归档存储 关于存储类型的更多信息，请参见存储类型介绍。
x-oss-meta-*	字符串	否	使用PutObject接口时，如果配置以x-oss-meta-*为前缀的参数，则该参数视为元数据，例如 x-oss-meta-location。一个Object可以有多个类似的参数，但所有的元数据总大小不能超过8 KB。 元数据支持短划线（-）、数字、英文字母（a~z）。英文字符的大写字母会被转成小写字母，不支持下划线（_）在内的其他字符。
x-oss-tagging	字符串	否	指定Object的标签，可同时设置多个标签，例如 TagA=A&TagB=B。  说明 Key和Value需要先进行URL编码，如果某项没有" = "，则看作Value为空字符串。

此接口还需要包含Host、Date等公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅包含公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 简单上传

请求示例

```
PUT /test.txt HTTP/1.1
Host: test.oss-cn-zhangjiakou.aliyuncs.com
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: */*
Connection: keep-alive
Content-Type: text/plain
date: Tue, 04 Dec 2018 15:56:37 GMT
authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2P****
Transfer-Encoding: chunked
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 04 Dec 2018 15:56:38 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5C06A3B67B8B5A3DA422299D
ETag: "D41D8CD98F00B204E9800998ECF8****"
x-oss-hash-crc64ecma: 0
Content-MD5: 1B2M2Y8AsgTpgAmY7PhCfg==
x-oss-server-time: 7
```

- 带有归档存储类型

请求示例

```
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/jpeg
Content-Length: 344606
x-oss-storage-class: Archive
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2P****
[344606 bytes of object data]
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Sat, 21 Nov 2015 18:52:34 GMT
Content-Type: image/jpg
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5650BD72207FB30443962F9A
x-oss-bucket-version: 1418321259
ETag: "A797938C31D59EDD08D86188F6D5B872"
```

- 已开启版本控制

请求示例

```
PUT /test HTTP/1.1
Content-Length: 362149
Content-Type: text/html
Host: versioning-put.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:53:24 GMT
Authorization: OSS lkojgn8y1exic6e:6yYhX+BuuEqzl1tAMW0wglyl****
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 09 Apr 2019 02:53:24 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5CAC0A3DB7AEADE01700****
x-oss-version-id: CAEQNhiBgMDJgZCA0BYiIDc4MGZjZGI2OTBjOTRmNTE5NmU5NmFhZjhjYmY0****
ETag: "4F345B1F066DB1444775AA97D5D2****"
```

SDK

PutObject接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)
- [Browser.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
MissingContentLength	411	请求头没有采用chunked encoding编码方式，或没有设置Content-Length参数。
InvalidEncryptionAlgorithmError	400	指定x-oss-server-side-encryption的值无效。 取值：AES256、KMS或SM4。
AccessDenied	403	添加Object时，用户对设置的Bucket没有访问权限。
NoSuchBucket	404	添加Object时，设置的Bucket不存在。
InvalidObjectName	400	传入的Object key长度大于1023字节。
InvalidArgument	400	返回该错误的可能原因如下： - 添加的Object大小超过5 GB。 - x-oss-storage-class等参数的值无效。
RequestTimeout	400	指定了Content-Length，但没有发送消息体，或者发送的消息体小于指定的大小。此种情况下服务器会一直等待，直至请求超时。
Bad Request	400	在请求中指定Content-MD5后，OSS会计算所发送数据的MD5值，并与请求中Content-MD5的值进行比较。如果二者不一致，则返回该错误。
KmsServiceNotEnabled	403	将x-oss-server-side-encryption指定为KMS，但没有预先购买KMS套件。
FileAlreadyExists	409	当请求的Header中携带 x-oss-forbid-overwrite=true 时，表示禁止覆盖同名文件。如果同名文件已存在，则返回该错误。
FileImmutable	409	Bucket中的数据处于被保护状态时，如果尝试删除或修改相应数据，则返回该错误。

7.1.2. GetObject

GetObject接口用于获取某个文件（Object）。此操作需要对此Object具有读权限。

注意事项

- GetObject接口默认可通过HTTP和HTTPS两种方式访问。如果要设置仅允许通过HTTPS方式访问，请使用Bucket Policy的授权访问方式。具体操作，请参见[通过Bucket Policy授权用户访问指定资源](#)。
- 如果Object类型为归档类型，需要先完成解冻文件（RestoreObject）请求，且该请求不能超时。

版本控制

默认情况下，调用GetObject接口仅返回Object的当前版本。

如果在查询参数中指定Object的versionId，则返回指定的Object版本。当versionId指定为null时，则返回versionId为null的Object版本。

请求语法

```
GET /ObjectName HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
Range: bytes=ByteRange(可选)
```

下载OSS中的大文件（超过100 MB）时，如果传输过程中受到网络环境影响，则会传输失败。您可以通过 HTTP Range请求获取大文件的部分内容。

OSS不支持多Range参数，即不支持指定多个范围。ByteRange指请求资源的范围，单位为Byte（字节），ByteRange有效区间在0至 object size - 1 的范围内。具体示例如下：

- Range: bytes=0-499 表示第0~499字节范围的内容。
- Range: bytes=500-999 表示第500~999字节范围的内容。
- Range: bytes=-500 表示最后500字节的内容。
- Range: bytes=500- 表示从第500字节开始到文件结束部分的内容。
- Range: bytes=0- 表示从第一个字节到最后一个字节，即完整的文件内容。

 **说明** 有关Rangeget兼容性的更多信息，请参见[兼容行为说明](#)。

请求头

OSS支持在GET请求中通过请求头来自定义响应头，但只有请求成功（即返回码为200 OK）才会将响应头的值设置成GET请求头中指定的值。

 **说明** OSS不支持在匿名访问的GET请求中自定义响应头。

名称	类型	是否必选	描述
response-content-type	字符串	否	指定OSS返回请求的content-type头。 默认值：无
response-content-language	字符串	否	指定OSS返回请求的content-language头。 默认值：无
response-expires	字符串	否	指定OSS返回请求的expires头。 默认值：无
response-cache-control	字符串	否	指定OSS返回请求的cache-control头。 默认值：无
response-content-disposition	字符串	否	指定OSS返回请求的content-disposition头。 默认值：无

名称	类型	是否必选	描述
response-content-encoding	字符串	否	指定OSS返回请求的content-encoding头。 默认值：无
Range	字符串	否	指定文件传输的范围。 - 如果指定的范围符合规范，返回消息中会包含整个Object的大小和此次返回Object的范围。例如：Content-Range: bytes 0~9/44，表示整个Object大小为44，此次返回的范围为0~9。 - 如果指定的范围不符合规范，则传送整个Object，并且结果中不包含Content-Range。 默认值：无
If-Modified-Since	字符串	否	如果指定的时间早于实际修改时间或指定的时间不符合规范，则直接返回Object，并返回200 OK；如果指定的时间等于或者晚于实际修改时间，则返回304 Not Modified。 时间格式：GMT，例如 Fri, 13 Nov 2015 14:47:53 GMT 默认值：无
If-Unmodified-Since	字符串	否	如果指定的时间等于或者晚于Object实际修改时间，则正常传输Object，并返回200 OK；如果指定的时间早于实际修改时间，则返回412 Precondition Failed。 时间格式：GMT，例如 Fri, 13 Nov 2015 14:47:53 GMT If-Modified-Since和If-Unmodified-Since可以同时使用。 默认值：无
If-Match	字符串	否	如果传入的ETag和Object的ETag匹配，则正常传输Object，并返回200 OK；如果传入的ETag和Object的ETag不匹配，则返回412 Precondition Failed。 Object的ETag值用于验证数据是否发生了更改，您可以基于ETag值验证数据完整性。 默认值：无
If-None-Match	字符串	否	如果传入的ETag值和Object的ETag不匹配，则正常传输Object，并返回200 OK；如果传入的ETag和Object的ETag匹配，则返回304 Not Modified。 If-Match和If-None-Match可以同时使用。 默认值：无

名称	类型	是否必选	描述
Accept-Encoding	字符串	否	指定客户端的编码类型。 如果要对返回内容进行Gzip压缩传输，您需要在请求头中以显示方式加入Accept-Encoding:gzip。OSS会根据Object的Content-Type和Object大小（不小于1 KB）判断是否返回经过Gzip压缩的数据。  说明 - 如果采用了Gzip压缩，则不会附带ETag信息。 - 目前OSS支持Gzip压缩的Content-Type为text/cache-manifest、text/xml、text/plain、text/css、application/javascript、application/x-javascript、application/rss+xml、application/json和text/json。 默认值：无

响应头

如果Object类型为软链接，则返回目标Object的内容。响应头中 **Content-Length**、**ETag**、**Content-Md5** 为目标Object的元数据；**Last-Modified** 取目标Object和软链接对应的最大值（即在两者中取更新较晚的时间）；其他均为软链接的元数据。

名称	类型	描述
x-oss-server-side-encryption	字符串	若Object在服务器端采用熵编码加密存储，使用GET请求时，系统会自动解密返回给用户，并且在响应头中返回x-oss-server-side-encryption，表明该Object的服务器端加密算法。
x-oss-tagging-count	字符串	对象关联的标签的个数。仅当有读取标签权限时返回。

示例

- 简单的GET请求

请求示例

```
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:UNQDb7GapEgJkcde6OhZ9J*****
```

返回示例


```
HTTP/1.1 200 OK
x-oss-request-id: 3a8f-2e2d-7965-3ff9-51c875b*****
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: "5B3C1A2E0563E1B002CC607C*****"
Content-Type: image/jpg
Content-Length: 344606
Server: AliyunOSS
[344606 bytes of object data]
```

- 带有Range参数

请求示例

```
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 qn6qrrqxo2oawuk5jbyc:qZzjF3DUtd+yK16BdhGtFcC*****
```

返回示例

```
HTTP/1.1 206 Partial Content
x-oss-request-id: 28f6-15ea-8224-234e-c0ce407*****
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: "5B3C1A2E05E1B002CC607C*****"
Accept-Ranges: bytes
Content-Range: bytes 100-900/344606
Content-Type: image/jpg
Content-Length: 801
Server: AliyunOSS
[801 bytes of object data]
```

- 带自定义返回消息头

请求示例

```
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
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC75A644*****
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: "5B3C1A2E053D1B002CC607*****"
Content-Length: 344606
Connection: keep-alive
Content-disposition: attachment; filename:testing.txt
Content-language: 中文
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]
```

- Object类型为软链接

请求示例

```
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 qn6qrrqxok53otfjbyc:qZzjF3DUtd+yK16BdhGtFc*****
```

返回示例

```
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: 582143E6A212AD*****
Accept-Ranges: bytes
ETag: "8086265EFC021F9A2F09BF4*****"
Last-Modified: Tue, 08 Nov 2016 03:17:58 GMT
x-oss-object-type: Symlink
Content-MD5: gIYmXvwCEe0fmi8Jv0Y****
```

- Restore操作已完成

请求示例

```
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 qn6qrrqx02o***k53otfjbyc:zUglwRPGkbByZxm1+y4eyu+*****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 58F723829F29F18D7F00*****
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: "5B3C1A2E0763E1B002CC607C*****"
Content-Type: image/jpg
Content-Length: 344606
Server: AliyunOSS
[354606 bytes of object data]
```

- 指定versionId访问指定版本Object

请求示例

```
GET /example?versionId=CAEQNhiBgMDJgZCA0BYiIDc4MGZjZGI2OTBjOTRmNTE5NmU5NmFhZjhjYmY0***
* HTTP/1.1
Host: versioning-get.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 02:58:06 GMT
Authorization: OSS lkojgxic6e:8wcOrEDt4iSxpBPfQW9OJNw*****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC0A3EDE0170*****
x-oss-version-id: CAEQNhiBgM0BYiIDc4MGZjZGI2OTBjOTRmNTE5NmU5NmFhZjhjYmY*****
x-oss-object-type: Normal
Date: Tue, 09 Apr 2019 02:58:06 GMT
Last-Modified: Fri, 22 Mar 2018 08:07:50 GMT
ETag: "5B3C1A2E053D7002CC607C5A*****"
Content-Type: text/html
Content-Length: 362149
Server: AliyunOSS
[362149 bytes of object data]
```

- 未指定versionId且Object的当前版本为删除标记

请求示例

```
GET /example HTTP/1.1
Host: versioning-get.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 03:22:33 GMT
Authorization: OSS duagpvtn35:taVlDvAJMhEumrR+oLMWtQp*****
```

返回示例

```

HTTP/1.1 404 Not Found
x-oss-request-id: 5CAC0FEADE0170*****
x-oss-delete-marker: true
x-oss-version-id: CAEQNxiBgYAOBYiIDc4ZDdmNTA2MGViZTRiNjE5NzZlZWm4OWM5OT*****
Date: Tue, 09 Apr 2019 03:22:33 GMT
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The specified key does not exist.</Message>
  <RequestId>5CAC0FEADE0170*****</RequestId>
  <HostId>versioning-get.oss-cn-hangzhou.aliyun*****</HostId>
  <Key>example</Key>
</Error>

```

- 指定versionId访问删除标记

请求示例

```

GET /example?versionId=CAEQMxiBgMCfqaWA0BYiIDliMWI4MGQ0MTVmMjQ3MmE5MDNlMmY4YmFkYTk3*
*** HTTP/1.1
Host: versioning-get.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 03:09:44 GMT
Authorization: OSS tvqm50uz4y:51UaP+wQt5k1RQang/U6Eeq*****

```

返回示例

```

HTTP/1.1 405 Method Not Allowed
x-oss-request-id: 5CAC0CF8DE0170*****
x-oss-delete-marker: true
x-oss-version-id: CAEQMxiBgMCfqaWADliMWI4MGQ0MTVmMjQ3MmE5MDNlMmY4YmFkYTk*****
Allow: DELETE
Date: Tue, 09 Apr 2019 03:09:44 GMT
Content-Type: application/xml
Content-Length: 318
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>MethodNotAllowed</Code>
  <Message>The specified method is not allowed against this resource.</Message>
  <RequestId>5CAC0CF8DE0170*****</RequestId>
  <HostId>versioning-get.oss-cn-hangzhou.aliyunc*****</HostId>
  <Method>GET</Method>
  <ResourceType>DeleteMarker</ResourceType>
</Error>

```

SDK

GetObject接口所对应的各语言SDK如下：

- [Java](#)

- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)
- [Browser.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	说明
NoSuchKey	404	目标Object不存在。
SymlinkTargetNotExist	404	Object类型为软链接，且目标Object不存在。
InvalidTargetType	400	Object类型为软链接，且目标Object类型仍为软链接。
InvalidObjectState	403	下载归档类型的Object时： • 没有提交RestoreObject请求或者上一次提交RestoreObject已经超时。 • 已经提交RestoreObject请求，但数据的RestoreObject操作还没有完成。
Not Modified	304	返回该错误的可能原因如下： • 指定了If-Modified-Since请求头，但请求的Object在指定的时间后没被修改过。 • 指定了If-None-Match请求头，且请求的Object的ETag值和您提供的ETag相等。
Precondition Failed	412	返回该错误的可能原因如下： • 指定了If-Unmodified-Since，但指定的时间早于Object实际修改时间。 • 指定了If-Match，但请求的Object的ETag值和您提供的ETag不相等。
Not Found	404	在请求中未指定Object的versionId，且Object的当前版本是删除标记（Delete Marker）时，返回该错误。
Method Not Allowed	405	在请求中指定了Object的versionId，且versionId对应的是删除标记时，返回该错误。

7.1.3. CopyObject

调用CopyObject接口拷贝同一地域下相同或不同存储空间（Bucket）之间的文件（Object）。

版本控制

`x-oss-copy-source` 默认拷贝Object的当前版本，您可以在 `x-oss-copy-source` 中加入versionId来拷贝指定的Object版本。拷贝Object时，如果源Object的对应版本为删除标记，则返回404表示该Object不存在。

如果需要恢复Object的早期版本为当前版本，您只需将Object的早期版本拷贝到同一个Bucket中，OSS会将该Object对应早期版本置为当前版本。

如果目标Bucket已开启版本控制，OSS将会为新拷贝的Object自动生成唯一的版本ID，此版本ID将会在响应Header中的 `x-oss-version-id` 返回。如果目标Bucket未开启或者暂停了版本控制，OSS将会为新拷贝的Object自动生成version ID为null的版本，且会覆盖原有versionId为null的版本。

使用限制

- 使用CopyObject接口时，Object的大小限制说明如下：
 - 如果源Bucket和目标Bucket相同，则Object的大小无限制。
 - 如果源Bucket和目标Bucket不同，则建议拷贝小于1 GB的Object。当您需要拷贝大于1 GB的Object时，请使用[UploadPart Copy](#)接口。

使用CopyObject或UploadPart Copy接口均要求对源Object有读权限。

- 在非版本控制的Bucket中，当调用CopyObject接口拷贝文件时，如果源Object与目标Object为同一个Object，则OSS只修改源Object的元数据，不拷贝源Object的内容。
- 在版本控制的Bucket中，不支持拷贝通过追加上传方式生成的Object。
- 如果源Object为软链接，则只拷贝软链接，无法拷贝软链接指向的文件内容。

计量计费

- 调用一次CopyObject接口会对源Object和目标Object所在的Bucket各增加一次Get请求次数。
- 调用CopyObject接口会对目标Object所在的Bucket增加相应的存储量。
- 调用CopyObject接口更改Object存储类型会涉及数据覆盖。例如低频访问IA创建后10天内被覆盖为标准存储Standard，则会产生20天的低频访问不足规定时长容量费用。关于存储费用的更多信息，请参见[存储费用](#)。

请求语法

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

请求头

拷贝操作涉及到的请求头均以x-oss-开头，因此所有请求头都要加到签名字符串中。

名称	类型	是否必选	取值	描述
----	----	------	----	----

名称	类型	是否必选	取值	描述
x-oss-forbid-overwrite	字符串	否	true	指定CopyObject操作时是否覆盖同名目标Object。当目标Bucket处于已开启或已暂停版本控制状态时，x-oss-forbid-overwrite请求Header设置无效，即允许覆盖同名Object。 - 未指定x-oss-forbid-overwrite或者指定x-oss-forbid-overwrite为false时，表示允许覆盖同名目标Object。 - 指定x-oss-forbid-overwrite为true时，表示禁止覆盖同名Object。 设置x-oss-forbid-overwrite请求Header会导致QPS处理性能下降，如果您有大量的操作需要使用x-oss-forbid-overwrite请求Header（QPS>1000），请联系技术支持，避免影响您的业务。 默认值：false
x-oss-copy-source	字符串	是	/oss-example/oss.jpg	指定拷贝的源地址。 默认值：无
x-oss-copy-source-if-match	字符串	否	5B3C1A2E053D763E1B002CC607C5****	如果源Object的ETag值和您提供的ETag相等，则执行拷贝操作，并返回200 OK。 默认值：无
x-oss-copy-source-if-none-match	字符串	否	5B3C1A2E053D763E1B002CC607C5****	如果源Object的ETag值和您提供的ETag不相等，则执行拷贝操作，并返回200 OK。 默认值：无
x-oss-copy-source-if-unmodified-since	字符串	否	2019-04-09T07:01:56.000Z	如果指定的时间等于或者晚于文件实际修改时间，则正常拷贝文件，并返回200 OK。 默认值：无
x-oss-copy-source-if-modified-since	字符串	否	2019-04-09T07:01:56.000Z	如果源Object在用户指定的时间以后被修改过，则执行拷贝操作。 默认值：无

名称	类型	是否必选	取值	描述
x-oss-metadata-directive	字符串	否	COPY	指定如何设置目标Object的元信息。 <i>COPY</i>（默认值）：复制源Object的元数据到目标Object。 OSS不会复制源Object的x-oss-server-side-encryption属性配置到目标Object。目标Object的服务器端加密编码方式取决于当前拷贝操作是否指定了x-oss-server-side-encryption。 <i>REPLACE</i>：忽略源Object的元数据，直接采用请求中指定的元数据。  注意 如果拷贝操作的源Object地址和目标Object地址相同，且未开启版本控制时，则无论x-oss-metadata-directive为何值，都会忽略源Object的元数据，目标Object将直接采用请求中指定的元数据。
x-oss-server-side-encryption	字符串	否	AES256	指定OSS创建目标Object时，服务器端加密编码加密算法。 取值：AES256、KMS 您只有购买了KMS套件，才能使用KMS加密算法，否则OSS会返回KmsServiceNotEnabled错误。 - 如果拷贝操作中未指定x-oss-server-side-encryption，则无论源Object是否进行过服务器端加密编码，拷贝后的目标Object均不进行服务器端加密编码。 - 如果拷贝操作中指定了x-oss-server-side-encryption，则无论源Object是否进行过服务器端加密编码，拷贝后的目标Object均会进行服务器端加密编码。并且拷贝操作的响应头中会包含x-oss-server-side-encryption，值为目标Object的加密算法。 在目标Object被下载时，响应头中也会包含x-oss-server-side-encryption，值为该Object的加密算法。
x-oss-server-side-encryption-key-id	字符串	否	9468da86-3509-4f8d-a61e-6eab1eac****	表示KMS托管的用户主密钥。 该参数仅在x-oss-server-side-encryption为KMS时有效。

名称	类型	是否必选	取值	描述
x-oss-object-acl	字符串	否	private	指定OSS创建目标Object时的访问权限。 取值： <i>default</i>（默认）：Object遵循所在存储空间的访问权限。 <i>private</i>：Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户没有权限操作该Object。 <i>public-read</i>：Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户只有该Object的读权限。请谨慎使用该权限。 <i>public-read-write</i>：Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息，请参见基于读写权限ACL的权限控制。
x-oss-storage-class	字符串	否	Standard	指定Object的存储类型。 对于任意存储类型Bucket，如果上传Object时指定该值，则此次上传的Object将存储为指定的类型。例如在IA类型的Bucket中上传Object时，如果指定x-oss-storage-class为Standard，则该Object直接存储为Standard类型。 取值： <i>Standard</i>：标准存储 <i>IA</i>：低频访问 <i>Archive</i>：归档存储 <i>ColdArchive</i>：冷归档存储 关于存储类型的更多信息，请参见存储类型介绍。
x-oss-tagging	字符串	否	a:1	指定Object的对象标签，可同时设置多个标签，例如TagA=A&TagB=B。 🔍 说明 Key和Value需要先进行URL编码，如果某项没有“=”，则看作Value为空字符串。
x-oss-tagging-directive	字符串	否	Copy	指定如何设置目标Object的对象标签。取值如下： <i>Copy</i>（默认值）：复制源Object的对象标签到目标Object。 <i>Replace</i>：忽略源Object的对象标签，直接采用请求中指定的对象标签。

此接口还需要包含Host、Date等公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅包含公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

响应元素

名称	类型	示例值	描述
CopyObjectResult	字符串	不涉及	CopyObject的结果。 默认值：无
ETag	字符串	5B3C1A2E053D76 3E1B002CC607C5* ***	目标Object的ETag值。 父元素：CopyObjectResult
LastModified	字符串	Fri, 24 Feb 2012 07:18:48 GMT	目标Object最后更新时间。 父元素：CopyObjectResult

示例

- 未开启版本控制

请求示例

```
PUT /test%2FAK.txt HTTP/1.1
Host: tesx.oss-cn-zhangjiakou.aliyuncs.com
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: text/html
Connection: keep-alive
x-oss-copy-source: /test/AK.txt
date: Fri, 28 Dec 2018 09:41:55 GMT
authorization: OSS qn6qrrqxo2oawuk53otfjbyc:gmnwPKuu20LQEjd+iPkL259A****
Content-Length: 0
```

返回示例

`x-oss-hash-crc64ecma` 表示Object的64位CRC值。该64位CRC值根据[ECMA-182](#)标准计算得出。进行CopyObject操作时，生成的Object不保证具有64位CRC值。

```

HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 28 Dec 2018 09:41:56 GMT
Content-Type: application/xml
Content-Length: 184
Connection: keep-alive
x-oss-request-id: 5C25EFE4462CE00EC6D87156
ETag: "F2064A169EE92E9775EE5324D0B1****"
x-oss-hash-crc64ecma: 12753002859196105360
x-oss-server-time: 150
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <ETag>"F2064A169EE92E9775EE5324D0B1****"</ETag>
  <LastModified>2018-12-28T09:41:56.000Z</LastModified>
</CopyObjectResult>

```

- 未指定versionId进行拷贝

请求示例

```

PUT /dest-object-example HTTP/1.1
Host: versioning-copy.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 03:45:32 GMT
Authorization: OSS qeyxjc9arppwa0t:QqwOjq7U7j04NVpPqdfcVk0l****
x-oss-copy-source: /versioning-copy-source/source-object

```

返回示例

示例中的 `x-oss-copy-source-version-id` 为源拷贝Object的versionId，在该示例中即为源拷贝Object的当前版本。`x-oss-version-id` 为新拷贝生成Object的versionId。

```

HTTP/1.1 200 OK
x-oss-copy-source-version-id: CAEQNRiBgIC28uaA0BYiIDY5OGIwNmNINjYyMTRjNTc4N2M2OGNiMjZkZTQ2****
x-oss-version-id: CAEQNxiBgIDG8uaA0BYiIGZhZDRkZTk5Zjg3YzRhNzdiMWEwZGVhNDM1NTFh****
x-oss-request-id: 5CAC155CB7AEADE01700****
Content-Type: application/xml
Content-Length: 184
Connection: keep-alive
Date: Tue, 09 Apr 2019 03:45:32 GMT
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <ETag>"C81E728D9D4C2F636F067F89CC14****"</ETag>
  <LastModified>2019-04-09T03:45:32.000Z</LastModified>
</CopyObjectResult>

```

- 指定versionId进行拷贝

请求示例

```
PUT /dest-object-example HTTP/1.1
Host: versioning-copy.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 03:45:32 GMT
Authorization: OSS qeyxjc9arppwa0t:5qG4DLahjxDpTpLlf2e8fBfX****
x-oss-copy-source: /versioning-copy-source/source-object?versionId=CAEQNRiBgICv8uaA0BYiIDliZDc3MTc1NjE5MjRkMDI4ZGU4MTZkYjY1ZDgy****
```

返回示例

示例中的 `x-oss-copy-source-version-id` 为源拷贝Object的versionId，在该示例中即为 `x-oss-copy-source` 中versionId指定的版本，`x-oss-version-id` 为新拷贝生成Object的versionId。

```
HTTP/1.1 200 OK
x-oss-copy-source-version-id: CAEQNRiBgICv8uaA0BYiIDliZDc3MTc1NjE5MjRkMDI4ZGU4MTZkYjY1ZDgy****
*
x-oss-version-id: CAEQNxiBgMDP8uaA0BYiIDlyNGNhZDQ1M2M3NzRkZThiNzE0N2I3ZDkxOWY4****
x-oss-request-id: 5CAC155CB7AEADE01700****
Content-Type: application/xml
Content-Length: 184
Connection: keep-alive
Date: Tue, 09 Apr 2019 03:45:32 GMT
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult>
  <ETag>"C4CA4238A0B923820DCC509A6F75****"</ETag>
  <LastModified>2019-04-09T03:45:32.000Z</LastModified>
</CopyObjectResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)
- [Ruby](#)

错误码

错误码	HTTP状态码	描述
InvalidArgument	400	x-oss-storage-class等参数的值无效。

错误码	HTTP状态码	描述
Precondition Failed	412	返回该错误的可能原因如下： - 指定了x-oss-copy-source-if-match请求头，但源Object的ETag值和您提供的ETag不相等。 - 指定了x-oss-copy-source-if-unmodified-since请求头，但指定的时间早于文件实际修改时间。
Not Modified	304	返回该错误的可能原因如下： - 指定了x-oss-copy-source-if-none-match请求头，但源Object的ETag值和您提供的ETag相等。 - 指定了x-oss-copy-source-if-modified-since请求头，但源Object在指定的时间后没被修改过。
KmsServiceNotEnabled	403	将x-oss-server-side-encryption指定为KMS，但没有预先购买KMS套件。
FileAlreadyExists	409	当请求的Header中携带x-oss-forbid-overwrite=true时，表示禁止覆盖同名文件。如果文件已存在，则返回该错误。
FileImmutable	409	Bucket内的数据处于被保护状态时，如果尝试删除或修改这些数据，则返回该错误。

7.1.4. AppendObject

调用AppendObject接口用于以追加写的方式上传文件（Object）。通过AppendObject操作创建的Object类型为Appendable Object，而通过PutObject上传的Object是Normal Object。

版本控制

在目标Bucket处于开启或暂停版本控制状态下，对Appendable类型Object执行相关操作时，有如下注意事项：

- 仅允许对当前版本为Appendable类型的Object执行追加上传（AppendObject）操作，且OSS不会为该Appendable类型的Object生成历史版本。

对当前版本为Appendable类型的Object执行PutObject或删除Object操作时，OSS会将该Appendable类型的Object保留为历史版本，但该Object不允许继续追加。

- 不支持对Appendable类型Object执行拷贝操作。
- 不支持对非Appendable类型的Object，包括Normal Object、删除标记（Delete Marker）等执行AppendObject操作。

使用限制

- 通过AppendObject方式最后生成的Object大小不得超过5 GB。
- 处于[合规保留策略](#)保护期的Object不支持AppendObject操作。
- AppendableObject不支持指定CMK ID进行服务端KMS加密。

请求语法

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

请求头

 注意 append和position均为 **CanonicalizedResource**，需要包含在签名中。

名称	类型	是否必选	示例值	描述
append	字符串	是	不涉及	用于指定AppendObject操作。每次执行AppendObject都会更新该Object的最后修改时间。
position	字符串	是	0	用于指定从何处进行追加。每次操作成功后，响应消息头x-oss-next-append-position会标明下一次追加的position。 首次追加操作的position必须为0，后续追加操作的position是Object的当前大小。例如，第一次AppendObject请求指定position值为0，content-length是65536，则第二次AppendObject需要指定position为65536。 - 当position值为0，且不存在同名Object时，则AppendObject与PutObject请求类似，即允许设置x-oss-server-side-encryption等请求头。如果加入了正确的x-oss-server-side-encryption头，那么后续的AppendObject响应头部也会包含x-oss-server-side-encryption头。后续如需更改元数据，可以使用CopyObject接口。 - 在position值正确的情况下，对已存在的Appendable Object追加一个大小为0的内容，该操作不会改变Object的状态。
Cache-control	字符串	否	no-cache	指定该Object的网页缓存行为。更多信息，请参见RFC2616。 默认值：无
Content-Disposition	字符串	否	attachment;filename=oss_downloaded.jpg	指定该Object被下载时的名称。更多信息，请参见RFC2616。 默认值：无
Content-Encoding	字符串	否	utf-8	指定该Object的内容编码格式。更多信息，请参见RFC2616。 默认值：无

名称	类型	是否必选	示例值	描述
Content-MD5	字符串	否	ohhmqLBJ FiKkPSBO1 eNaUA==	Content-MD5是一串由MD5算法生成的值，该请求头用于检查消息内容是否与发送时一致。 获取Content-MD5值：对消息内容（不包括头部）执行MD5算法，获得128比特位数字，然后对该数字进行base64编码。 默认值：无 限制：无
Expires	GMT	否	Wed, 08 Jul 2015 16:57:01 GMT	过期时间。更多信息，请参见 RFC2616 。 默认值：无
x-oss-server-side-encryption	字符串	否	AES256	指定服务器端加密方式。 合法值： <i>AES256</i>：使用OSS完全托管密钥进行加解密（SSE-OSS）。 <i>KMS</i>：使用KMS托管密钥进行加解密。 <i>SM4</i>：国密SM4算法。
x-oss-object-acl	字符串	否	private	指定Object的访问权限。 取值： <i>default</i>（默认）：Object遵循所在存储空间的访问权限。 <i>private</i>：Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户没有权限操作该Object。 <i>public-read</i>：Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户只有该Object的读权限。请谨慎使用该权限。 <i>public-read-write</i>：Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息，请参见 基于读写权限ACL的权限控制 。

名称	类型	是否必选	示例值	描述
x-oss-storage-class	字符串	否	Standard	指定Object的存储类型。 对于任意存储类型的Bucket，如果上传Object时指定此参数，则此次上传的Object将存储为指定的类型。例如在IA类型的Bucket中上传Object时，如果指定x-oss-storage-class为Standard，则该Object直接存储为Standard。 取值： <i>Standard</i>：标准存储 <i>IA</i>：低频访问 <i>Archive</i>：归档存储 关于存储类型的更多信息，请参见存储类型介绍。  注意 该值仅在首次执行AppendObject操作时有效，后续追加时不生效。
x-oss-meta-*	字符串	否	x-oss-meta-location	创建AppendObject时可以添加x-oss-meta-*, 继续追加时不可以携带此参数。如果配置以x-oss-meta-*为前缀的参数，则该参数视为元数据。 元数据大小限制：一个Object可以包含多个元数据，但所有的元数据总大小不能超过8 KB。 元数据命名规则：支持短划线（-）、数字、英文字母（a~z）。英文字符的大写字母会被转成小写字母，不支持下划线（_）在内的其他字符。

关于此接口涉及的其他公共请求头的更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

响应消息头	类型	示例	描述
x-oss-next-append-position	64位整型	1717	下一次请求应当提供的position，即当前Object大小。 当AppendObject执行成功，或者因position和Object大小不匹配而引起的409错误时，会返回此消息头。
x-oss-hash-crc64ecma	64位整型	3231342946509354535	表明Object的64位CRC值。该64位CRC由ECMA-182标准计算得出。

关于此接口涉及的其他公共响应头的更多信息，请参见[公共响应头（Common Response Headers）](#)。

CRC64的计算方式

Appendable Object的CRC64采用[ECMA-182](#)标准。您可以使用以下方法进行计算：

- 用Boost CRC模块的方式计算


```
typedef boost::crc_optimal<64, 0x42F0E1EBA9EA3693ULL, 0xffffffffffffffffULL, 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();
}
```

- 用Python crcmod的方式计算

```
do_crc64 = crcmod.mkCrcFun(0x142F0E1EBA9EA3693L, initCrc=0L, xorOut=0xfffffffffffffL, rev=True)
print do_crc64( "123456789" )
```

和其他操作的关系

其他操作	关系描述
PutObject	如果对一个已经存在的Appendable Object进行PutObject操作，该Appendable Object会被新的Object覆盖，类型转换为Normal Object。
HeadObject	对Appendable Object执行HeadObject操作会返回x-oss-next-append-position、x-oss-hash-crc64ecma以及x-oss-object-type。Appendable Object的x-oss-object-type为Appendable。
GetBucket (ListObjects)	在GetBucket请求的响应中，会将Appendable Object的Type设为Appendable。
CopyObject	• 允许使用CopyObject修改Appendable Object自定义的元信息。 • 不允许使用CopyObject拷贝一个Appendable Object。 • 不允许使用CopyObject修改Appendable Object的服务器端加密属性。

示例

- 请求示例

```
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
x-oss-storage-class: Archive
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 qn6qrrqxo2oawuk53otf****:kZoYNv66bsmc10+dcGKw5x2P****
[1717 bytes of object data]
```

返回示例

```
HTTP/1.1 200 OK
Date: Wed, 08 Jul 2015 06:57:01 GMT
ETag: "0F7230CAA4BE94CCBDC99C550000*****"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-hash-crc64ecma: 14741617095266562575
x-oss-next-append-position: 1717
x-oss-request-id: 559CC9BDC755F95A6448****
```

- 版本控制请求示例

对于开启或暂停了版本控制的Bucket，AppendObject的响应Header中会返回x-oss-version-id，且x-oss-version-id只能为null。

```
POST /example?append&position=0 HTTP/1.1
Host: versioning-append.oss.aliyuncs.com
Date: Tue, 09 Apr 2019 03:59:33 GMT
Content-Length: 3
Content-Type: application/octet-stream
Authorization: OSS bwo4j5l8d3j****:MCY5nnfgfJU/f3Xe0odqBtG5****
```

返回示例

```
HTTP/1.1 200 OK
Date: Tue, 09 Apr 2019 03:59:33 GMT
ETag: "2776271A4A09D82CA518AC5C0000*****"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-version-id: null
x-oss-hash-crc64ecma: 3231342946509354535
x-oss-next-append-position: 3
x-oss-request-id: 5CAC18A5B7AEADE01700****
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Ruby](#)

错误码

错误代码	HTTP 状态码	说明
ObjectNotAppendable	409	对一个非Appendable Object进行AppendObject操作。
PositionNotEqualToLength	409	- position的值和当前Object的长度不一致。 您可以通过响应头x-oss-next-append-position得到下一次position，并再次进行请求。由于并发的关系，即使把position的值设置为x-oss-next-append-position，该请求依然可能因为PositionNotEqualToLength而失败。 - position值为0时，如果没有同名Appendable Object，或者同名Appendable Object长度为0，该请求成功；其他情况均视为position和Object长度不匹配的情形，返回此错误码。
InvalidArgument	400	x-oss-storage-class、x-oss-object-acl等值设置无效。
FileImmutable	409	Bucket内的数据处于被保护状态时，如果您尝试删除或修改这些数据，将返回此错误码。
KmsServiceNotEnabled	403	使用KMS加密算法时，没有在控制台预先开通密钥管理服务KMS。

7.1.5. DeleteObject

调用DeleteObject删除某个文件（Object）。

注意事项

- 要删除文件，您必须有Object的写权限。
- 无论要删除的Object是否存在，删除成功后均会返回204状态码。
- 如果Object类型为软链接，使用DeleteObject接口只会删除该软链接。

版本控制

版本控制状态下的删除行为说明如下：

- 未指定versionId（临时删除）：

如果在未指定versionId的情况下执行删除操作时，默认不会删除Object的当前版本，而是对当前版本插入删除标记（Delete Marker）。此时，在未指定versionId的情况下执行GetObject操作，OSS会检测到当前版本为删除标记，并返回 404 Not Found。此外，响应中还会返回 header: x-oss-delete-marker = true 以及新生成的删除标记的版本号 x-oss-version-id。

x-oss-delete-marker 的值为true，表示与返回的 x-oss-version-id 对应的版本为删除标记。

- 指定versionId（永久删除）：

如果在指定versionId的情况下执行删除操作时，OSS会根据 params 中指定的 versionId 参数永久删除该版本。如果要删除ID为“null”的版本，请在 params 参数中添加 params['versionId'] = “null”，OSS 将“null”字符串当成“null”的versionId，从而删除versionId为“null”的Object。

请求语法

```
DELETE /ObjectName HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 02 Jan 2019 13:28:38 GMT
Authorization: SignatureValue
```

请求头

此接口仅包含公共请求头。关于公共请求头的更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

名称	类型	示例值	描述
x-oss-delete-marker	布尔型	true	Object为删除标记。 - 未指定versionId执行DeleteObject操作时，OSS会创建删除标记，响应中会返回此Header，且值为true。 - 指定versionId来永久删除指定Object版本时，如果该版本是删除标记，响应中会返回此Header，且值为true。 有效值： <i>true</i>
x-oss-version-id	字符串	CAEQMxiBgIDh3ZCB0BYilGE4YjlyMjExZDhhYjQxNzZiNGUyZTI4ZjljZDcz****	Object的versionId。 - 未指定versionId执行DeleteObject操作时，OSS会创建删除标记，响应中会返回此Header，表示新创建的删除标记的versionId。 - 指定versionId来永久删除Object指定版本时，响应中会返回此Header，表示删除Object的versionId。

此接口还需要包含公共响应头。关于公共响应头的更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 执行DeleteObject操作

请求示例

```
DELETE /AK.txt HTTP/1.1
Host: test.oss-cn-zhangjiakou.aliyuncs.com
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Windows/7/AMD64;3.7.0)
Accept: text/html
Connection: keep-alive
date: Wed, 02 Jan 2019 13:28:38 GMT
authorization: OSS qn6qrrqxo2oawuk53otfjbyc:zUglwRPGkbByZxm1+y4eyu+NIUs=zV0****
Content-Length: 0
```

返回示例

```
HTTP/1.1 204 No Content
Server: AliyunOSS
Date: Wed, 02 Jan 2019 13:28:38 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5C2CBC8653718B5511EF4535
x-oss-server-time: 134
```

- 未指定versionId执行DeleteObject操作

此时OSS中会插入删除标记，响应中将返回 `x-oss-delete-marker=true`。

请求示例

```
DELETE /example HTTP/1.1
Host: versioning-delete.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 04:08:23 GMT
Authorization: OSS twnetzwjkqr9eq6:z73SSKA6t2tNTP4GuPjPiyV/****
```

返回示例

```
HTTP/1.1 204 NoContent
x-oss-delete-marker: true
x-oss-version-id: CAEQMxiBgIDh3ZCB0BYilGE4YjlyMjExZDhhYjQxNzZiNGUyZTI4ZjljZDcz****
x-oss-request-id: 5CAC1AB7B7AEADE01700****
Date: Tue, 09 Apr 2019 04:08:23 GMT
Connection: keep-alive
Server: AliyunOSS
```

- 指定versionId执行DeleteObject操作

通过指定versionId来执行DeleteObject操作时，将永久删除该指定versionId的Object。

请求示例

```
DELETE /example?versionId=CAEQOBiBgIDNlJeB0BYiIDAwYjJINDQ4YjJkMzQxMmY5NTM5N2UzZWNiZTQ2*
*** HTTP/1.1
Host: versioning-delete.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 04:11:54 GMT
Authorization: OSS gb3m2qiwirupd6v:UjOXBmlbJD3qXL+DP1EDNyCI****
```

返回示例

```
HTTP/1.1 204 No Content
x-oss-version-id: CAEQOBiBgIDNlJeB0BYiIDAwYjJINDQ4YjJkMzQxMmY5NTM5N2UzZWNiZTQ2****
x-oss-request-id: 5CAC1B8AB7AEADE01700****
Date: Tue, 09 Apr 2019 04:11:54 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

- 指定versionId删除“删除标记”

指定删除的版本为删除标记时，响应中将返回 `x-oss-delete-marker=true`。

请求示例

```
DELETE /example?versionId=CAEQOBiBgIDNlJeB0BYiIDAwYjJINDQ4YjJkMzQxMmY5NTM5N2UzZWNiZTQ2*  
*** HTTP/1.1  
Host: versioning-delete.oss-cn-hangzhou.aliyuncs.com  
Date: Tue, 09 Apr 2019 04:16:25 GMT  
Authorization: OSS jh475i54ffozhoy:4tX6Z+fnhtINhp0g+sRiLEQb****
```

返回示例

```
HTTP/1.1 204 No Content  
x-oss-delete-marker: true  
x-oss-version-id: CAEQNhiBgIDFtp.B0BYiIDk4NzgwMmU4NDMyOTQyM2NiMDQxOTcxYWNhMjc1****  
x-oss-request-id: 5CAC1C99B7AEADE01700****  
Date: Tue, 09 Apr 2019 04:16:25 GMT  
Content-Length: 0  
Connection: keep-alive  
Server: AliyunOSS
```

SDK

DeleteObject接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)
- [Browser.js](#)

错误码

错误码	HTTP状态码	描述
FileImmutable	409	Bucket中的数据处于被保护状态时，如果尝试删除或修改这些数据，则返回该错误。

7.1.6. DeleteMultipleObjects

DeleteMultipleObjects接口用于删除同一个存储空间（Bucket）中的多个文件（Object）。

 **说明** 单次调用DeleteMultipleObjects接口最多允许删除1000个文件。

版本控制

在开启版本控制的Bucket中，同样可以使用DeleteMultipleObjects接口来批量删除Object。在Delete请求中如果没有指定versionId，将插入删除标记。如果指定了versionId，将永久删除该Object的指定版本。

请求语法

```
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>
```

请求头

OSS会根据以下请求头验证收到的消息体，消息体正确才会执行删除操作。

名称	类型	是否必选	描述
Encoding-type	字符串	否	Key使用UTF-8字符。如果Key中包含XML 1.0标准不支持的控制字符，您可以通过指定Encoding-type对返回结果中的Key进行编码。 默认值：无 可选值： <i>url</i>
Content-Length	字符串	是	用于描述HTTP消息体的传输长度。 OSS会根据此请求头验证收到的消息体，消息体正确才会执行删除操作。
Content-MD5	字符串	是	Content-MD5是一串由MD5算法生成的值，该请求头用于检查消息内容是否与发送时一致。上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性。 ❓ 说明 将DeleteMultipleObjects的请求消息体经过MD5加密后得到一个128位字节数组。然后将该字节数组用base64算法编码，编码后得到的字符串即Content-MD5字段内容。

请求元素

名称	类型	是否必选	描述
----	----	------	----

名称	类型	是否必选	描述
Delete	容器	是	保存DeleteMultipleObjects请求的容器。 子节点：一个或多个Object元素，Quiet元素 父节点：None
Object	容器	是	保存一个Object信息的容器。 子节点：Key 父节点：Delete
Key	字符串	是	被删除Object的名字。 父节点：Object
Quiet	枚举字符串	是	打开简单响应模式的开关。 DeleteMultipleObjects提供以下两种消息返回模式： - 简单模式（quiet）：OSS返回的消息体中只包含删除过程中出错的Object结果。如果所有删除都成功，则没有消息体。 - 详细模式（verbose）：OSS返回的消息体中会包含所有删除Object的结果。默认采用详细模式。 有效值： <i>true</i> （开启简单模式）、 <i>false</i> （开启详细模式） 默认值： <i>false</i> 父节点：Delete
VersionId	字符串	否	被删除Object的版本号。 父节点：Object

响应元素

名称	类型	描述
Deleted	容器	保存被成功删除的Object的容器。 子节点：Key 父节点：DeleteResult
DeleteResult	容器	保存DeleteMultipleObjects请求结果的容器。 子节点：Deleted 父节点：None

名称	类型	描述
Key	字符串	被删除Object的名字。 父节点：Deleted
EncodingType	字符串	返回结果中编码使用的类型。如果请求的参数中指定了 Encoding-type，则返回的结果会对Key进行编码。 父节点：容器
DeleteMarker	布尔	表示该版本是否为删除标记。true为是，false为否。 ❓ 说明 只有当创建删除标记和永久删除删除标记时，才会返回该元素，且值为true。
DeleteMarkerVersionId	字符串	表示删除标记对应的版本ID。 父节点：Deleted

示例

- 关闭简单响应模式的请求示例

```
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/jEfb****
<?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>
```

返回示例

```

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>

```

- 打开简单响应模式的请求示例

```

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:WuV0Jks8RyGSNQrBca64kEEx****
<?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>

```

返回示例

```

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

```

- 未指定versionID执行DeleteMultipartObjects操作的请求示例

```

POST /?delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 04:20:03 GMT
Content-MD5: xSLOYWaPC86RPwWXNiFeXg==
Authorization: OSS xq39jyxyrddzvvh:/G5kxMlw1iIMQjPp2HkJEp5q****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
  </Object>
  <Object>
    <Key>test.jpg</Key>
  </Object>
</Delete>

```

返回示例

以下示例中由于要删除的两个Object（multipart.dat和test.jpg）都未指定versionId，因此OSS为其插入了删除标记，并返回<DeleteMarker>true</DeleteMarker>和<DeleteMarkerVersionId>XXXXXX</DeleteMarkerVersionId>。

```

HTTP/1.1 200 OK
x-oss-request-id: 5CAC1D73B7AEADE01700****
Date: Tue, 09 Apr 2019 04:20:03 GMT
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult>
  <Deleted>
    <Key>multipart.data</Key>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>CAEQMhiBgIDXiaaB0BYiIGQzYmRkZGUxMTM1ZDRjOTZhNjk4YjRjMTAyZjht****
  </DeleteMarkerVersionId>
  </Deleted>
  <Deleted>
    <Key>test.jpg</Key>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>CAEQMhiBgIDB3aWB0BYiIGUzYTA3YzliMzVmNzRkZGM5NjllYTVlMjYyYWEy****
  </DeleteMarkerVersionId>
  </Deleted>
</DeleteResult>

```

- 指定versionId来删除指定版本Object的请求示例

 **说明** 指定versionId的同时必须指定Key。

```
POST /?delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:09:34 GMT
Content-Length:151
Content-MD5: 2TpK+dL/tGyuSA+YCEuSVg==
Authorization: OSS qac50zy2vzbvq4z:/pingCxyqfxc0+50Bfi2SX9c****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
    <VersionId>CAEQNRiBgIDyz.6C0BYiIGQ2NWEwNmVhNTA3ZTQ3MzM5ODliYjM1ZTdjYjA4****</VersionId>
  </Object>
</Delete>
```

返回示例

示例中返回了被删除的对象Key和相应的版本ID。

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC371EB7AEADE01700****
Date: Tue, 09 Apr 2019 06:09:34 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>
    <VersionId>CAEQNRiBgIDyz.6C0BYiIGQ2NWEwNmVhNTA3ZTQ3MzM5ODliYjM1ZTdjYjA4****</VersionId>
  </Deleted>
</DeleteResult>
```

- 指定versionId来删除删除标记的请求示例

```
POST /?delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:14:50 GMT
Content-Length: 178
Content-MD5: dX9IFePFgYhmINvAhG30Bg==
Authorization: OSS 9dfwrokza5gf1p7:iyl6IU6TcfFXcu5p0ds5dUdo****
<?xml version="1.0" encoding="UTF-8"?>
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>multipart.data</Key>
    <VersionId>CAEQNRiBgICEoPIC0BYiIGMxZWJmYmMzYjE0OTQ0ZmZhYjgzNzkzYjc2NjZk****</VersionId>
  </Object>
</Delete>
```

返回示例

- 示例中返回的Key和VersionId分别代表被删除的对象Key及其对应的版本ID。

- DeleteMarker和DeleteMarkerVersionId分别代表被删除的是删除标记及其对应的版本ID，此时的VersionId和DeleteMarkerVersionId相同，且DeleteMarker和DeleteMarkerVersionId同时出现。

```
HTTP/1.1 200 OK
x-oss-request-id: 5CAC385AB7AEADE01700****
Date: Tue, 09 Apr 2019 06:14:50 GMT
Content-Length: 364
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>demo.jpg</Key>
    <VersionId>CAEQNRiBgICEoPiC0BYilGMxZWJmYmMzYjE0OTQ0ZmZhYjgzNzkzYjc2NjZk****</VersionId>
    <DeleteMarker>true</DeleteMarker>
    <DeleteMarkerVersionId>111111</DeleteMarkerVersionId>
  </Deleted>
</DeleteResult>
```

SDK

DeleteMultipleObjects接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [iOS](#)
- [Node.js](#)
- [Browser.js](#)
- [Ruby](#)

错误码

错误码	HTTP 状态码	描述
InvalidDigest	400	上传了Content-MD5请求头后，OSS会计算消息体的Content-MD5并检查一致性，如果不一致会返回此错误码。
MalformedXML	400	• 消息体最大允许2 MB的内容，超过2 MB则返回此错误码。 • 单次调用DeleteMultipleObjects接口最多支持删除1000个Object。单次请求超过1000个Object返回此错误码。
FileImmutable	409	Bucket内的数据处于被保护状态时，若您尝试删除或修改这些数据，将返回此错误码。

7.1.7. HeadObject

HeadObject接口用于获取某个文件（Object）的元信息。使用此接口不会返回文件内容。

版本控制

HeadObject操作默认获取Object当前版本的元信息。如果Object的当前版本为删除标记，则返回404 Not Found。请求参数中指定versionId则返回指定版本Object的元信息。

请求语法

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

请求头

名称	类型	是否必选	描述
If-Modified-Since	字符串	否	如果传入参数中的时间早于实际修改时间，则返回200 OK和Object Meta；否则返回304 not modified。 默认值：无
If-Unmodified-Since	字符串	否	如果传入参数中的时间等于或者晚于文件实际修改时间，则返回200 OK和Object Meta；否则返回412 precondition failed。 默认值：无
If-Match	字符串	否	如果传入期望的ETag和Object的ETag匹配，则返回200 OK和Object Meta；否则返回412 precondition failed。 默认值：无
If-None-Match	字符串	否	如果传入期望的ETag值和Object的ETag不匹配，则返回200 OK和Object Meta；否则返回304 Not Modified。 默认值：无

响应头

如果请求的文件类型为软链接，则各个响应头的信息说明如下：

- Content-Length、ETag以及Content-Md5均为目标文件的元信息。
- Last-Modified取软链接的最后修改时间及目标文件的最后修改时间两者中的较大值。
- 其他响应头均表示软链接的元信息。

名称	类型	描述
----	----	----

名称	类型	描述
x-oss-meta-*	字符串	以x-oss-meta-为前缀的参数作为用户自定义meta header。当用户在PutObject时设置了以x-oss-meta-为前缀的自定义meta，则响应中会包含这些自定义meta。
非x-oss-meta-开头的自定义header	字符串	当用户在PutObject时，自定义一些非x-oss-meta为前缀的Header，如x-oss-persistent-headers: <i>key1:base64_encode(value1),key2:base64_encode(value2)....</i> ，响应中会增加相应的自定义Header。 具体操作，请参见 OSS如何添加非x-oss-meta-开头的自定义 。
x-oss-server-side-encryption	字符串	若该Object为进行服务器端熵编码加密存储的，则在响应头中会返回此参数，其值表明该Object的服务器端加密算法。
x-oss-server-side-encryption-key-id	字符串	如果用户在创建Object时使用了服务端加密，且加密方法为KMS，则响应中会包含此Header，表示加密所使用的用户KMS key ID。
x-oss-storage-class	字符串	表示Object的存储类型，分别为：标准存储类型（Standard）、低频访问存储类型（IA）、归档存储类型（Archive）和冷归档存储类型（ColdArchive）。 - 标准存储类型提供高可靠、高可用、高性能的对象存储服务，能够支持频繁的数据访问。 - 低频访问存储类型适合需要长期存储但不经常被访问的数据（平均每月访问频率1到2次）。 - 归档存储类型适合需要长期存储（建议半年以上）的归档数据，在存储周期内极少被访问，数据进入到可读取状态需要1分钟的解冻时间。 - 冷归档存储类型适用于需要长期保存且几乎不访问的数据。
x-oss-object-type	字符串	表示Object的类型。 - 通过PutObject上传或者通过CreateDirectory创建的Object类型为Normal。 - 通过AppendObject上传的Object类型为Appendable。 - 通过MultipartUpload上传的Object类型为Multipart。
x-oss-next-append-position	字符串	对于Appendable类型的Object会返回此Header，指明下一次请求应当提供的position。
x-oss-hash-crc64ecma	字符串	表示该Object的64位CRC值。该64位CRC根据ECMA-182标准计算得出。 对OSS支持CRC64校验前创建的Object，调用HeadObject接口时可能不会返回此响应头。
x-oss-expiration	字符串	如果为该Object设置了生命周期规则（Lifecycle），响应中将包含x-oss-expiration header。其中expiry-date的值表示该Object的过期日期，rule-id的值表示相匹配的规则ID。

名称	类型	描述
x-oss-restore	字符串	如果Bucket类型为Archive，且用户已经提交Restore请求，则响应头中会以x-oss-restore返回该Object的Restore状态，分如下几种情况： - 如果没有提交Restore或者Restore已经超时，则不返回该字段。 - 如果已经提交Restore，且Restore没有完成，则返回的x-oss-restore值为ongoing-request="true"。 - 如果已经提交Restore，且Restore已经完成，则返回的x-oss-restore值为ongoing-request="false"，expiry-date="Sun, 16 Apr 2017 08:12:33 GMT"，其中expiry-date是Restore完成后Object进入可读状态的过期时间。
x-oss-process-status	字符串	当用户通过MNS消息服务创建OSS事件通知后，在进行请求OSS相关操作时如果有匹配的事件通知规则，则响应中会携带这个Header，值为经过Base64编码JSON格式的事件通知结果。
x-oss-request-charged	字符串	当Object所属的Bucket被设置为请求者付费模式，且请求者不是Bucket的拥有者时，响应中将携带此Header，值为requester。
Content-Md5	字符串	- 对于Normal类型的Object，根据RFC 1864标准对消息内容（不包括Header）计算Md5值获得128比特位数字，对该数字进行Base64编码作为一个消息的Content-Md5值。 - Multipart和Appendable类型的文件不会返回这个Header。
Last-Modified	字符串	Object最后一次修改的日期，格式为HTTP 1.1协议中规定的GMT时间。
Access-Control-Allow-Origin	字符串	当Object所在的Bucket配置了CORS规则，且请求的Origin满足指定的CORS规则时会在响应中包含这个Origin。
Access-Control-Allow-Methods	字符串	当Object所在的Bucket配置了CORS规则，且请求的Access-Control-Request-Method满足指定的CORS规则时会在响应中包含允许的Methods。
Access-Control-Max-Age	字符串	当Object所在的Bucket配置了CORS规则，且请求满足Bucket配置的CORS规则时会在响应中包含MaxAgeSeconds。
Access-Control-Allow-Headers	字符串	当Object所在的Bucket配置了CORS规则，且请求满足指定的CORS规则时会在响应中包含这些Headers。
Access-Control-Expose-Headers	字符串	表示允许访问客户端JavaScript程序的headers列表。当Object所在的Bucket配置了CORS规则，且请求满足指定的CORS规则时会在响应中包含ExposeHeader。
x-oss-tagging-count	字符串	对象关联的标签个数。仅当用户有读取标签权限时返回。

示例

- 未启用版本控制

请求示例

```
HEAD /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanIJ5dLA092wpD****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A6448****
x-oss-object-type: Normal
x-oss-storage-class: Archive
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpeg
Connection: keep-alive
Server: AliyunOSS
```

- 请求对象指定版本（启用版本控制）

请求示例

```
HEAD /example?versionId=CAEQNRiBgICb8o6D0BYiIDNINzk5NGE2M2Y3ZjRhZTViYTAxZGE0ZTEyMWYy****
Host: versioning-test.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 06:27:12 GMT
Authorization: OSS ryghu9rp3mqp1ya:RvyjGvKxaUhdF0ibyEwX5mOM****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-versionId: CAEQNRiBgICb8o6D0BYiIDNINzk5NGE2M2Y3ZjRhZTViYTAxZGE0ZTEyMWYy****
x-oss-request-id: 5CAC3B40B7AEADE01700****
x-oss-object-type: Normal
x-oss-storage-class: Archive
Date: Fri, 7 Aug 2020 06:27:12 GMT
Last-Modified: Fri, 7 Aug 2020 06:27:12 GMT
ETag: "A082B659EF78733A5A042FA253B1****"
Content-Length: 481827
Content-Type: text/html
Connection: keep-alive
Server: AliyunOSS
```

- 请求对象最新版本（启用版本控制）

请求示例

```
HEAD /example HTTP/1.1
Host: versioning-test.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 06:27:12 GMT
Authorization: OSS ryghu9rp3mqp1ya:RvyjGvKxaUhdF0ibyEwX5mOM****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-versionId: CAEQMxiBgMCZov2D0BYiIDY4MDIlOTc2YmY5MjQxMzdiOGI3OTlhNTU0ODIx****
x-oss-request-id: 5CAC3B40B7AEADE01700****
x-oss-object-type: Normal
x-oss-storage-class: Archive
Date: Fri, 7 Aug 2020 06:27:12 GMT
Last-Modified: Fri, 7 Aug 2020 06:27:12 GMT
ETag: "3663F7B0B9D3153F884C821E7CF4****"
Content-Length: 485859
Content-Type: text/html
Connection: keep-alive
Server: AliyunOSS
```

- 提交Restore请求但Restore未完成

请求示例

```
HEAD /oss.jpg HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS e1Unnbm1rgdnpl:KKxkdNrUBu2t1kqIDh0MLbDb****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 58F71A164529F18D7F00****
x-oss-object-type: Normal
x-oss-storage-class: Archive
x-oss-restore: ongoing-request="true"
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 7 Aug 2020 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpeg
Connection: keep-alive
Server: AliyunOSS
```

- 提交Restore请求且Restore已完成

请求示例

```
HEAD /oss.jpg HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 09:35:51 GMT
Authorization: OSS e1Unnbm1rgdnpl:21qtGJ+ykDVmdu6O6FMJnn+W****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 58F725344529F18D7F00****
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: Fri, 7 Aug 2020 09:35:51 GMT
Last-Modified: Fri, 7 Aug 2020 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
```

- 使用服务端加密SSE-OSS

请求示例

```
HEAD /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanIJ5dLA092wpD****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A6448****
x-oss-object-type: Normal
x-oss-storage-class: Archive
x-oss-server-side-encryption: AES256
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 7 Aug 2020 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpeg
Connection: keep-alive
Server: AliyunOSS
```

- 使用服务器端加密SSE-KMS

请求示例

```
HEAD /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 7 Aug 2020 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanIJ5dLA092wpD****
```

返回示例

```

HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
x-oss-object-type: Normal
x-oss-storage-class: Archive
x-oss-server-side-encryption: KMS
x-oss-server-side-encryption-key-id: 9468da86-3509-4f8d-a61e-6eab1eac****
Date: Fri, 7 Aug 2020 07:32:52 GMT
Last-Modified: Fri, 7 Aug 2020 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a3986****"
Content-Length: 344606
Content-Type: image/jpeg
Connection: keep-alive
Server: AliyunOSS

```

错误码

错误码	HTTP 状态码	描述
Not Found	404	请求的文件不存在。
SymlinkTargetNotExist	404	请求的文件类型为软链接。
InvalidTargetType	400	请求的文件类型为软链接，且对应的目标文件类型也为软链接。
Not Modified	304	返回该错误的可能原因如下： - 指定了If-Modified-Since请求头，但源Object在指定的时间后没被修改过。 - 指定了If-None-Match请求头，且源Object的ETag值和您提供的ETag相等。
PreconditionFailed	412	返回该错误的可能原因如下： - 指定了If-Unmodified-Since，但指定的时间早于Object实际修改时间。 - 指定了If-Match，但源Object的ETag值和您提供的ETag不相等。

7.1.8. GetObjectMeta

调用GetObjectMeta接口获取一个文件（Object）的元数据信息，包括该Object的ETag、Size、LastModified信息，并且不返回该Object的内容。

注意事项

 **说明** 如果Object类型为软链接，则会返回软链接自身信息。

当Bucket未启用版本控制时，要获取文件的元数据信息，您必须有 `oss:GetObject` 权限。当Bucket已启用版本控制时，要获取文件指定版本（请求中携带了x-oss-version-id请求头）的元数据信息，您必须有 `oss:GetObjectVersion` 权限。具体操作，请参见[为RAM用户授权自定义的RAM Policy](#)。

版本控制

GetObjectMeta操作默认获取Object当前版本的元数据信息。如果Object的当前版本为删除标记，则返回404 Not Found。请求参数中指定versionId则返回指定版本Object的元数据信息。

请求语法

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

请求头

此接口仅涉及公共请求头。更多信息，请参见公共请求头（Common Request Headers）。

响应头

响应头	类型	示例值	描述
Content-Length	字符串	344606	Object的文件大小，单位为字节。
ETag	字符串	5B3C1A2E053D763E1B002CC607C5**	Object生成时会创建ETag (entity tag)，ETag用于标识一个Object的内容。 对于通过PutObject请求创建的Object，ETag值是其内容的MD5值；对于其他方式创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。ETag值可以用于检查Object内容是否发生变化。不建议用户使用ETag作为Object内容的MD5校验来验证数据完整性。 默认值：无
x-oss-version-id	字符串	CAEQNRiBgIDMh4mD0BYiIDUzNDA4OGNmZjBjYTQ0YmI4Y2I4ZmVlYzJlNGVk****	Object的版本ID。只有查看Object指定版本的元数据信息时才显示该字段。

此接口还需要包含公共响应头。更多信息，请参见公共响应头（Common Response Headers）。

示例

- 未启用版本控制

请求示例

```
HEAD /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 qn6qrrqxo2oawuk53otf****:CTkuxpLai4XZ+Wwlfnm0Fmgb****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A6448****
Date: Wed, 29 Apr 2015 05:21:12 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5****"
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 344606
Connection: keep-alive
Server: AliyunOSS
```

- 已启用版本控制

请求示例

```
GET /example?objectMeta&versionId=CAEQNRiBgIDMh4mD0BYiIDUzNDA4OGNmZjBjYTQ0YmI4Y2I4ZmVlYzJlNGVk**** HTTP/1.1
Host: versioning-test.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:24:00 GMT
Authorization: OSS 5n4nrhyqrCs****:i/M/c36KzrOEA/bBSHLIIAt****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-version-id: CAEQNRiBgIDMh4mD0BYiIDUzNDA4OGNmZjBjYTQ0YmI4Y2I4ZmVlYzJlNGVk****
x-oss-request-id: 5CAC3A80B7AEADE0170005F6
Date: Tue, 09 Apr 2019 06:24:00 GMT
ETag: "1CF5A685959CA2ED8DE6E5F8ACC2****"
Last-Modified: Tue, 09 Apr 2019 06:24:00 GMT
Content-Length: 119914
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [管理文件元信息](#)

错误码

错误码	HTTP状态码	描述
Not Found	404	目标Object不存在。

7.1.9. PostObject

调用PostObject用于通过HTML表单上传的方式将文件（Object）上传到指定存储空间（Bucket）。

注意事项

- 通过PostObject上传的Object大小不能超过5 GB。
- Post请求需要对Bucket拥有写权限。如果Bucket为public-read-write，可以不上传签名信息，否则要求对该操作进行签名验证。
- 与Put操作不同，Post操作使用AccessKey Secret对Policy进行签名，计算出签名字符串作为Signature表单域的值，OSS通过验证该值从而判断签名的合法性。
- 提交表单的URL为Bucket域名，不需要在URL中指定Object。即请求行是 `POST /HTTP/1.1`，不能写为 `POST /ObjectName HTTP/1.1`。
- 如果POST请求中包含Header签名信息或URL签名信息，OSS不会对此类信息做检查。

版本控制

在开启了版本控制的Bucket中发起PostObject请求时，OSS将为新添加的Object自动生成唯一的版本ID，并在响应Header中通过x-oss-version-id的形式返回。

在暂停了版本控制的Bucket中发起PostObject请求时，OSS将为新添加的Object自动生成一个null的版本ID，并在响应Header中通过x-oss-version-id的形式返回。同一个Object仅允许一个null的版本ID。

请求语法

```
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
--9431149156168--
```

请求头

 **说明** PostObject的消息实体通过多重表单格式（multipart/form-data）编码，在PutObject操作中参数通过HTTP请求头传递，在PostObject操作中则作为消息体中的表单域传递。PostObject不支持传入x-oss-tagging请求头。

名称	类型	是否必选	描述
----	----	------	----

名称	类型	是否必选	描述
Content-Type	字符串	否	指定文件的类型和网页的编码，确定浏览器读取文件的形式和编码。 Post操作提交表单编码必须为 <code>multipart/form-data</code>，即Header中Content-Type 为 <code>multipart/form-data;boundary=xxxxxx</code> 的形式。 boundary为边界字符串，是由表单随机生成的一个随机值，无需指定。如果通过SDK拼表单，则SDK也会生成一个随机值。
x-oss-object-acl	字符串	否	在请求头中指定Object的访问权限。 当请求头和文件表单域中同时指定x-oss-object-acl时，文件表单域中设置的访问权限的优先级高于请求中设置的访问权限。 取值： <code>default</code>（默认）：Object遵循所在存储空间的访问权限。 <code>private</code>：Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户没有权限操作该Object。 <code>public-read</code>：Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户只有该Object的读权限。请谨慎使用该权限。 <code>public-read-write</code>：Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息，请参见基于读写权限ACL的权限控制。

此接口还需要包含Host、Date等公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求元素


注意
 所有表单域的key大小不能超过8 KB，表单域的value不能超过2 MB。

名称	类型	是否必选	描述
Cache-Control	字符串	否	指定该Object被下载时网页的缓存行为。更多信息，请参见RFC 2616。 默认值：无

名称	类型	是否必选	描述
Content-Disposition	字符串	否	指定该Object被下载时的名称。更多信息，请参见 RFC 2616 。 默认值：无
Content-Encoding	字符串	否	指定该Object被下载时的内容编码格式。更多信息，请参见 RFC 2616 。 默认值：无
Expires	字符串	否	过期时间。更多信息，请参见 RFC 2616 。 默认值：无
OSSAccessKeyId	字符串	是，有条件	Bucket拥有者的AccessKey ID。 默认值：无 限制：当Bucket为非public-read-write或者提供了Policy（或Signature）表单域时，必须提供OSSAccessKeyId表单域。
policy	字符串	是，有条件	Policy规定了请求表单域的合法性。不包含Policy表单域的请求被认为是匿名请求，并只能访问public-read-write的Bucket。 默认值：无 限制：当Bucket为非public-read-write或者提供了OSSAccessKeyId（或Signature）表单域时，必须提供Policy表单域。 🔔 注意 表单和Policy必须使用UTF-8编码，且Policy表单域要经过Base64编码。
Signature	字符串	是，有条件	根据AccessKey Secret和Policy计算的签名信息，OSS验证该签名信息从而验证该Post请求的合法性。更多信息，请参见 附录：Post Signature 。 默认值：无 限制：当Bucket为非public-read-write或者提供了OSSAccessKeyId（或Policy）表单域时，必须提供Signature表单域。 ❓ 说明 表单域对大小写不敏感，但表单域的值对大小写敏感。
x-oss-server-side-data-encryption	字符串	否	仅当加密方式为KMS时，通过此选项将默认加密算法AES256修改为SM4。 取值：SM4

名称	类型	是否必选	描述
x-oss-server-side-encryption-key-id	字符串	否	表示KMS托管的用户主密钥。此选项仅当x-oss-server-side-encryption值为KMS时有效。
x-oss-content-type	字符串	否	由于浏览器会自动在文件表单域中增加Content-Type, 为了解决此问题, OSS支持用户在Post请求体中增加x-oss-content-type, 该项允许用户指定Content-Type, 且拥有最高优先级。 优先级顺序: x-oss-content-type > 文件表单域的Content-Type > Content-Type 默认值: 无
x-oss-forbid-overwrite	字符串	否	指定PostObject操作时是否覆盖同名Object。 当目标Bucket处于已开启或已暂停的版本控制状态时, x-oss-forbid-overwrite请求Header设置无效, 即允许覆盖同名Object。 - 不指定x-oss-forbid-overwrite指定x-oss-forbid-overwrite为 <i>false</i> 时, 表示允许覆盖同名Object。 - 指定x-oss-forbid-overwrite为 <i>true</i> 时, 表示禁止覆盖同名Object; 设置x-oss-forbid-overwrite请求Header会导致QPS处理性能下降, 如果您有大量的操作需要使用x-oss-forbid-overwrite请求头 (QPS > 1000), 请联系技术支持, 避免影响您的业务。
x-oss-object-acl	字符串	否	在文件表单域中指定Object的访问权限。此项支持同时在请求头和文件表单域中指定。 当请求头和文件表单域中同时指定x-oss-object-acl时, 文件表单域中设置的访问权限的优先级高于请求中设置的访问权限。 取值: <i>default</i> (默认): Object遵循所在存储空间的访问权限。 <i>private</i>: Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限, 其他用户没有权限操作该Object。 <i>public-read</i>: Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限, 其他用户只有该Object的读权限。请谨慎使用该权限。 <i>public-read-write</i>: Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息, 请参见基于读写权限ACL的权限控制。

名称	类型	是否必选	描述
x-oss-storage-class	字符串	否	指定Object的存储类型。 对于任意存储类型的Bucket，如果上传Object时指定此参数，则此次上传的Object将存储为指定的类型。例如在IA类型的Bucket中上传Object时，如果指定x-oss-storage-class为Standard，则该Object直接存储为Standard。 取值： • <i>Standard</i>: 标准存储 • <i>IA</i>: 低频访问 • <i>Archive</i>: 归档存储 • <i>ColdArchive</i>: 冷归档存储 关于存储类型的更多信息，请参见存储类型介绍。
key	字符串	是	上传Object的名称。如果名称包含路径，例如 <code>destfolder/example.jpg</code>，则OSS会自动创建相应的文件夹。 默认值：无
success_action_redirect	字符串	否	上传成功后客户端跳转到的URL。如果未指定该表单域，返回结果由success_action_status表单域指定。如果上传失败，OSS返回错误码，并不进行跳转。 默认值：无
success_action_status	字符串	否	未指定success_action_redirect表单域时，该表单域指定了上传成功后返回给客户端的状态码。 有效值：200、201、204（默认）。 • 如果该域的值设置为200或者204，OSS返回一个空文档和相应的状态码。 • 如果该域的值设置为201，OSS返回一个XML文件和201状态码。 • 如果该域的值未设置或者设置为一个非法值，OSS返回一个空文档和204状态码。
x-oss-meta-*	字符串	否	用户指定的user meta值。 默认值：无 如果请求中携带以x-oss-meta-为前缀的表单域，则视为user meta，例如 <code>x-oss-meta-location</code>。  说明 一个Object可以有多个类似的参数，但所有的user meta总大小不能超过8 KB。

名称	类型	是否必选	描述
x-oss-security-token	字符串	否	如果本次访问是使用STS临时授权方式，则需要指定该项为SecurityToken的值，同时OSSAccessKeyId需要使用与之配对的临时AccessKey Id。计算签名时，与使用普通AccessKey Id签名方式一致。 默认值：无 搭建STS服务的具体操作请参见开发指南中的使用STS临时访问凭证访问OSS。您可以通过调用STS服务的AssumeRole接口或者使用各语言STS SDK来获取临时访问凭证。
file	字符串	是	文件或文本内容。浏览器会自动根据文件类型来设置Content-Type，并覆盖用户的设置。OSS一次只能上传一个文件。 默认值：无  注意 file必须是表单中的最后一个域。

响应头

名称	类型	描述
x-oss-server-side-encryption	字符串	如果请求Header中指定了x-oss-server-side-encryption，则响应Header中将包含该头部，指明所使用的服务器端加密算法。

此接口还需包含公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

响应元素

名称	类型	描述
PostResponse	容器	保存Post请求结果的容器。 子节点：Bucket、ETag、Key、Location
Bucket	字符串	Bucket名称。 父节点：PostResponse
ETag	字符串	ETag (Entity Tag) 在每个Object生成的时候被创建。Post请求创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。ETag值可以用于检查该Object内容是否发生变化。 父节点：PostResponse

名称	类型	描述
Location	字符串	新创建Object的URL。 父节点：PostResponse

示例

- 请求示例：

```
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"
44CF9590006BF252****
--9431149156168
Content-Disposition: form-data; name="policy"
eyJleHBpcmF0aW9uIjoiMjAxMy0xMi0wMVQxMjowMDowMFoiLCJjb25kaXRpb25zIjpbWyJjb250ZW50LWxlbmd0aC1yYW5nZSIsIDAsIDEwNDg1NzYwXSx7ImJ1Y2tldCI6ImFoYWwhIn0sIHsiQS16ICJhIn0seyJrZXkiOiAiQUJDIn1dfQ==
--9431149156168
Content-Disposition: form-data; name="Signature"
kZoYNv66bsmc10+dcGKw5x2P****
--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--
```

- 返回示例：

```
HTTP/1.1 200 OK
x-oss-request-id: 61d2042d-1b68-6708-5906-33d81921362e
Date: Fri, 24 Feb 2014 06:03:28 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5*****"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

错误码

错误码	HTTP状态码	描述
FieldItemTooLong	400	表单域的key或value的大小超出限制。
InvalidArgument	400	无论Bucket是否为public-read-write，一旦上传OSSAccessKeyId、Policy、Signature表单域中的任意一个，则另两个表单域为必选项。如果缺失，将返回此错误。
InvalidDigest	400	用户上传了Content-MD5请求Header，OSS会计算body的Content-MD5并检查一致性，如果不一致，将返回此错误。
EntityTooLarge	400	请求的body总长度不允许超过5 GB。如果文件长度过大，将返回此错误。
InvalidEncryptionAlgorithmError	400	如果上传时指定了x-oss-server-side-encryption Header请求域，则必须设置其值为AES256、KMS或SM4。如果设置为其他值，将返回此错误。
IncorrectNumberOfFilesInPOSTRequest	400	请求必须指定key表单域。如果缺失，将返回此错误。
FileAlreadyExists	409	请求的Header中携带x-oss-forbid-overwrite=true时，表示禁止覆盖同名文件。如果文件已存在，将返回此错误。
KmsServiceNotEnabled	403	将x-oss-server-side-encryption指定为KMS，但没有预先购买KMS套件。
FileImmutable	409	Bucket内的数据处于被保护状态时，如果您尝试删除或修改这些数据，将返回此错误码。

附录：Post Policy

Post请求的Policy表单域用于验证请求的合法性。Policy为一段经过UTF-8和Base64编码的JSON文本，声明了Post请求必须满足的条件。虽然对于public-read-write的Bucket上传时，Post表单域为可选项，但强烈建议使用该域来限制Post请求。

Post Policy示例如下：

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

Post Policy中必须包含Expiration和Conditions。

- Expiration

Expiration项指定了Policy的过期时间，以ISO8601 GMT 时间表示。例如 2014-12-01T12:00:00.000Z 指定了Post请求必须在2014年12月1日12点之前。

- Conditions

Conditions是一个列表，可以用于指定Post请求的表单域的合法值。

 **说明** 表单域对应的值在检查Policy之后进行扩展，因此，Policy中设置的表单域的合法值应当对应于扩展之前的表单域的值。

Policy中支持的Conditions项请参见下表。

名称	描述
content-length-range	上传Object的最小和最大允许大小。该Condition支持content-length-range匹配方式。
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	HTTP请求Header。该Condition支持精确匹配和starts-with匹配方式。  注意 建议您在Policy中增加Content-Type限制，防止通过表单上传时Content-Type被恶意修改。
key	Object名称。该Condition支持精确匹配和starts-with匹配方式。
success_action_redirect	上传成功后的跳转URL地址。该Condition支持精确匹配和starts-with匹配方式。
success_action_status	未指定success_action_redirect时，上传成功后的返回状态码。该Condition支持精确匹配和starts-with匹配方式。
x-oss-meta-*	用户指定的user meta。该Condition支持精确匹配和starts-with匹配方式。

如果Post请求中包含额外的表单域，OSS可以将这些额外的表单域加入到Policy的Conditions中并检查合法性。

- Conditions匹配方式

Conditions匹配方式	描述
精确匹配	表单域的值必须精确匹配Conditions中声明的值。例如指定key表单域的值必须为a: {"key": "a"}，则可以写为["eq", "\$key", "a"]。
Starts With	表单域的值必须以指定值开始。例如指定key的值必须以user/user1开始，则可以写为["starts-with", "\$key", "user/user1"]。
指定文件大小范围	指定所允许上传的文件最大和最小范围，例如允许的文件大小为1~10字节，则可以写为["content-length-range", 1, 10]。

在Post Policy中\$表示变量。如果要描述\$，需要使用转义字符\\$。下表描述了在Post Policy的JSON中需要进行转义的字符。

转义字符	描述
\/	斜杠
\\	反斜杠
\"	双引号
\\$	美元符
\b	空格
\f	换页
\n	换行
\r	回车
\t	水平制表符
\uxxxx	Unicode字符

附录：Post Signature

对于验证的Post请求，HTML表单中必须包含Policy和Signature信息。Policy控制请求中哪些值是允许的。计算Signature的具体流程如下：

- 1. 创建一个UTF-8编码的Policy。
- 2. 将Policy进行Base64编码，其值即为Policy表单域填入的值，将该值作为将要签名的字符串。
- 3. 使用AccessKeySecret对要签名的字符串进行签名，签名方法为 `Signature = base64(hmac-sha1(base64(policy), AccessKeySecret))`。

更多参考

- Web端表单直传OSS示例：[JavaScript客户端签名直传](#)
- Java示例：[表单上传](#)

7.1.10. Callback

您只需在发送给OSS的请求中携带相应的Callback参数即能实现回调。本文介绍Callback的实现原理。

② 说明

- 目前支持Callback的API接口包括Put Object、Post Object和CompleteMultipartUpload。关于Callback的更多信息，请参见[原理介绍](#)。
- Callback目前不支持服务器名称指示SNI（Server Name Indication）。

步骤1：构造参数

• Callback参数

Callback参数是由一段经过Base64编码的JSON字符串（字段）。构建callback参数的关键是指定请求回调的服务器URL（callbackUrl）以及回调的内容（callbackBody）。

JSON字段示例如下：

```
{
  "callbackUrl": "172.16.XX.XX/test.php",
  "callbackHost": "oss-cn-hangzhou.aliyuncs.com",
  "callbackBody": "{\"mimeType\":\"${mimeType}\",\"size\":\"${size}\"}",
  "callbackBodyType": "application/json"
}
```

```
{
  "callbackUrl": "172.16.XX.XX: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}"
}
```

JSON字段的详细信息请参见下表。

字段	是否必选	示例值	描述
----	------	-----	----

字段	是否必选	示例值	描述
callbackUrl	是	172.16.XX.XX:23456/index.html	文件上传成功后，OSS向此URL发送回调请求，请求方法为POST，body为callbackBody指定的内容，支持使用HTTPS地址。 正常情况下，该URL需要响应 HTTP/1.1 200 OK，body必须为JSON格式，响应头Content-Length必须为合法的值，且大小不超过3 MB。 支持同时配置最多5个URL，多个URL间以分号（;）分隔。OSS会依次发送请求直到第一个返回成功为止。如果未配置或者配置值为空则表示未配置callback。 为了保证正确处理中文等情况，callbackUrl需做URL编码处理。例如 <code>http://example.com/中文.php?key=value&中文名称=中文值</code> 需要编码成 <code>http://example.com/%F4%B8%AD%F6%96%87.nhn?key=value&%F4%B8%AD%F6%96%87%E5%90%8D%E7%A7%B0=%E4%B8%AD%E6%96%87%E5%80%BC</code>。
callbackHost	否	oss-cn-hangzhou.aliyuncs.com	发起回调请求时Host头的值，该值必须符合域名或IP地址规则。只有在设置了callbackUrl时才有效。 如果未配置callbackHost，则OSS会解析callbackUrl中的URL并将解析出的host填充到callbackHost中。
callbackBody	是	<code>{"mimeType\":\"\${mimeType}\",\"size\":\"\${size}"}</code>	发起回调时请求body的值，例如 <code>key=\${object}&etag=\${etag}&my_var=\${x:my_var}</code>。该值中支持使用OSS系统变量、自定义变量和常量。 - 支持的系统变量请参见下表。 - 自定义变量的支持方式在PutObject和CompleteMultipart中是通过callback-var来传递，在PostObject中则是将各个变量通过表单域来传递。

字段	是否必选	示例值	描述
callbackBodyType	否	application/json	发起回调请求的Content-Type，取值范围如下： - application/x-www-form-urlencoded（默认）：callbackBody中的变量将会被经过URL编码的值替换掉。 - application/json：会按照JSON格式替换其中的变量。

callbackBody中支持设置的系统变量请参见下表。

 **说明** imageInfo针对图片格式，如果为非图片格式，imageInfo.height、imageInfo.width、imageInfo.format都为空。

系统变量	描述
bucket	存储空间。
object	对象（文件）。
etag	文件的ETag，即返回给用户的ETag字段。
size	Object大小，CompleteMultipartUpload时为整个Object的大小。
mimeType	资源类型，例如jpeg图片的资源类型为image/jpeg。
imageInfo.height	图片高度。
imageInfo.width	图片宽度。
imageInfo.format	图片格式，例如jpg、png等。

● callback-var自定义参数

用户可以通过callback-var参数来配置自定义参数。自定义参数是一个Key-Value的Map，用户可以配置自己需要的参数到该Map。在OSS发起POST回调请求时，会将这些参数和上述的系统参数一起放在POST请求的body中以方便接收回调方获取。

构造自定义参数的方法和callback参数的方法相同，也是以JSON格式来传递。该JSON字符串是一个包含所有自定义参数的Key-Value的Map。

 **说明** 用户自定义参数的Key一定要以x:开头且必须为小写，否则虽然OSS会返回200，但是自定义参数未被正确赋值。

假如您需要配置的两个自定义参数分别为x:var1和x:var2，对应的值分别为value1和value2，那么构造出来的JSON格式如下：

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

❓ 说明 如果传入的callback或者callback-var参数不合法，则会返回400错误，错误码为“InvalidArgument”，不合法的情况包括以下几类：

- PutObject()和CompleteMultipartUpload()接口中URL和header同时传入callback(x-oss-callback)或者callback-var(x-oss-callback-var)参数。
- callback或者callback-var参数过长，超过5 KB。由于PostObject()没有callback-var参数，因此无此限制，下同。
- callback或者callback-var参数没有经过base64编码或经过base64解码后不是合法的JSON格式。
- callback参数解析后callbackUrl字段包含的URL超过限制（5个），或者URL中传入的port不合法，例如

```
{"callbackUrl":"172.16.XX.XX:test",
  "callbackBody":"test"}
```

- callback参数解析后callbackBody字段为空。
- callback参数解析后callbackBodyType字段的值不是 application/x-www-form-urlencoded 或者 application/json 。
- callback参数解析后callbackBody字段中变量的格式不合法，合法的格式为\${var}。
- callback-var参数解析后不是预期的JSON格式，预期的格式为 {"x:var1":"value1","x:var2":"value2"} 。

步骤2：构造回调请求

构造完成callback和callback-var两个参数后，需将参数附加到OSS的请求中。

附加方式共有三种，方式如下：

- 在URL中携带参数。
- 在Header中携带参数。
- 在POST请求的body中使用表单域来携带参数。

❓ 说明 在使用PostObject接口上传Object时只能使用这种方式来指定回调参数。

以上三种方式只能同时使用其中一种，否则OSS会返回InvalidArgument错误。

要将参数附加到OSS的请求中，首先要将上文构造的JSON字符串使用base64编码，然后按照如下的方法附加到OSS的请求中。

- 如果在URL中携带参数。把 callback=[CallBack] 或者 callback-var=[CallBackVar] 作为一个URL参数带入请求发送。计算签名CanonicalizedResource时，将callback或者callback-var当做一个sub-resource计算在内。
- 如果在Header中携带参数。把 x-oss-callback=[CallBack] 或者 x-oss-callback-var=[CallBackVar] 作为一个Header带入请求发送。在计算签名CanonicalizedOSSHeaders时，将x-oss-callback-var和x-oss-callback计算在内。示例如下：

```

PUT /test.txt HTTP/1.1
Host: callback-test.oss-test.aliyun-inc.com
Accept-Encoding: identity
Content-Length: 5
x-oss-callback-var: eyJ4Om15X3Zhcil6ImZvci1jYWxsYmFjay10ZXN0In0=
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: eyJjYWxsYmFja1VyYmFja1VybCI6IjEwMS40My4xMTMuODoyMzQ1Ni9pbmRleC5odG1sliwglCJjYWxsYmFja0JvZm9iaWwJHidWNrZXQ9JHtidWNrZXR9Jm9iamVjdD0ke29iamVjdH0mZXRhZz0ke2V0YWd9JnNpemU9JHtzaXplfSZtaW1lVHlwZT0ke21pbWVUeXBIfSZpbWFnZUluZm8uaGVpZ2h0PSR7aW1hZ2VJbmZvLmhlaWdo dH0maW1hZ2VJbmZvLndpZHRoPSR7aW1hZ2VJbmZvLndpZHRofSZpbWFnZUluZm8uZm9ybWF0PSR7aW1hZ2VJbmZvLmZvcmlhdH0mbXlfdmFyPSR7eDpteV92YXJ9In0=
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 mlepow3zr4u7b14:5a74vhd4UXpmyuudV14Kaen5****
Test

```

- 在POST请求的body中使用表单域来携带参数。
 - 如果需要在POST上传Object时附带回调参数，callback参数要使用独立的表单域来附加，示例如下：

```

--9431149156168
Content-Disposition: form-data; name="callback"
eyJjYWxsYmFja1VybCI6IjEwMS40My4xMTMuODoyMzQ1Ni9pbmRleC5odG1sliwglCJjYWxsYmFja0hvc3QiO ilxMC4xMDEuMTY2LjMwliwiY2FsbGJhY2tCb2R5IjoizmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9JHt4 OnRhYmxlfSIsImNhbmGxiYWNRQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVyYGVuY29kZW QifQ==

```

- 如果是自定义参数，不能直接将callback-var参数直接附加到表单域中，每个自定义的参数都需要使用独立的表单域来附加。如果用户的自定义参数JSON字段示例如下：

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

那么POST请求的表单域为：

```
--9431149156168
Content-Disposition: form-data; name="callback"
eyJjYWxsYmFja1VyY2tCb2R5ljoizmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9JHt4OnRhYmxfSlsImNhGxiYWNRQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVyYGVuY29kZWQifQ==
--9431149156168
Content-Disposition: form-data; name="x:var1"
value1
--9431149156168
Content-Disposition: form-data; name="x:var2"
value2
```

同时可以在policy中添加callback条件，如果不添加callback条件，则不对该参数做上传验证，如下所示。

```
{ "expiration": "2021-12-01T12:00:00.000Z",
  "conditions": [
    { "bucket": "johnsmith" },
    { "callback": "eyJjYWxsYmFja1VyY2tCb2R5ljoizmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9JHt4OnRhYmxfSlsImNhGxiYWNRQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVyYGVuY29kZWQifQ==",
      [ "starts-with", "$key", "user/eric/" ]
    ]
  }
```

步骤3：发起回调请求

如果文件上传成功，OSS会根据用户请求中的callback参数和callback-var自定义参数，将特定内容以POST方式发送给应用服务器。

```
POST /index.html HTTP/1.0
Host: 172.16.XX.XX
Connection: close
Content-Length: 181
Content-Type: application/x-www-form-urlencoded
User-Agent: http-client/0.0.1
bucket=callback-test&object=test.txt&etag=D8E8FCA2DC0F896FD7CB4CB0031BA249&size=5&mimeType=te
xt%2Fplain&imageInfo.height=&imageInfo.width=&imageInfo.format=&x:var1=for-callback-test
```

（可选）步骤4：回调签名

用户设置callback参数后，OSS将按照用户设置的callbackUrl发送POST回调请求给用户的应用服务器。应用服务器收到回调请求之后，如果希望验证回调请求确实是由OSS发起的话，可以通过在回调中带上签名来验证OSS的身份。

● 生成签名

签名发生在OSS端，采用RSA非对称方式。

私钥加密生成签名的过程为：

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

 **说明** 其中private_key为私钥，只有OSS知晓，path为回调请求的资源路径，query_string为查询字符串，body为回调的消息体。

签名过程分为以下三步：

- i. 获取待签名字符串：资源路径经过URL解码后，加上原始的查询字符串、加上一个回车符、然后加上回调消息体。
- ii. RSA签名：使用密钥对待签名字符串进行签名，签名的hash函数为md5。
- iii. 将签名后的结果做base64编码，得到最终的签名，签名放在回调请求的authorization头中。

示例如下：

```
POST /index.php?id=1&index=2 HTTP/1.0
Host: 172.16.XX.XX
Connection: close
Content-Length: 18
authorization: kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2t****
Content-Type: application/x-www-form-urlencoded
User-Agent: http-client/0.0.1
x-oss-pub-key-url: aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9jYWxsYmFja19wdWJfa2V5X3YxLn*****
bucket=yonghu-test
```

path为 /index.php，query_string为 ?id=1&index=2，body为 bucket=yonghu-test，最终签名结果为 kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA=。

● 验证签名

验证签名的过程即为签名的逆过程，由应用服务器验证，过程如下：

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

字段的含义与签名过程中描述相同，其中public_key为公钥，authorization为回调头中的签名，整个验证签名的过程分为以下几步：

- i. 回调请求的x-oss-pub-key-url头保存的是公钥URL地址的base64编码，因此需要对其做base64解码后获取到公钥，即

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


② 说明 为了保证该public key是由OSS颁发的，用户需要校验 x-oss-pub-key-url 头的值必须以 http://gosspublic.alicdn.com/ 或者 https://gosspublic.alicdn.com/ 开头。

ii. 获取base64解码后的签名

```
signature = base64_decode(authorization头的值)
```

iii. 获取待签名字符串，方法与签名一致

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

iv. 验证签名

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

以上述示例来说明验证签名的完整过程：

- i. 获取到公钥的URL地址，即 aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9iYWxsYmFia19wdWJfa2V5X3YxLnBibQ== 经过base64解码后得到 http://gosspublic.alicdn.com/callback_pub_key_v1.pem 。
- ii. 签名头 kKOeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA== 做base64解码（由于为非打印字符，无法显示出解码后的结果）。
- iii. 获取待签名字符串，即url_decode(“index.php”) + “?id=1&index=2” + “\n” + “bucket=yonghu-test”，并做MD5校验。
- iv. 验证签名。

● 应用服务器示例

如下以Python为例，演示了一个简单的应用服务器，主要说明验证签名的方法，此示例需要安装M2Crypto库。

```
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("htt
ps://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 according 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):

```

```
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()
```

其它语言的服务端代码请参见下表。

SDK语言	描述
Java	- 下载地址: Java - 运行方法: 解压包运行 <code>java -jar oss-callback-server-demo.jar 9000</code> (9000指运行的端口, 可以自行指定)。
Python	- 下载地址: Python - 运行方法: 解压包直接运行 <code>python callback_app_server.py</code>, 运行该程序需要安装RSA的依赖。
Go	- 下载地址: Go - 运行方法: 解压后参看README.md。
PHP	- 下载地址: PHP - 运行方法: 部署到Apache环境下, 因为PHP本身语言的特点, 取一些数据头部会依赖于环境。所以可以参考例子根据所在环境修改。
.NET	- 下载地址: .NET - 运行方法: 解压后参看 <code>README.md</code>。
Node.js	- 下载地址: Node.js - 运行方法: 解压包直接运行 <code>node example.js</code>。
Ruby	- 下载地址: Ruby - 运行方法: <code>ruby aliyun_oss_callback_server.rb</code>

步骤5：返回回调结果

应用服务器返回响应给OSS。

返回的回调请求为：

```
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"}
```

 **说明** 应用服务器返回OSS的响应必须带有Content-Length的Header，Body大小不能超过1 MB。

步骤6：返回上传结果

OSS将应用服务器返回的内容返回给用户。

返回的内容响应为：

```
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"}
```

 **说明**

- 如果类似CompleteMultipartUpload这样的请求，在返回请求本身body中存在内容（例如XML格式的信息），使用上传回调功能后会覆盖原有的body中的内容例如 `{"a":"b"}`，希望对此处做好判断处理。
- 如果回调失败，则返回203，错误码为“CallbackFailed”。文件已经成功上传到了OSS，但回调失败。回调失败只是表示OSS没有收到预期的回调响应，不代表应用服务器没有收到回调请求（例如应用服务器返回的内容不是JSON格式）。

7.1.11. RestoreObject

调用RestoreObject接口解冻归档类型（Archive）或冷归档（Cold Archive）的文件（Object）。

版本控制

Object的各个版本可以对应不同的存储类型。调用RestoreObject接口默认解冻Object当前版本，您可以通过指定versionId的方式来解冻Object指定版本。

 **说明**

- RestoreObject接口只针对归档或冷归档类型的Object，不适用于标准类型和低频访问类型的Object。
- 如果针对该Object第一次调用RestoreObject接口，则返回202。
- 如果已经成功调用过RestoreObject接口，且Object已完成解冻，再次调用时返回200 OK。

解冻过程说明

对于归档类型或者冷归档类型的Object，如果需要读取Object，请提前解冻。归档类型的Object解冻有分钟级延迟，冷归档类型的Object解冻有数小时延迟。

归档类型或者冷归档类型的Object在执行解冻前后的状态变换过程如下：

- 1. Object初始时处于冷冻状态。
- 2. 提交一次解冻请求后，Object处于解冻中状态。
- 3. 服务端完成解冻任务后，Object进入解冻状态，此时您可以读取Object。

对于归档类型的Object，解冻状态默认持续24小时，24小时内再次调用RestoreObject接口则解冻状态会自动延长24小时，一次解冻流程内可有效调用7次RestoreObject接口达到最长7天的解冻持续时间。您也可以通过传入解冻天数，一次调用RestoreObject接口指定最长7天的解冻持续时间。

对于冷归档类型的Object，您可以指定解冻天数和解冻优先级，解冻天数最短为1天，最长为7天。不同解冻优先级的首字节取回时间如下：

- 高优先级（Expedited）：表示1小时内完成解冻。
 - 标准（Standard）：表示2~5小时内完成解冻。如果不传入JobParameters节点，则默认为Standard。
 - 批量（Bulk）：表示5~12小时内完成解冻。
- 4. 解冻状态结束后，Object再次返回到冷冻状态。

计费说明

状态变换过程中产生的相关费用如下：

- 对处于冷冻状态的Object执行解冻操作，会产生数据取回费用。
- 解冻状态最多延长7天。在此期间不再重复收取数据取回费用。
- 解冻状态结束后，Object又回到冷冻状态，再次执行解冻操作会收取数据取回费用。

请求语法

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

请求头

此接口仅包含公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

请求参数

名称	类型	是否必选	示例值	描述
RestoreRequest	容器	是	不涉及	解冻请求信息。 子节点：Days、JobParameters
Days	整型	是	2	解冻的天数。取值范围为1~7天。 父节点：RestoreRequest

名称	类型	是否必选	示例值	描述
JobParameters	容器	否	不涉及	解冻优先级的容器。只有解冻冷归档类型的Object时才生效。 如果不传入JobParameters节点，则解冻优先级默认为Standard。 父节点：RestoreRequest 子节点：Tier
Tier	字符串	是	Standard	解冻优先级。取值范围如下： - 高优先级（Expedited）：表示1小时内完成解冻。 - 标准（Standard）：表示2~5小时内完成解冻。 - 批量（Bulk）：表示5~12小时内完成解冻。 父节点：JobParameters

响应头

名称	类型	示例值	描述
x-oss-object-restore-priority	字符串	Standard	解冻优先级。只有解冻处于解冻状态的冷归档类型的Object时才显示该字段。
x-oss-version-id	字符串	CAEQNRiBgMClj7qD0BYiIDQ5Y2QyMjc3NGZkODRlMTU5M2VkY2U3MWRiNGRh****	Object的版本ID。只有解冻Object指定版本时才显示该字段。

此接口还包含公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 首次对处于冷冻状态的归档类型Object提交解冻请求

请求示例

```
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 e1Unnbm1rg****:y4eyu+4yje5ioRCr****
```

返回示例

```
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
```

- 对处于解冻中状态的归档类型Object提交解冻请求

请求示例

```
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 e1Unnbm1rg****:21qtGJ+ykDVmdy4eyu+N****
```

返回示例

```
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.XX.XX</HostId>
</Error>
```

- 对处于解冻状态的归档类型Object提交解冻请求

请求示例

```
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 e1Unnbm1rg****:u6O6FMJnn+WuBwbByZxm1+y4eyu+N****
<RestoreRequest>
  <Days>2</Days>
</RestoreRequest>
```

返回示例

```
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
```

- 对处于解冻状态的冷归档类型Object提交解冻请求

请求示例

```
POST /coldarchiveobject?restore HTTP/1.1
Host: cold-archive-bucket.oss-cn-hangzhou.aliyuncs.com
User-Agent: aliyun-sdk-go/v2.1.0 (Darwin/17.5.0/x86_64;go1.11.8)/ossutil-v1.6.12
Content-Length: 99
Authorization: OSS LTAI4FjmjjhiYK6kMaV****:Gi1x7YHqTw+NQCJo0fKBHcYQ****
Content-Type: text/plain; charset=utf-8
Date: Tue, 21 Apr 2020 11:09:19 GMT
Accept-Encoding: gzip
<RestoreRequest>
  <Days>2</Days>
  <JobParameters>
    <Tier>Standard</Tier>
  </JobParameters>
</RestoreRequest>
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 21 Apr 2020 11:09:19 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5E9ED45F093E2F3930318EA0
x-oss-object-restore-priority: Standard
x-oss-server-time: 10
```

- 指定versionId来解冻指定版本的Object

请求示例

```
POST /oss.jpg?restore&versionId=CAEQNRiBgMClj7qD0BYiIDQ5Y2QyMjc3NGZkODRIMTU5M2VkY2U3MWRiNGRh**** HTTP/1.1
Host: oss-archive-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:50:48 GMT
Authorization: OSS o3shiyktjw1****:2JND5qqlAlaA1/kLO4kBbGTw****
```

返回示例

```
HTTP/1.1 202 Accepted
Date: Tue, 09 Apr 2019 06:50:48 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
x-oss-version-id: CAEQNRiBgMClj7qD0BYiIDQ5Y2QyMjc3NGZkODRIMTU5M2VkY2U3MWRiNGRh****
x-oss-request-id: 5CAC40C8B7AEADE017000653
```

SDK

此接口对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)

- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
OperationNotSupported	400	目标Object不支持解冻，该Object不是归档或冷归档类型。
NoSuchKey	404	目标Object不存在。
RestoreAlreadyInProgress	409	您已经成功调用过RestoreObject接口，且服务端正在执行解冻操作。请不要重复提交RestoreObject。

7.1.12. SelectObject

SelectObject用于对目标文件执行SQL语句，返回执行结果。

 说明

- 此操作要求您对该Object有读权限。
- 正确执行时，该API返回206。如果SQL语句不正确，或者和文件不匹配，则会返回400错误。
- 关于SelectObject的功能介绍请参见开发指南中的[SelectObject](#)。

请求语法

请求语法分为 CSV 和 JSON 两种格式。

- 请求语法（CSV）

```

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>
  <Expression>base64 encode的SQL语句，例如c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2JqZWN0IHdoZX
  JllF80ID4gNDU=</Expression>
  <InputSerialization>
    <CompressionType>None|GZIP</CompressionType>
    <CSV>
      <FileHeaderInfo>
        NONE|IGNORE|USE
      </FileHeaderInfo>
      <RecordDelimiter>base64 encode的字符</RecordDelimiter>
      <FieldDelimiter>base64 encode的字符</FieldDelimiter>
      <QuoteCharacter>base64 encode的字符</QuoteCharacter>
      <CommentCharacter>base64 encode的字符</CommentCharacter>
      <Range>line-range=start-end|split-range=start-end</Range>
      <AllowQuotedRecordDelimiter>true|false</AllowQuotedRecordDelimiter>
    </CSV>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <RecordDelimiter>base64 encode的字符</RecordDelimiter>
      <FieldDelimiter>base64 encode的字符</FieldDelimiter>
    </CSV>
    <KeepAllColumns>>false|true</KeepAllColumns>
    <OutputRawData>>false|true</OutputRawData>
    <EnablePayloadCrc>true</EnablePayloadCrc>
    <OutputHeader>>false</OutputHeader>
  </OutputSerialization>
  <Options>
    <SkipPartialDataRecord>>false</SkipPartialDataRecord>
    <MaxSkippedRecordsAllowed>
      max allowed number of records skipped
    </MaxSkippedRecordsAllowed>
  </Options>
</SelectRequest>

```

- 请求语法 (JSON)

```
POST /object?x-oss-process=json/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>
  <Expression>
    base64 encode的SQL语句，例如c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2JqZWN0IHdoZXJlIF80ID4gN
DU=
  </Expression>
  <InputSerialization>
    <CompressionType>None|GZIP</CompressionType>
    <JSON>
      <Type>DOCUMENT|LINES</Type>
      <Range>
        line-range=start-end|split-range=start-end
      </Range>
      <ParseJsonNumberAsString> true|false
    </ParseJsonNumberAsString>
    </JSON>
  </InputSerialization>
  <OutputSerialization>
    <JSON>
      <RecordDelimiter>
        Base64 of record delimiter
      </RecordDelimiter>
    </JSON>
    <OutputRawData>false|true</OutputRawData>
    <EnablePayloadCrc>true</EnablePayloadCrc>
  </OutputSerialization>
  <Options>
    <SkipPartialDataRecord>
      false|true
    </SkipPartialDataRecord>
    <MaxSkippedRecordsAllowed>
      max allowed number of records skipped
    </MaxSkippedRecordsAllowed>
  </Options>
</SelectRequest>
```

请求元素

名称	类型	描述
SelectRequest	容器	保存Select请求的容器 子节点： Expression, InputSerialization, OutputSerialization 父节点：None

名称	类型	描述
Expression	字符串	以Base64 编码的SQL语句 子节点：None 父节点：SelectRequest
InputSerialization	容器	输入序列化参数（可选） 子节点：CompressionType、CSV、JSON 父节点：SelectRequest
OutputSerialization	容器	输出序列化参数（可选） 子节点：CSV、JSON、OutputRawData 父节点：SelectRequest
CSV(InputSerialization)	容器	输入CSV的格式参数（可选） 子节点：FileHeaderInfo、RecordDelimiter、FieldDelimiter、QuoteCharacter、CommentCharacter、Range 父节点：InputSerialization
CSV(OutputSerialization)	容器	输出CSV的格式参数（可选） 子节点：RecordDelimiter、FieldDelimiter 父节点：OutputSerialization
JSON(InputSerialization)	容器	输入JSON的格式参数（可选） 子节点：Type、Range、ParseJsonNumberAsString
JSON(OutputSerialization)	容器	输出JSON的格式参数（可选） 子节点：RecordDelimiter
Type	枚举	指定输入JSON的类型：DOCUMENT 、 LINES

名称	类型	描述
OutputRawData	bool, 默认false	指定输出数据为纯数据（可选） 子节点：None 父节点：OutputSerialization  说明 - 您在请求中指定OutputRawData值时，OSS服务端会按照请求中的要求返回数据。 - 您在请求中不指定OutputRawData值时，OSS服务端会自动选择一种格式返回。 - 当您显式地指定OutputRawData为True时，如果该SQL长时间没有返回数据，则HTTP请求可能因没有数据返回而超时。
CompressionType	枚举	指定文件压缩类型：None GZIP 子节点：None 父节点：InputSerialization
FileHeaderInfo	枚举	指定CSV文件头信息（可选） 取值： - Use：该CSV文件有头信息，可以用CSV列名作为Select中的列名。 - Ignore：该CSV文件有头信息，但不可用CSV列名作为Select中的列名。 - None：该文件没有头信息，为默认值。 子节点：None 父节点：CSV（输入）
RecordDelimiter	字符串	指定换行符，以Base64编码。默认值为 <code>\n</code>（可选）。未编码前的值最多为两个字符，以字符的ANSI值表示，例如在Java中使用 <code>\n</code> 表示换行。 子节点：None 父节点：CSV（输入、输出），JSON（输出）
FieldDelimiter	字符串	指定CSV列分隔符，以Base64编码。默认值为 <code>,</code>（可选）。未编码前的值必须为一个字符，以字符的ANSI值表示，例如在Java中使用 <code>,</code> 表示逗号。 子节点：None 父节点：CSV（输入，输出）

名称	类型	描述
QuoteCharacter	字符串	指定CSV的引号字符，以Base64编码。默认值为 <code>\</code>（可选）。在CSV中引号内的换行符，列分隔符将被视作普通字符。未编码前的值必须为一个字符，以字符的ANSI值表示，例如在Java中使用 <code>\</code> 表示引号。 子节点：None 父节点：CSV（输入）
CommentCharacter	字符串	指定CSV的注释符，以Base64编码。默认值为空（即没有注释符）。
Range	字符串	指定查询文件的范围（可选）。支持两种格式：  说明 使用Range参数查询的文件需要有 select meta。select meta详情请参考CreateSelectObjectMeta。 - 按行查询：line-range=start-end。例如line-range=10-20表示扫描第10行到第20行。 - 按Split查询：split-range=start-end。例如split-range=10-20表示扫描第10到第20个split。 其中start和end均为inclusive。其格式和range get中的range参数一致。 仅在文档是CSV或者JSON Type为LINES时使用。 子节点：None 父节点：CSV（输入）、JSON（输入）
KeepAllColumns	bool	指定返回结果中包含CSV所有列的位置（可选，默认值为false）。但仅仅在select 语句里出现的列会有值，不出现的列则为空，返回结果中每一行的数据按照CSV列的顺序从低到高排列。例如以下语句： <pre>select _5, _1 from ossobject.</pre> 如果KeepAllColumns = true，假设一共有6列数据，则返回以下数据： Value of 1st column,,,,,Value of 5th column,\n 子节点：None 父节点：OutputSerialization（Csv）
EnablePayloadCrc	bool	在每个Frame中会有一个32位的crc32校验值。客户端可以计算相应payload的Crc32值进行数据完整性校验。 子节点：None 父节点：OutputSerialization

名称	类型	描述
Options	容器	额外的可选参数。 子节点：SkipPartialDataRecord、MaxSkippedRecordsAllowed 父节点：SelectRequest
OutputHeader	bool	在返回结果开头输出CSV头信息。 默认false。 子节点：None 父节点：OutputSerialization
SkipPartialDataRecord	bool	忽略缺失数据的行。当该参数为false时，OSS会忽略缺失某些列（该列值当做null）而不报错。当该参数为true时，该行数据因为不完整而被整体跳过。当跳过的行数超过指定的最大跳过行数时OSS会报错并停止处理。 默认false。 子节点：None 父节点：Options
MaxSkippedRecordsAllowed	Int	指定最大能容忍的跳过的行数。当某一行数据因为不匹配SQL中期望的类型、或者某一列或者多列数据缺失且SkipPartialDataRecord为True时，该行数据会被跳过。如果跳过的行数超过该参数的值，OSS会停止处理并报错。 ❓ 说明 如果某一行是非法CSV行，例如在一列中间连续含有奇数个quote字符，则OSS会马上停止处理并报错，因为该错误很可能会影响对整个CSV文件的解析。即该参数用来调整对非整齐数据的容忍度，但不应用于非法的CSV文件。 默认 0。 子节点：None 父节点：Options

名称	类型	描述
ParseJsonNumberAsString	bool	将JSON中的数字（整数和浮点数）解析成字符串。目前JSON中的浮点数解析时会损失精度，如果要完整保留原始数据，则推荐用该选项。如果需要进行数值计算，则可以在SQL中cast成需要的格式，例如int、double、decimal。 默认值：false 子节点：None 父节点：JSON
AllowQuotedRecordDelimiter	bool	指定CSV内容是否含有在引号中的换行符。 例如某一列值为"abc\ndef"（此处\n为换行），则该值需设置为true。当该值为false时，select支持header range的语义，可以更高效的进行分片查询。 默认值：true 子节点：None 父节点：InputSerialization

返回Body

- 当请求响应中HTTP Status是4xx，表明该请求没有通过相应的SQL语法检查或者目标文件有问题，返回错误信息的Body与Get API返回的错误信息一致。
- 当请求响应中HTTP Status是5xx，表明服务器内部错误，返回的错误信息的格式和Get API返回的错误信息一致。
- 当请求成功返回206时：
 - 若header x-oss-select-output-raw值为true，则表明返回结果是对象数据（而不是Frame包装的），客户端可以完全按照Get API的方式获取数据。
 - 若x-oss-select-output-raw的值为false，请求结果以Frame形式返回。
- Frame的格式为 `Version|Frame-Type|Payload Length|Header Checksum|Payload|Payload Checksum<1 byte><--3 bytes--><---4 bytes----><-----4 bytes--><variable><----4bytes----->`。

 **说明** Checksum均为CRC32。Frame里所有的整数均以大端编码（big endian），且Version目前为1。

Frame Type

Select Object分为Data Frame、Continuous Frame以及End Frame三种不同的Frame Type，详细信息请参见下表。

名称	Frame-Type值	Payload格式	描述
Data Frame	8388609	offset data<-8 bytes> <---variable-->	包含Select请求返回的数据，并用offset来汇报进展。offset为当前扫描位置（从文件头开始的偏移），值为8位整数。

名称	Frame-Type值	Payload格式	描述
Continuous Frame	8388612	offset<---8 bytes-->	用以汇报当前进展以及维持HTTP连接。如果该查询在5s内未返回数据，则返回一个Continuous Frame。
End Frame	8388613	offset total scanned bytes http status code error message<--8bytes-><--8bytes-----><---4 bytes-----><-variable----->	用来返回最终的状态，扫描的字节数以及可能的出错信息。 - offset为扫描后最终的位置偏移。 - total scanned bytes为最终扫描过的数据大小。 - http status code为最终的处理结果。  说明 返回status code的原因在于SelectObject为流式处理，因而在发送Response Header时仅处理了第一个Block。如果第一个Block数据和SQL匹配，则在Response Header中的Status为206。如果如下的数据非法，且已无法更改Header中的Status，则只能在End Frame里包含最终的Status及其出错信息。此时客户端应视其为最终状态。 - error message为错误信息，包括所有跳过的行号及其总行数。  说明 error message格式为 <code>ErrorCodes.DetailMessage</code>。其中ErrorCodes包含一个或者多个ErrorCode，多个ErrorCode之间用逗号隔开。ErrorCodes和DetailMessage之间用句号（.）隔开。

样例请求

样例请求分为 CSV 和 JSON 两种格式。

- 样例请求（CSV）

```
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 LTAIJPLocA0fD:FC/9JRbBGRw4o2QqdaL246Px****
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>c2VsZWN0IGNvdW50KCopIGZyb20gb3Nzb2JqZWN0IHdoZXJlIF80ID4gNDU=
  </Expression>
  <InputSerialization>
    <Compression>None</Compression>
    <CSV>
      <FileHeaderInfo>Ignore</FileHeaderInfo>
      <RecordDelimiter>Cg==</RecordDelimiter>
      <FieldDelimiter>LA==</FieldDelimiter>
      <QuoteCharacter>Ig==</QuoteCharacter>
      <CommentCharacter>lw==</CommentCharacter/>
    </CSV>
  </InputSerialization>
  <OutputSerialization>
    <CSV>
      <RecordDelimiter>Cg==</RecordDelimiter>
      <FieldDelimiter>LA==</FieldDelimiter>
      <QuoteCharacter>Ig==</QuoteCharacter>
    </CSV>
    <KeepAllColumns>>false</KeepAllColumns>
    <OutputRawData>>false</OutputRawData>
  </OutputSerialization>
</SelectRequest>
```

- 样例请求 (JSON)

```
POST /oss-select/sample_json.json?x-oss-process=json%2Fselect HTTP/1.1
Host: host name
Accept-Encoding: identity
User-Agent: aliyun-sdk-python/2.6.0(Darwin/16.7.0/x86_64;3.5.4)
Accept: */*
Connection: keep-alive
date: Mon, 10 Dec 2018 18:28:11 GMT
authorization: OSS AccessKeySignature
Content-Length: 317
<SelectRequest>
  <Expression>c2VsZWN0ICogZnJvbSBvc3NvYmplY3Qub2JqZWN0c1sqXSB3aGVyZSBwYXJ0eSA9ICdEZW1
vY3JhdCc=
  </Expression>
  <InputSerialization>
    <JSON>
      <Type>DOCUMENT</Type>
    </JSON>
  </InputSerialization>
  <OutputSerialization>
    <JSON>
      <RecordDelimiter>LA==</RecordDelimiter>
    </JSON>
  </OutputSerialization>
  <Options />
</SelectRequest>
```

SQL语句正则表达式

SQL语句正则表达式为 `SELECT select-list from table where_opt limit_opt`。

 **说明** 其中，SELECT、OSSOBJECT以及WHERE为关键字不得更改。

select_list: column name
| column index (例如 _1, _2. 仅Csv文件有column index)
| json path (例如s.contacts.firstname 仅应用在JSON文件)
| function(column index | column name)
| function(json_path) (仅应用在JSON文件)
| select_list AS alias

 **说明** 支持的function为AVG、SUM、MAX、MIN、COUNT、CAST（类型转换函数）。其中COUNT后面只能用*。

table: OSSOBJECT

| OSSOBJECT json_path (仅应用于JSON文件)

对于CSV文件, table必须为OSSOBJECT。对于JSON文件 (包括DOCUMENT或者LINES), table可以在OSSOBJECT后指定json_path。

json_path: ['string'] (string若没有空格或者*, 则引号可以去掉; 等价于.'string')

| [n] (应用在数组元素中, 表示第n个元素, 从0开始)

| [*] (应用在数组或者对象中, 表示任意子元素)

| '.string' (string若没有空格或者*, 则引号可以去掉)

| json_path jsonpath (json_path可以连接, 例如 [n].property1.attributes[*])

Where_opt:

| WHERE expr

expr:

| literal value

| column name

| column index

| json path (仅应用于JSON文件)

| expr op expr

| expr OR expr

| expr AND expr

| expr IS NULL

| expr IS NOT NULL

| (column name | column index | json path) IN (value1, value2,...)

| (column name | column index | json path) NOT in (value1, value2,...)

| (column name | column index | json path) between value1 and value2

| NOT (expr)

| expr op expr

| (expr)

| cast (column index | column name | json path | literal as INT|DOUBLE|)

- op: 包括 > 、 < 、 >= 、 <= 、 != 、 = 、 , 、 LIKE 、 + 、 - 、 * 、 / 、 % 以及字符串连接 || 。
- cast: 对于同一个column, 只能cast成一种类型。
- 聚合和Limit的混用: Select avg(cast(_1 as int)) from ossobject limit 100 , 其含义是指在前100行中计算第一列的AVG值。与MySQL不同的是在Select Object中聚合永远只返回一行数据, 因而对聚合来说无需限制其输出规模。Select Object中limit将先于聚合函数执行。

SQL语句限制

SQL语句限制如下:

- 目前仅仅支持UTF-8编码的文本文件以及GZIP压缩过的UTF-8文本。GZIP文件不支持deflate格式。
- 仅支持单文件查询, 不支持join、order by、group by、having。
- Where语句里不能包含聚合条件 (e.g. where max(cast(age as int)) > 100这个是不允许的)。
- 支持的最大的列数是1000, SQL中最大列名长度不能超过1024字节。
- 在LIKE语句中, 支持最多5个%通配符。*和%是等价的, 表示0或多个任意字符。LIKE支持Escape关键字, 用于将Pattern中的%*?特殊字符Escape成普通字符串。
- 在IN语句中, 最多支持1024个常量项。
- Select后的Projection可以是列名, CSV列索引 (_1, _2等), 或者是聚合函数, 或者是CAST函数; 不支持其他表达式。例如select _1 + _2 from ossobject是不允许的。

- 支持的CSV单行及单列的最大字符数均为256KB。
- 支持的 from 后json path指定的JSON最大节点为512KB，最大深度为10层，其数组内元素的个数最多为5000。select 以及where后的字段必须是来自from后json path对应的节点。
- JSON文件的SQL中，select 或者where条件的表达式中不能有[*]数组通配符，数组通配符只能出现在from 后的json path中。（例如select s.contacts[*] from ossobject s是不允许的，但是select * from ossobject.contacts[*]是可以的）。
- SQL最大长度为16KB，where后面表达式个数最多20个，表达式深度最多10层，聚合操作最多100个。

数据的容错处理

以下列举了CSV的某些行缺失了某些列、JSON缺失某些Key、CSV的某些列类型不匹配SQL，以及JSON中某些Key类型不匹配SQL等常见的数据容错处理方法。

- CSV的某些行缺失了某些列

当SkipPartialDataRecord没有被指定或者其值为False时，OSS会把缺失的列都当做null来计算SQL中的表达式。

当SkipPartialDataRecord指定为True时，则OSS会忽略该行数据。此时，如果MaxSkippedRecordsAllowed未指定或者指定的值小于忽略的行数，则OSS会通过http status code或者End Frame的形式返回400错误。

当SQL为 `select _1,_3 from ossobject`，而CSV的某一行指定为“张小,阿里巴巴”。

- 当SkipPartialDataRecord指定为False时，返回内容为“张小,\n”
- 当SkipPartialDataRecord指定为True时，该行被跳过。

- JSON缺失某些Key

和CSV一样，JSON的某些对象可能未包括SQL中指定的Key。

- 当SkipPartialDataRecord没有被指定或者其值为False时，此时OSS会把缺失的Key当做Null来计算SQL中的表达式。
- 如果SkipPartialDataRecord为true时，OSS则会忽略该JSON节点的数据。此时，如果MaxSkippedRecordsAllowed未指定或者指定的值小于忽略的行数，则OSS会通过http status code或者End Frame返回400错误。

例如SQL为 `select s.firstName, s.lastName, s.age from ossobject.contacts[*] s`，某个JSON节点为 `{ "firstName": "小", "lastName": "张" }`。

- 当SkipPartialDataRecord为False或未指定时，该节点的返回值为 `{ "firstName": "小", "lastName": "张" }`。
- 当SkipPartialDataRecord为True时则跳过该行。

说明 对于JSON返回数据中Key的处理，SelectJson的输出只能为JSON LINES。输出结果中的Key值按以下规则确定。

- 当SQL为 `select * from ossobject...` 时，如果*返回的内容是一个JSON对象（{...}），则原样返回；如果*不是一个JSON对象（例如是字符串或者数组），则会使用一个DummyKey_1来表示。

当数据为{ "Age" :5} `select * from ossobject.Age s where s = 5`，此时*对应的内容5不是一个JSON对象，故返回的内容为{ "_1" :5}。假设SQL为 `select * from ossobject s where s.Age = 5`，此时*对应的内容是{ "Age" :5}，故原样返回。

- 当SQL没有使用select *而是指定了列时，则返回的内容格式为 { "{列1}" :值, "{列2}" :值...}。其中{列n}的生成方法如下：
 - 当select中指定了该列的Alias时用Alias。
 - 如果该列是一个JSON对象的Key时，用该Key作为输出的Key值。
 - 如果该列是JSON数组的某个元素时或者是聚合函数，用该列在输出中的序号（从1开始）加上前缀_作为输出值的key。

当数据为{ "contacts" :{ "Age" :35, "Children" :["child1", "child2", "child3"]}}:

- 当SQL为 `select s.contacts.Age, s.contacts.Children[0] from ossobject s`，由于Age是输入JSON对象的Key，但Children[0]是数组Children的第一个元素，在输出中是第二列，故为 "_2" : "child1"，则输出为 { "Age" :35, "_2" : "child1" }。
- 当SQL为 `select max(cast(s.Age as int)) from ossobject.contacts s`，由于该列是聚合函数，故而用该列在输出中的序号加上前缀_1表示Key，则输出为 { "_1" :35}。
- 如果指定了Alias `select s.contacts.Age, s.contacts.Children[0] as firstChild from ossobject`，则输出为 { "Age" :35, "firstChild" : "child1" }。

- JSON文件和SQL中Key的匹配是大小写敏感的，例如select s.Age 和select s.age是不同的。

● CSV的某些列类型不匹配SQL

当CSV中某些行的数据类型和SQL中指定的数据类型不匹配时，该行数据会被忽略。和上面情况类似，其行为受到MaxSkippedRecordsAllowed的影响。当忽略行数超过上限时，OSS将处理中转并返回400。

当SQL为 `select _1, _3 from ossobject where _3 > 5`，某CSV行为 张小, 阿里巴巴, 待入职 时，由于第三列不是整数类型，则该行被跳过。

● JSON中某些Key类型不匹配SQL

当SQL为 `select s.name from ossobject s where s.aliren_age > 5`，JSON节点是 {"Name": "张小", "aliren_age": "待入职"}，则该节点被跳过。

CreateSelectObjectMeta

CreateSelectObjectMeta API用于获取目标文件总的行数，总的列数（对于CSV文件），以及Splits个数。如果该信息不存在，则会扫描整个文件，分析并记录下CSV文件的上述信息。重复调用该API则会保存上述信息而不必重新扫描整个文件。

说明

- CreateSelectObjectMeta操作要求您对该Object有写权限。
- 如果该API执行正确，返回200。如果目标文件不是合法CSV或者JSON LINES文件，或者指定的CSV分隔符和目标CSV不匹配，则返回400。

- 请求语法
 - 请求语法（CSV）

```
POST /samplecsv?x-oss-process=csv/meta
<CsvMetaRequest>
  <InputSerialization>
    <CompressionType>None</CompressionType>
    <CSV>
      <RecordDelimiter>base64 encode的字符</RecordDelimiter>
      <FieldDelimiter>base64 encode的字符</FieldDelimiter>
      <QuoteCharacter>base64 encode的字符</QuoteCharacter>
    </CSV>
  </InputSerialization>
  <OverwriteIfExists>>false|true</OverwriteIfExists>
</CsvMetaRequest>
```

- 请求语法（JSON）

```
POST /samplecsv?x-oss-process=json/meta
<JsonMetaRequest>
  <InputSerialization>
    <CompressionType>None</CompressionType>
    <JSON>
      <Type>LINES</Type>
    </JSON>
  </InputSerialization>
  <OverwriteIfExists>>false|true</OverwriteIfExists>
</JsonMetaRequest>
```

- 请求元素

名称	类型	描述
CsvMetaRequest	容器	保存创建Select csv Meta请求的容器。 子节点: InputSerialization 父节点: None
JsonMetaRequest	容器	保存创建 Select json meta请求的容器。 子节点: InputSerialization 父节点: None
InputSerialization	容器	输入序列化参数（可选） 子节点: CompressionType、CSV、JSON 父节点: CsvMetaRequest、JsonMetaRequest

名称	类型	描述
OverwriteIfExists	bool	重新计算SelectMeta, 覆盖已有数据。(可选, 默认是false, 即如果Select Meta已存在则直接返回) 子节点: None 父节点: CsvMetaRequest、JsonMetaRequest
CompressionType	枚举	指定文件压缩类型(可选)。目前不支持任何压缩, 故只能为None。 子节点: None 父节点: InputSerialization
RecordDelimiter	字符串	指定CSV换行符, 以Base64编码。默认值为'\n' (可选)。未编码前的值最多为两个字符, 以字符的ANSI值表示, 例如在Java里用\n表示换行。 子节点: None 父节点: CSV
FieldDelimiter	字符串	指定CSV列分隔符, 以Base64编码。默认值为 , (可选) 未编码前的值必须为一个字符, 以字符的ANSI值表示, 例如Java里用 , 表示逗号。 子节点: None 父节点: CSV (输入, 输出)
QuoteCharacter	字符串	指定CSV的引号字符, 以Base64编码。默认值为\" (可选)。 在CSV中引号内的换行符, 列分隔符将被视作普通字符。未编码前的值必须为一个字符, 以字符的ANSI值表示, 例如Java里用\"表示引号。 子节点: None 父节点: CSV (输入)
CSV	容器	指定CSV输入格式。 子节点: RecordDelimiter, FieldDelimiter, QuoteCharacter 父节点: InputSerialization
JSON	容器	指定JSON输入格式。 子节点: Type 父节点: InputSerialization

名称	类型	描述
Type	枚举	指定JSON类型。 合法值：LINES

- 返回Body

和SelectObject 类似，CreateSelectObjectMeta API也以Frame的形式返回。

名称	Frame-Type值	Payload格式	描述
Meta End Frame (Csv)	8388614	offset total scanned bytes status splits count rows count columns count error message <-8 bytes><-----8 bytes-----> ><--4bytes><--4 bytes--><-8 bytes><--4 bytes--> <variable size>	用来汇报CreateSelectObjectMeta API最终的状态。 ◦ offset：8位整数，扫描结束时的文件偏移。 ◦ total scanned bytes：8位整数，最终扫描过的数据大小。 ◦ status：4位整数，最终的状态。 ◦ splits_count：4位整数，总split个数。 ◦ rows_count：8位整数，总行数。 ◦ cols_count：4位整数，总列数。 ◦ error_message：详细的错误信息。若无错误，则error_message为空。
Meta End Frame (Json)	8388615	offset total scanned bytes status splits count rows count error message <-8 bytes><-----8 bytes-----> ><--4bytes><--4 bytes--><-8 bytes><variable size>	用来汇报CreateSelectObjectMeta API最终的状态。 ◦ offset：8位整数，扫描结束时的文件偏移。 ◦ total scanned bytes：8位整数，最终扫描过的数据大小。 ◦ status：4位整数，最终的状态。 ◦ splits_count：4位整数，总split个数。 ◦ rows_count：8位整数，总行数。 ◦ error_message：详细的错误信息。若无错误，则error_message为空。

- 样例请求

- 样例请求（CSV）

```
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 AccessKeySignature
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>lg==</QuoteCharacter>
    </CSV>
  </InputSerialization>
  <OverwriteIfExists>false</OverwriteIfExists>
</CsvMetaRequest>
```

- 样例请求（JSON）

```
POST /oss-select/sample.json?x-oss-process=json%2Fmeta HTTP/1.1
Date: Fri, 25 May 2018 23:06:41 GMT
Content-Type:
Authorization: OSS AccessKeySignature
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"?>
<JsonMetaRequest>
  <InputSerialization>
    <JSON>
      <Type>LINES</Type>
    </JSON>
  </InputSerialization>
  <OverwriteIfExists>false</OverwriteIfExists>
</JsonMetaRequest>
```

支持的时间格式

自动识别是指无需指定日期格式就能把符合以下格式之一的字符串转成time stamp类型。例如cast-('20121201' as time stamp)会被解析成2012年12月1日。

以下格式为自动识别的格式：

格式	说明
YYYYMMDD	年月日

格式	说明
YYYY/MM/DD	年/月/日
DD/MM/YYYY/	日/月/年
YYYY-MM-DD	年-月-日
DD-MM-YY	日-月-年
DD.MM.YY	日.月.年
HH:MM:SS.mss	小时:分钟:秒.毫秒
HH:MM:SS	小时:分钟:秒
HH MM SS mss	小时 分钟 秒 毫秒
HH.MM.SS.mss	小时.分钟.秒.毫秒
HHMM	小时分钟
HHMMSSmss	小时分钟秒毫秒
YYYYMMDD HH:MM:SS.mss	年月日 小时:分钟:秒.毫秒
YYYY/MM/DD HH:MM:SS.mss	年/月/日 小时:分钟:秒.毫秒
DD/MM/YYYY HH:MM:SS.mss	日/月/年 小时:分钟:秒.毫秒
YYYYMMDD HH:MM:SS	年月日 小时:分钟:秒
YYYY/MM/DD HH:MM:SS	年/月/日 小时:分钟:秒
DD/MM/YYYY HH:MM:SS	日/月/年 小时:分钟:秒
YYYY-MM-DD HH:MM:SS.mss	年-月-日 小时:分钟:秒.毫秒
DD-MM-YYYY HH:MM:SS.mss	日-月-年 小时:分钟:秒.毫秒
YYYY-MM-DD HH:MM:SS	年-月-日 小时:分钟:秒
YYYYMMDDT HH:MM:SS	年月日T小时:分钟:秒
YYYYMMDDT HH:MM:SS.mss	年月日T小时:分钟:秒.毫秒
DD-MM-YYYYT HH:MM:SS.mss	日-月-年T小时:分钟:秒.毫秒
DD-MM-YYYYT HH:MM:SS	日-月-年T小时:分钟:秒
YYYYMMDDT HHMM	年月日T小时分钟
YYYYMMDDT HHMMSS	年月日T小时分钟秒

格式	说明
YYYYMMDDTHHMMSSMSS	年月日T小时分钟秒毫秒
ISO8601-0	年-月-日T小时:分钟+小时:分钟, 或者年-月-日T小时:分钟-小时:分钟 这里+表示时区在标准时间UTC前面, -表示在后面。注意这个格式可用ISO8601-0来表示
ISO8601-1	年-月-日T小时:分钟:秒+小时:分钟, 或者年-月-日T小时:分钟-小时:分钟 这里+表示时区在标准时间UTC前面, -表示在后面。这个格式可用ISO8601-1来表示
CommonLog	例如28/Feb/2017:12:30:51 +0700
RFC822	例如Tue, 28 Feb 2017 12:30:51 GMT
?D/?M/YY	日/月/年, 此处日月均可用1位或者2位表示
?D/?M/YY ?H:?M	日/月/年/小时:分钟, 此处日月分钟小时均可用1位或者2位表示
?D/?M/YY ?H:?M:?S	日/月/年/小时:分钟:秒, 此处日月秒分钟小时均可用1位或者2位表示

由于以下格式会引起歧义, 因此需要指定格式。如cast('20121201' as timestamp format 'YYYYDDMM')会将20121201解析成2012年1月12日。

格式	说明
YYYYDDMM	年日月
YYYY/DD/MM	年/日/月
MM/DD/YYYY	月/日/年
YYYY-DD-MM	年-日-月
MM-DD-YYYY	月-日-年
MM.DD.YYYY	月.日.年

ErrorCode

SelectObject通过以下两种形式返回ErrorCode。

- 一种是和其他OSS请求一样, 在HTTP响应Headers里返回http status code, 并在body里返回Error信息。此类错误通常是存在明显的输入错误(例如输入的SQL不正确), 或者明显的数据库错误。
- 另一种ErrorCode是在响应Body的End Frame中里返回。这种情况通常是SelectObject处理数据的过程中, 检测到数据不正确或者和SQL不匹配(例如在某一行里原本是整数列现在是一个字符串, 且SQL里指

定了该列是整数），此时往往部分数据已处理完毕并已发送到客户端，且http status code仍然是206。
在下表的ErrorCode中，有些ErrorCode（例如InvalidCSVLine）可能以http status code的形式返回，也可能在End Frame中的status code返回，以何种形式返回ErrorCode取决于发生错误的CSV行在文件的具体位置。

ErrorCode	描述	HTTP Status Code	Http Status Code in End Frame
InvalidSqlParameter	非法的SQL参数。 请求中SQL为空或者SQL长度超过最大值或者SQL不是以base64编码。	400	无
InvalidInputFieldDelimiter	非法的CSV列分隔符（输入）。 该参数不是以Base64编码或者解码后的长度大于1。	400	无
InvalidInputRecordDelimiter	非法的CSV行分隔符（输入）。该参数不是以Base64编码或者解码后长度大于2。	400	无
InvalidInputQuote	非法的CSV Quote字符（输入）。该参数不是以Base64编码或者解码后长度超过1。	400	无
InvalidOutputFieldDelimiter	非法的CSV列分隔符（输出）。该参数不是以Base64编码或者解码后的长度大于1。	400	无
InvalidOutputRecordDelimiter	非法的CSV行分隔符（输出）。该参数不是以Base64编码或者解码后长度大于2。	400	无
UnsupportedCompressionFormat	非法的Compression参数。该参数值不是NONE或者GZIP（此处大小写无关）。	400	无
InvalidCommentCharacter	非法的CSV注释字符。该参数值不是以Base64编码或者解码后的值超过1。	400	无
InvalidRange	非法的Range参数。该参数不是以line-range=或者split-range=作为前缀，或者range值本身不符合HTTP range规范。	400	无
DecompressFailure	Compression参数是GZIP且解压失败。	400	无
InvalidMaxSkippedRecordsAllowed	MaxSkippedRecordsAllowed参数不是整数。	400	无
SelectCsvMetaUnavailable	指定了Range参数，但目标对象没有CSV meta时，此时应该先调用Create Select object MetaAPI。	400	无
InvalidTextEncoding	对象不是以UTF-8编码。	400	无
InvalidOSSSelectParameters	同时指定了EnablePayloadCrc为True以及OutputRawData为True时，两个参数冲突，返回此错误。	400	无

ErrorCode	描述	HTTP Status Code	Http Status Code in End Frame
InternalError	OSS系统出现错误。	500或者206	无或者500
SqlSyntaxError	Base64解码后的SQL语法错误。	400	无
SqlExceedsMaxInCount	SQL中IN语句包含的值数目超过1024。	400	无
SqlExceedsMaxColumnNameLength	SQL中列名超过最大长度时1024。	400	无
SqlInvalidColumnIndex	SQL中列Index小于1或者大于1000。	400	无
SqlAggregationOnNonNumericType	SQL中聚合函数应用在非数值列中。	400	无
SqlInvalidAggregationOnTimestamp	SQL中SUM/AVG聚合应用在日期类型列。	400	无
SqlValueTypeOfInMustBeSame	SQL中IN语句包含不同类型的值。	400	无
SqlInvalidEscapeChar	SQL LIKE语句中Escape的字符为?或%或*。	400	无
SqlOnlyOneEscapeCharIs Allowed	SQL LIKE语句中Escape字符长度大于1。	400	无
SqlNoCharAfterEscapeChar	SQL LIKE语句中Escape字符后面没有被Escape字符。	400	无
SqlInvalidLimitValue	SQL Limit语句后的数字小于1。	400	无
SqlExceedsMaxWildcardCount	SQL LIKE语句中通配符*或者%的个数超过最大值。	400	无
SqlExceedsMaxConditionCount	SQL Where语句中条件表达式个数超过最大值。	400	无
SqlExceedsMaxConditionDepth	SQL Where语句中条件树的深度超过最大值。	400	无
SqlOneColumnCastToDifferentTypes	SQL语句中同一列被Cast成不同类型。	400	无
SqlOperationAppliedToDifferentTypes	SQL条件语句中一个操作符应用在两个不同类型的对象。 例如_col1 > 3如果其中col1是字符串时就会报这个错误。	400	无

ErrorCode	描述	HTTP Status Code	Http Status Code in End Frame
SqlInvalidColumnName	SQL语句中使用列名且该列名并不在CSV文件Header中出现。	400	无
SqlNotSupportedTimestampFormat	SQL CAST 语句中指定了时间戳格式，且该格式不支持。	400	无
SqlNotMatchTimestampFormat	SQL CAST 语句中指定了时间戳格式，但该格式和提供的时间戳字符串不匹配。	400	无
SqlInvalidTimestampValue	SQL CAST 语句中未指定时间戳格式，且提供的时间戳字符串无法cast成时间戳类型。	400	无
SqlInvalidLikeOperand	SQL LIKE语句中左边不是列名或者列Index或者左边的列不是字符串类型，或者右边字符串。	400	无
SqlInvalidMixOfAggregationAndColumn	SQL Select语句中同时包含聚合操作以及非聚合操作的列名、列Index。	400	无
SqlExceedsMaxAggregationCount	SQL Select语句中包含的聚合操作超过最大值。	400	无
SqlInvalidMixOfStarAndColumn	SQL Select语句中同时包含*以及列名、列Index。	400	无
SqlInvalidKeepAllColumnsWithAggregation	SQL包含聚合操作，同时KeepAllColumns参数为True。	400	无
SqlInvalidKeepAllColumnsWithDuplicateColumn	SQL包含重复的列名、列索引，同时KeepAllColumns参数为True。	400	无
SqlInvalidSqlAfterAnalysis	SQL过分复杂导致分析后无法支持。	400	无
InvalidArithmeticOperand	SQL中算术操作应用在非数字类型的常量或者列。	400	无
SqlInvalidAndOperand	SQL中AND操作连接的两个表达式不是bool类型。	400	无
SqlInvalidOrOperand	SQL中OR操作连接的两个表达式不是bool类型。	400	无
SqlInvalidNotOperand	SQL中NOT操作应用在非bool类型的表达式。	400	无
SqlInvalidIsNullOperand	SQL中IS NULL应用在常量上。	400	无
SqlComparerOperandTypeMismatch	SQL中比较操作应用两个不同类型的对象。	400	无

ErrorCode	描述	HTTP Status Code	Http Status Code in End Frame
SqlInvalidConcatOperand	SQL中字符串连接操作（ ）连接两个常量。	400	无
SqlUnsupportedSql	SQL过分复杂从而生成的SQL计划超过最大值。	400	无
HeaderInfoExceedsMaxSize	SQL指定了输出Header信息，且Header信息超过最大允许的长度。	400	无
OutputExceedsMaxSize	SQL的输出有放大导致输出结果中单行超过最大长度。	400	无
InvalidCsvLine	CSV行是非法时（包括长度超过最大允许的长度），或者被跳过的行数超过MaxSkippedRecordsAllowed。	206或者400	400或者无
NegativeRowIndex	SQL中作为数组Index的值为负数。	400	无
ExceedsMaxNestedColumnDepth	SQL中的JSON属性嵌套层数超过最大值。	400	无
NestedColumnNotSupportInCsv	嵌套的属性（含有数组[]或者.）不支持在CSV的SQL中。	400	无
TableRootNodeOnlySupportInJson	只有JSON文件支持在From ossobject后指定根节点路径。	400	无
JsonNodeExceedsMaxSize	JSON文件的SQL的根节点大小超过允许最大值。	400或者206	无或者400
InvalidJsonData	非法的JSON数据（格式不正确）。	400或者206	无或者400
ExceedsMaxJsonArraySize	JSON文件的SQL的根节点中的某个数组的元素个数超过最大值。	400或者206	无或者400
WildcardNotAllowed	JSON文件的SQL中wild card *不能出现select或者where的JSON属性中。例如 <code>select s.a.b[*] from ossobject where a.c[*] > 0</code> 是不支持的。	400	无
JsonNodeExceedsMaxDepth	JSON文件的SQL根节点深度超过最大值。	400或者206	无或者400

SDK

- [Java](#)
- [Python](#)

7.2. 分片上传（MultipartUpload）

7.2.1. InitiateMultipartUpload

使用Multipart Upload模式传输数据前，您必须先调用InitiateMultipartUpload接口来通知OSS初始化一个Multipart Upload事件。

注意事项

- 调用接口会返回一个OSS服务器创建的全局唯一的Upload ID，用于标识本次Multipart Upload事件。您可以根据这个ID来发起相关的操作，例如中止Multipart Upload、查询Multipart Upload等。
- 初始化Multipart Upload请求，并不影响已存在的同名Object。
- 该操作计算认证签名时，需要添加 `?uploads` 到 `CanonicalizedResource` 中。

请求语法

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

请求参数

调用此接口时，可以通过encoding-type对返回结果中的Key进行编码。

名称	类型	描述
encoding-type	字符串	指定对返回的Key进行编码，目前支持URL编码。Key使用UTF-8字符，但XML 1.0标准不支持解析一些控制字符，例如ASCII值从0到10的字符。对于Key中包含XML 1.0标准不支持的控制字符，可以通过指定encoding-type对返回的Key进行编码。 默认值：无 可选值： <i>url</i>

请求头

初始化Multipart Upload请求时，支持标准的HTTP请求头Cache-Control、Content-Disposition、Content-Encoding、Content-Type、Expires以及以 `x-oss-meta-` 开头的用户自定义Header。更多信息，请参见PutObject。

名称	类型	描述
Cache-Control	字符串	指定该Object被下载时的网页的缓存行为。更多信息，请参见RFC 2616。 默认值：无
Content-Disposition	字符串	指定该Object被下载时的名称。更多信息，请参见RFC 2616。 默认值：无

名称	类型	描述
Content-Encoding	字符串	指定该Object被下载时的内容编码格式。更多信息，请参见 RFC 2616 。 默认值：无
Expires	整数	过期时间，单位为毫秒。更多信息，请参见 RFC 2616 。 默认值：无
x-oss-forbid-overwrite	字符串	指定InitiateMultipartUpload操作时是否覆盖同名Object。当目标Bucket处于已开启或已暂停的版本控制状态时，x-oss-forbid-overwrite请求Header设置无效，即允许覆盖同名Object。 - 不指定x-oss-forbid-overwrite或者指定x-oss-forbid-overwrite为<code>false</code>时，表示允许覆盖同名Object。 - 指定x-oss-forbid-overwrite为<code>true</code>时，表示禁止覆盖同名Object。 设置x-oss-forbid-overwrite请求Header会导致QPS处理性能下降，如果您有大量的操作需要使用x-oss-forbid-overwrite请求Header（QPS>1000），请联系技术支持，避免影响您的业务。
x-oss-server-side-encryption	字符串	指定上传该Object的每个part时使用的服务器端加密方式。 取值：AES256、KMS或SM4  注意 使用KMS加密算法前，您需要先开通密钥管理服务KMS。 指定此参数后，在响应头中会返回此参数，OSS会对上传的每个part进行加密编码存储。当下载该Object时，响应头中会包含x-oss-server-side-encryption，且该值会被设置成此Object的加密算法。
x-oss-server-side-data-encryption	字符串	指定Object的加密算法。若未指定此选项，表明Object使用AES256加密算法。此选项仅当x-oss-server-side-encryption为KMS时有效。 取值：SM4
x-oss-server-side-encryption-key-id	字符串	表示KMS托管的用户主密钥。 该参数在x-oss-server-side-encryption值为KMS时有效。

名称	类型	描述
x-oss-storage-class	字符串	指定Object的存储类型。 对于任意存储类型Bucket，如果上传Object时指定该值，则此次上传的Object将存储为指定的类型。例如在IA类型的Bucket中上传Object时，如果指定x-oss-storage-class为Standard，则该Object直接存储为Standard类型。 取值： - Standard: 标准存储 - IA: 低频访问 - Archive: 归档存储 - ColdArchive: 冷归档存储 关于存储类型的更多信息，请参见存储类型介绍。
x-oss-tagging	字符串	指定Object的标签，可同时设置多个标签，例如：TagA=A&TagB=B  说明 Key和Value需要先进行URL编码，如果某项没有 = ，则看作Value为空字符串。

响应元素

服务器收到初始化Multipart Upload请求后，会返回一个XML格式的请求体。该请求体内包含Bucket、Key和UploadID三个元素。

名称	类型	描述
Bucket	字符串	初始化一个Multipart Upload事件的Bucket名称。 父节点：InitiateMultipartUploadResult
InitiateMultipartUploadResult	容器	保存Initiate Multipart Upload请求结果的容器。 子节点：Bucket, Key, UploadId, EncodingType 父节点：None
Key	字符串	初始化一个Multipart Upload事件的Object名称。 父节点：InitiateMultipartUploadResult
UploadId	字符串	唯一标识此次Multipart Upload事件的ID，用于后续调用UploadPart和CompleteMultipartUpload接口。 父节点：InitiateMultipartUploadResult

名称	类型	描述
EncodingType	字符串	指明返回结果中编码使用的类型。如果请求的参数中指定了encoding-type，那返回的结果会对Key进行编码。 父节点：InitiateMultipartUploadResult

示例

- 请求示例

```
POST /multipart.data?uploads HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
x-oss-storage-class: Archive
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:/cluRFtRwMTZpC2hTj4F67AG****
```

- 返回示例

```
HTTP/1.1 200 OK
Content-Length: 230
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 42c25703-7503-fbd8-670a-bda01eae****
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> oss-example</Bucket>
  <Key>multipart.data</Key>
  <UploadId>0004B9894A22E5B1888A1E29F823****</UploadId>
</InitiateMultipartUploadResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Browser.js](#)

错误码

错误码	HTTP状态码	描述
InvalidEncryptionAlgorithmError	400	指定OSS服务器端除AES256、KMS或SM4以外的加密方式。
InvalidArgument	400	在上传每个part时继续添加x-oss-server-side-encryption请求头。
InvalidArgument	400	指定存储类型的值不合法。
KmsServiceNotEnabled	403	使用KMS加密算法时，没有在控制台开通密钥管理服务KMS。
FileAlreadyExists	409	当请求的Header中携带x-oss-forbid-overwrite=true时，表示禁止覆盖同名文件。如果已存在同名文件，则返回该错误。

7.2.2. UploadPart

初始化一个Multipart Upload后，调用UploadPart接口根据指定的Object名和uploadId来分块（Part）上传数据。

注意事项

- 调用UploadPart接口上传Part数据前，必须先调用InitiateMultipartUpload接口来获取OSS服务器生成的uploadId。
- 如果使用同一个partNumber上传了新的数据，则OSS上已有的partNumber对应的Part数据将被覆盖。
- OSS会将服务器端收到Part数据的MD5值放在ETag头返回给用户。
- 如果调用InitiateMultipartUpload接口时，指定了x-oss-server-side-encryption请求头，则会对上传的Part进行加密编码，并在UploadPart响应头中返回x-oss-server-side-encryption头，其值表明该Part的服务器端加密算法。更多信息，请参见InitiateMultipartUpload。

请求语法

```
PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
```

请求元素

名称	类型	是否必选	描述
----	----	------	----

名称	类型	是否必选	描述
partNumber	正整数	是	每一个上传的Part都有一个标识它的号码（partNumber）。 取值：1~10000 单个Part的大小限制为100 KB~5 GB。  说明 MultipartUpload事件中除最后一个Part以外，其他Part的大小都要大于或等于100 KB。因不确定是否为最后一个Part，UploadPart接口并不会立即校验上传Part的大小，只有当CompleteMultipartUpload时才会校验。
uploadId	字符串	是	uploadId用于唯一标识上传的Part属于哪个Object。

示例

请求示例

```
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 qn6qrrqxo2oawuk53otf****:J/lICfXEvPmmSW86bBAfMmUm****
[6291456 bytes data]
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
ETag: "7265F4D211B56873A381D321F586****"
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0****
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [.NET](#)

- [Node.js](#)
- [Android](#)
- [iOS](#)

错误码

错误码	HTTP状态码	描述
NoSuchUpload	404	分片上传ID不存在。
InvalidArgument	400	超出partNumber范围（1~10000）。
InvalidDigest	400	为了保证数据在网络传输过程中不出现错误，用户发送请求时可以携带Content-MD5，OSS计算上传数据的MD5与用户上传的MD5值不一致。

7.2.3. UploadPartCopy

通过在UploadPart请求的基础上增加一个请求头x-oss-copy-source来调用UploadPartCopy接口，实现从一个已存在的Object中拷贝数据来上传一个Part。

注意事项

当拷贝一个大于1 GB的文件时，必须使用UploadPartCopy的方式进行拷贝。如果想通过单个操作拷贝小于1 GB的文件，请参见[CopyObject](#)。

使用UploadPartCopy接口时，有如下注意事项：

- 不允许拷贝以AppendObject方式上传的Object。
- 执行UploadPartCopy的源Bucket地址和目标Bucket地址必须是同一个Region。
- 调用该接口上传Part数据前，必须先调用InitiateMultipartUpload接口来获取一个OSS服务器颁发的Upload ID。
- 若调用InitiateMultipartUpload接口时，指定了x-oss-server-side-encryption请求头，则会对上传的Part进行加密编码，并在UploadPart响应头中返回x-oss-server-side-encryption头，其值表明该Part的服务器端加密算法，详情请参见[InitiateMultipartUpload](#)。
- MultipartUpload要求除最后一个Part以外，其他的Part大小都要大于100 KB。因不确定是否为最后一个Part，UploadPart接口并不会立即校验上传Part的大小，只有当CompleteMultipartUpload的时候才会校验。

版本控制

UploadPartCopy默认从一个已存在的Object的当前版本中拷贝数据来上传一个Part。允许通过在请求头x-oss-copy-source中附带versionId的子条件，实现从Object的指定版本进行拷贝，例如x-oss-copy-source: /SourceBucketName/SourceObjectName?versionId=111111。

 **说明** SourceObjectName要进行URL编码。响应中将会返回被拷贝Object的versionId: x-oss-copy-source-version-id。

如果未指定versionId且拷贝Object的当前版本为删除标记（Delete Marker），OSS将返回404 Not Found。通过指定versionId来拷贝删除标记时，OSS将返回400 Bad Request。

请求语法

```
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
```

请求头

除了通用的请求头，UploadPartCopy请求中通过下述请求头指定拷贝的源Object地址和拷贝的范围。

名称	类型	描述
x-oss-copy-source	字符串	拷贝源地址（必须有可读权限）。 默认值：无
x-oss-copy-source-range	整型	源Object的拷贝范围。例如设置 bytes=0~9，表示拷贝0到9这10个字符。 默认值：无 - 不指定该请求头时，表示拷贝整个源Object。 - 当指定该请求头时，则返回消息中会包含整个文件的长度和此次拷贝的范围，例如：Content-Range: bytes 0~9/44，表示整个文件长度为44，此次拷贝的范围为0~9。 - 当指定的范围不符合规范时，则拷贝整个源Object，并且不在结果中提及Content-Range。

下述请求Header作用于x-oss-copy-source指定的源Object。

名称	类型	描述
x-oss-copy-source-if-match	字符串	如果源Object的ETAG值和用户提供的ETAG相等，则执行拷贝操作；否则返回412 HTTP错误码（预处理失败）。 默认值：无

名称	类型	描述
x-oss-copy-source-if-none-match	字符串	如果传入的ETag值和Object的ETag不匹配，则正常传输文件，并返回200 OK；否则返回304 Not Modified。 默认值：无
x-oss-copy-source-if-unmodified-since	字符串	如果传入参数中的时间等于或者晚于文件实际修改时间，则正常传输文件，并返回200 OK；否则返回412 precondition failed错误。 默认值：无
x-oss-copy-source-if-modified-since	字符串	如果指定的时间早于实际修改时间，则正常传输文件，并返回200 OK；否则返回304 not modified。 默认值：无 时间格式：GMT时间，例如Fri, 13 Nov 2015 14:47:53 GMT

示例

- 请求示例

```
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 qn6qrrqxo2oawuk53otf****:J/lICfXEvPmmSW86bBAfMmUm****
x-oss-copy-source: /oss-example/src-object
x-oss-copy-source-range: bytes=100-6291756
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0****
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>"5B3C1A2E053D763E1B002CC607C5****"</ETag>
</CopyPartResult>
```

- 指定versionId进行UploadPart Copy的请求示例

```
PUT /multipart.data?partNumber=2&uploadId=63C06A5CFF6F4AE4A6BB3AD7F01C**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 07:01:56 GMT
Authorization: OSS 6jfttm6x6san0ewtgyk****:v/sJFtYvg7DTa4pJ2AMShZL/****
x-oss-copy-source: /oss-example/src-object?versionId=CAEQMxiBgMC0vs6D0BYiIGJiZWRjOTRjNTg0NzQ1MTRiN2Y1OTYxMTdkYjQ0****
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
x-oss-copy-source-version-id: CAEQMxiBgMC0vs6D0BYiIGJiZWRjOTRjNTg0NzQ1MTRiN2Y1OTYxMTdkYjQ0****
x-oss-request-id: 5CAC4364B7AEADE017000660
Date: Tue, 09 Apr 2019 07:01:56 GMT
<?xml version="1.0" encoding="UTF-8"?>
<CopyPartResult>
  <LastModified>2019-04-09T07:01:56.000Z</LastModified>
  <ETag>"25A9F4ABFCC05743DF6E2C886C56****"</ETag>
</CopyPartResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [C](#)
- [.NET](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
OperationNotSupported	400	Bucket的类型为Archive时调用UploadPart Copy接口。

7.2.4. CompleteMultipartUpload

在将所有数据Part都上传完成后，您必须调用CompleteMultipartUpload接口来完成整个文件的分片上传。

注意事项

调用CompleteMultipartUpload操作时，用户必须提供所有有效的Part列表（包括Part Number和ETag）。OSS收到用户提交的Part列表后，会逐一验证每个Part的有效性。当所有的Part验证通过后，OSS将把这些Part组合成一个完整的Object。

● 确认Part的大小

CompleteMultipartUpload时会确认除最后一个Part以外所有Part的大小是否都大于或等于100 KB，并检查用户提交的Part列表中的每一个Part Number和ETag。所以在上传Part时，客户端除了需要记录Part号码外，还需要记录每次上传Part成功后服务器返回的ETag值。

● 处理请求

由于OSS处理CompleteMultipartUpload请求时会持续一定的时间。在这段时间内，如果客户端与OSS之间连接中断，OSS仍会继续该请求。

● Part Number

服务端在调用CompleteMultipartUpload接口时会对Part Number做校验。

Part Number取值为1~10000。Part Number可以不连续，但必须升序排列。例如第一个Part的Part Number是1，第二个Part的Part Number可以是5。

● UploadId

同一个Object可以同时拥有不同的UploadId，当Complete一个UploadId后，此UploadId将无效，但该Object的其他UploadId不受影响。

● x-oss-server-side-encryption请求头

如果调用InitiateMultipartUpload接口时，指定了x-oss-server-side-encryption请求头，则在CompleteMultipartUpload的响应头中返回x-oss-server-side-encryption，其值表示该Object的服务器端加密算法。

版本控制

在开启版本控制的情况下，调用CompleteMultipartUpload接口来完成整个文件的Multipart Upload，OSS会为整个文件生成唯一的版本ID，并在响应header中以x-oss-version-id的形式返回。

请求语法

```
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>
```

请求参数 (Request Parameters)

调用CompleteMultipartUpload接口时，可以通过Encoding-type对返回结果中的Key进行编码。

名称	类型	描述
----	----	----

名称	类型	描述
encoding-type	字符串	指定对返回的Key进行编码，目前只支持URL编码。 Key使用UTF-8字符，但XML 1.0标准不支持解析一些控制字符，例如ASCII码值从0到10的字符。当Key中包含XML 1.0标准不支持的控制字符时，您可以通过指定Encoding-type对返回的Key进行编码。 默认值：无 有效值：<i>url</i>

请求头

名称	类型	是否必选	描述
x-oss-forbid-overwrite	字符串	否	指定CompleteMultipartUpload操作时是否覆盖同名Object。 - 不指定x-oss-forbid-overwrite或者指定x-oss-forbid-overwrite为<i>false</i>时，表示允许覆盖同名Object。 - 指定x-oss-forbid-overwrite为<i>true</i>时，表示禁止覆盖同名Object。  说明 - 当目标Bucket的版本控制状态为“开启”或“暂停”时，x-oss-forbid-overwrite请求Header设置无效，即允许覆盖同名Object。 - 设置x-oss-forbid-overwrite请求Header会导致QPS处理性能下降，如果您有大量的操作需要使用x-oss-forbid-overwrite请求Header（QPS > 1000），请工单联系我们进行确认，避免影响您的业务。

名称	类型	是否必选	描述
x-oss-complete-all:yes	字符串	否	指定是否列举当前UploadId已上传的所有Part。 - 如果指定了x-oss-complete-all:yes，则OSS会列举当前UploadId已上传的所有Part，然后按照PartNumber的序号排序并执行CompleteMultipartUpload操作。执行CompleteMultipartUpload过程中无法检测正在上传或者漏传的Part，因此用户需要自己确保Part的完整性。 - 如果指定了x-oss-complete-all:yes，则不允许继续指定body，否则报错。 - 如果指定了x-oss-complete-all:yes，response的格式保持不变。

此接口还需要包含Host、Date等公共请求头。关于公共请求头的更多信息，请参见[公共HTTP头定义](#)。

请求元素

名称	类型	描述
CompleteMultipartUpload	容器	保存CompleteMultipartUpload请求内容的容器。 子节点：一个或多个Part元素 父节点：无
ETag	字符串	Part成功上传后，OSS返回的ETag值。 父节点：Part
Part	容器	保存已上传Part信息的容器。 子节点：ETag、PartNumber 父节点：CompleteMultipartUpload
PartNumber	整数	Part数目。 父节点：Part

响应元素

名称	类型	描述
Bucket	字符串	Bucket名称。 父节点：CompleteMultipartUploadResult

名称	类型	描述
CompleteMultipartUploadResult	容器	保存Complete Multipart Upload请求结果的容器。 子节点：Bucket、Key、ETag、Location 父节点：None
ETag	字符串	Object生成时会创建相应的ETag，ETag用于标识一个Object的内容。 通过CompleteMultipartUpload请求创建的Object，ETag值是基于一定计算规则生成的唯一值，但不是其内容的MD5值。 ❓ 说明 ETag值可以用于检查Object内容是否发生变化。不建议使用ETag作为Object内容的MD5来校验数据完整性。 父节点：CompleteMultipartUploadResult
Location	字符串	新创建Object的URL。 父节点：CompleteMultipartUploadResult
Key	字符串	新创建Object的名字。 父节点：CompleteMultipartUploadResult
EncodingType	字符串	显示返回结果中编码使用的类型。如果请求参数中指定了Encoding-type，那返回的结果会对Key进行编码。 父节点：容器

示例

- 未开启版本控制
请求示例

```
POST /multipart.data?uploadId=0004B9B2D2F7815C432C9057C031****&encoding-type=url 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 qn6qrrqx02oawuk53otf****:8VwFhFUWmVecK6jQlHXMK/z****
<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"3349DC700140D7F86A0784842780****"</ETag>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <ETag>"8EFDA8BE206636A695359836FE0A****"</ETag>
  </Part>
  <Part>
    <PartNumber>8</PartNumber>
    <ETag>"8C315065167132444177411FDA14****"</ETag>
  </Part>
</CompleteMultipartUpload>
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Content-Length: 329
Content-Type: Application/xml
Connection: keep-alive
x-oss-request-id: 594f0751-3b1e-168f-4501-4ac71d21****
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" >
  <EncodingType>url</EncodingType>
  <Location>http://oss-example.oss-cn-hangzhou.aliyuncs.com /multipart.data</Location>
  <Bucket>oss-example</Bucket>
  <Key>multipart.data</Key>
  <ETag>"B864DB6A936D376F9F8D3ED3BBE540****"</ETag>
</CompleteMultipartUploadResult>
```

- 已启用版本控制

请求示例

```
POST /multipart.data?uploadId=63C06A5CFF6F4AE4A6BB3AD7F01C**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 223
Date: Tue, 09 Apr 2019 07:01:56 GMT
Authorization: OSS 6jfttm6x6s****:XljBrYBYxDnxKdFMj9WYI6qu****
<CompleteMultipartUpload>
  <Part>
    <PartNumber>1</PartNumber>
    <ETag>"25A9F4ABFCC05743DF6E2C886C56****"</ETag>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <ETag>"25A9F4ABFCC05743DF6E2C886C56****"</ETag>
  </Part>
</CompleteMultipartUpload>
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Content-Length: 314
Content-Type: Application/xml
Connection: keep-alive
x-oss-version-id: CAEQMxiBgID6v86D0BYiIDc3ZDI0YTBjZGQzYjQ2Mjk4OWVjYWNIbDljYzhln****
x-oss-request-id: 5CAC4364B7AEADE017000662
Date: Tue, 09 Apr 2019 07:01:56 GMT
<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult>
  <Location>http://oss-example.oss-cn-hangzhou.aliyuncs.com/multipart.data</Location>
  <Bucket>oss-example</Bucket>
  <Key>multipart.data</Key>
  <ETag>"097DE458AD02B5F89F9D0530231876****"</ETag>
</CompleteMultipartUploadResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [PHP](#)
- [C](#)
- [Node.js](#)
- [Android](#)

错误码

错误码	HTTP 状态码	描述
InvalidDigest	400	为了保证数据在网络传输过程中不出现错误，用户发送请求时可以携带Content-MD5，OSS计算上传数据的MD5与用户上传的MD5值不一致。
FileAlreadyExists	409	当请求的Header中携带 <code>x-oss-forbid-overwrite=true</code> 时，表示禁止覆盖同名文件。如果已存在同名文件，则返回该错误。

7.2.5. AbortMultipartUpload

Abort Multipart Upload接口用于取消Multipart Upload事件并删除对应的Part数据。

注意事项

调用Abort Multipart Upload接口时，有如下注意事项：

- 您需要提供Multipart Upload事件相应的uploadId。
- 取消一个Multipart Upload事件过程中，如果所属的某些Part仍然在上传，那么此次取消操作将无法删除这些Part。
- 取消一个Multipart Upload事件后，您无法再使用此uploadId做任何操作，已经上传的Part数据也会被删除。
- 建议您及时完成分片上传或者取消分片上传，因为已上传但是未完成或未取消的分片会占用存储空间进而产生存储费用。

请求语法

```
DELETE /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

请求元素

名称	类型	是否必选	描述
uploadId	字符串	是	此次Multipart Upload事件的唯一标识。

其他公共请求头例如Host、Date等，详情请参见[公共HTTP头定义](#)。

示例

- 请求示例

```
Delete /multipart.data?&uploadId=0004B9895DBBB6E**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:J/lICfXEvPmmSW86bBAfMmUm****
```

- 返回示例

```
HTTP/1.1 204
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 059a22ba-6ba9-daed-5f3a-e48027df****
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [PHP](#)
- [C](#)
- [Node.js](#)
- [Android](#)

错误码

错误码	HTTP 状态码	描述
NoSuchUpload	404	此uploadId不存在。

7.2.6. ListMultipartUploads

调用ListMultipartUploads接口列举所有执行中的Multipart Upload事件，即已经初始化但还未完成（Complete）或者还未中止（Abort）的Multipart Upload事件。

请求语法

```
Get /?uploads HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

请求参数

名称	类型	描述
delimiter	字符串	用于对Object名称进行分组的字符。所有名称包含指定的前缀且首次出现delimiter字符之间的Object作为一组元素CommonPrefixes。
max-uploads	字符串	限定此次返回Multipart Upload事件的最大个数，默认值为1000。最大值为1000。

名称	类型	描述
key-marker	字符串	与upload-id-marker参数配合使用，用于指定返回结果的起始位置。 - 如果未设置upload-id-marker参数，查询结果中包含所有Object名称的字典序大于key-marker参数值的Multipart Upload事件。 - 如果设置了upload-id-marker参数，查询结果中包含： - 所有Object名称的字典序大于key-marker参数值的Multipart Upload事件。 - Object名称等于key-marker参数值，但是UploadId比upload-id-marker参数值大的Multipart Upload事件。
prefix	字符串	限定返回的Object Key必须以prefix作为前缀。注意使用prefix查询时，返回的Key中仍会包含prefix。  说明 您可以灵活地使用prefix参数对Bucket内的Object进行分组管理（类似文件夹功能）。
upload-id-marker	字符串	与key-marker参数配合使用，用于指定返回结果的起始位置。 - 如果未设置key-marker参数，则OSS会忽略upload-id-marker参数。 - 如果设置了key-marker参数，查询结果中包含： - 所有Object名称的字典序大于key-marker参数值的Multipart Upload事件。 - Object名称等于key-marker参数值，但是UploadId比upload-id-marker参数值大的Multipart Upload事件。
encoding-type	字符串	指定对返回的内容进行编码，指定编码的类型。Delimiter、KeyMarker、Prefix、NextKeyMarker和Key使用UTF-8字符，但xml 1.0标准不支持解析一些控制字符，例如ASCII值从0到10的字符。对于包含xml 1.0标准不支持的控制字符，可以通过指定encoding-type对返回的Delimiter、KeyMarker、Prefix、NextKeyMarker和Key进行编码。 默认值：无

响应元素

名称	类型	描述
----	----	----

名称	类型	描述
ListMultipartUploadsResult	容器	保存ListMultipartUpload请求结果的容器。 子节点：Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MaxUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated, Upload 父节点：None
Bucket	字符串	Bucket名称。 父节点：ListMultipartUploadsResult
EncodingType	字符串	指明返回结果中编码使用的类型。如果请求参数中指定了encoding-type, 那返回的结果会对Delimiter、KeyMarker、Prefix、NextKeyMarker和Key这些元素进行编码。 父节点：ListMultipartUploadsResult
KeyMarker	字符串	列表的起始Object位置。 父节点：ListMultipartUploadsResult
UploadIdMarker	字符串	列表的起始UploadId位置。 父节点：ListMultipartUploadsResult
NextKeyMarker	字符串	如果本次没有返回全部结果，响应请求中将包含NextKeyMarker元素，用于表示接下来请求的KeyMarker值。 父节点：ListMultipartUploadsResult
NextUploadMarker	字符串	如果本次没有返回全部结果，响应请求中将包含NextUploadMarker元素，用于表示接下来请求的UploadMarker值。 父节点：ListMultipartUploadsResult
MaxUploads	整数	返回的最大Upload个数。 父节点：ListMultipartUploadsResult
IsTruncated	枚举字符串	表示本次返回的MultipartUpload结果列表是否被截断。取值范围如下： <code>true</code>：表示本次没有返回全部结果。 <code>false</code>（默认）：表示本次已经返回了全部结果。 父节点：ListMultipartUploadsResult

名称	类型	描述
Upload	容器	保存Multipart Upload事件信息的容器。 子节点：Key, UploadId, Initiated 父节点：ListMultipartUploadsResult
Key	字符串	初始化Multipart Upload事件的Object名称。 父节点：Upload ? 说明 OSS的返回结果按照Object名称字典序升序排列，对于同一个Object，则按照UploadId的字典序升序排列。
UploadId	字符串	Multipart Upload事件的ID。 父节点：Upload
Initiated	日期	Multipart Upload事件初始化的时间。 父节点：Upload

示例

请求示例

```
Get /?uploads HTTP/1.1
Host:oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 23 Feb 2012 06:14:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JX75CtQqsmBBz+dcivn7kwBM****
```

返回示例

```
HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1839
Content-type: application/xml
x-oss-request-id: 58a41847-3d93-1905-20db-ba6f561c****
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</NextUploadIdMarker>
  <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>0004B999EF5A239BB9138C6227D6****</UploadId>
    <Initiated>2012-02-23T04:18:23.000Z</Initiated>
  </Upload>
  <Upload>
    <Key>oss.avi</Key>
    <UploadId>0004B99B8E707874FC2D692FA5D7****</UploadId>
    <Initiated>2012-02-23T06:14:27.000Z</Initiated>
  </Upload>
</ListMultipartUploadsResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Go](#)
- [C++](#)
- [PHP](#)
- [.NET](#)

7.2.7. ListParts

ListParts接口用于列举指定Upload ID所属的所有已经上传成功Part。

注意事项

- OSS的返回结果按照Part号码升序排列。

- 由于网络传输可能出错，所以不推荐使用ListParts返回结果中的Part Number和ETag值来生成已经上传成功的Part列表。

请求语法

```
Get /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

请求头

此接口仅涉及公共请求头。更多信息，请参见公共请求头（Common Request Headers）。

请求参数（Request Parameters）

名称	类型	示例值	描述
uploadId	字符串	0004B999EF5A239BB9138C6227D69F95	MultipartUpload事件的ID。 默认值：无
max-parts	整数	1000	规定在OSS响应中的最大Part数目。 默认值：1,000 最大值：1,000
part-number-marker	整数	100	指定List的起始位置，只有Part Number数目大于该参数的Part会被列出。 默认值：无
encoding-type	字符串	url	指定对返回的内容进行编码，指定编码的类型。Key使用UTF-8字符，但xml 1.0标准不支持解析一些控制字符，比如ascii值从0到10的字符。对于Key中包含xml 1.0标准不支持的控制字符，可以通过指定Encoding-type对返回的Key进行编码。 默认值：无 可选值：url

响应头

此接口仅涉及公共响应头。更多信息，请参见公共响应头（Common Response Headers）。

响应元素（Response Elements）

名称	类型	示例值	描述
ListPartsResult	容器	不涉及	保存List Part请求结果的容器。 子节点：Bucket, Key, UploadId, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part 父节点：无
Bucket	字符串	multipart_upload	Bucket名称。 父节点：ListPartsResult
EncodingType	字符串	url	指明对返回结果进行编码使用的类型。如果请求的参数中指定了Encoding-type, 那会对返回结果中的Key进行编码。 父节点：ListPartsResult
Key	字符串	multipart.data	Object名称。 父节点：ListPartsResult
UploadId	字符串	0004B999EF5A239BB9138C6227D69F95	Upload事件ID。 父节点：ListPartsResult
PartNumberMarker	整数	10	本次List结果的Part Number起始位置。 父节点：ListPartsResult
NextPartNumberMarker	整数	5	如果本次没有返回全部结果, 响应请求中将包含NextPartNumberMarker元素, 用于标明接下来请求的PartNumberMarker值。 父节点：ListPartsResult
MaxParts	整数	1000	返回请求中最大的Part数目。 父节点：ListPartsResult
IsTruncated	枚举字符串	false	标明本次返回的ListParts结果列表是否被截断。“true”表示本次没有返回全部结果; “false”表示本次已经返回了全部结果。 有效值: true、false 父节点：ListPartsResult

名称	类型	示例值	描述
Part	容器	不涉及	保存Part信息的容器。 子节点： PartNumber,LastModified,ETag,Size 父节点：ListPartsResult
PartNumber	整数	1	标示Part的数字。 父节点：ListPartsResult.Part
LastModified	日期	2012-02-23T07:01:34.000Z	Part上传的时间。 父节点：ListPartsResult.part
ETag	字符串	3349DC700140D7F86A0784842780****	已上传Part内容的ETag。 父节点：ListPartsResult.Part
Size	整数	6291456	已上传Part大小。 父节点：ListPartsResult.Part

示例

请求示例

```
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 qn6qrrxo2oawuk53otfjbyc:4qOnUMc9UQWqkz8wDqD3llsa9P8=
```

返回示例

```
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>"3349DC700140D7F86A0784842780*****"</ETag>
    <Size>6291456</Size>
  </Part>
  <Part>
    <PartNumber>2</PartNumber>
    <LastModified>2012-02-23T07:01:12.000Z</LastModified>
    <ETag>"3349DC700140D7F86A0784842780*****"</ETag>
    <Size>6291456</Size>
  </Part>
  <Part>
    <PartNumber>5</PartNumber>
    <LastModified>2012-02-23T07:02:03.000Z</LastModified>
    <ETag>"7265F4D211B56873A381D321F586*****"</ETag>
    <Size>1024</Size>
  </Part>
</ListPartsResult>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)

7.3. 权限控制（ACL）

7.3.1. PutObjectACL

调用Put ObjectACL接口修改文件（Object）的访问权限（ACL）。此操作只有Bucket Owner有权限执行，且需对Object有读写权限。

版本控制

调用Put ObjectACL接口时，默认只能设置Object当前版本的ACL。您可以通过指定versionId参数来设置指定Object版本的ACL。如果Object的对应版本为删除标记（Delete Marker），则OSS将返回404 Not Found。

ACL说明

Put ObjectACL接口通过Put 请求中的 `x-oss-object-acl` 头来设置Object ACL。目前Object包括如下四种访问权限。

名称	描述
private	Object是私有资源。只有该Object的Owner拥有该Object的读写权限，其他用户没有权限操作该Object。
public-read	Object是公共读资源。Object Owner拥有该Object的读写权限。非Object Owner只有该Object的读权限。
public-read-write	Object是公共读写资源。所有用户拥有对该Object的读写权限。
default	Object遵循其所在Bucket的读写权限，即Bucket是什么权限，Object就是什么权限。

?

说明

- Object ACL优先级高于Bucket ACL。例如Bucket ACL是private的，而Object ACL是public-read-write的，则所有用户都拥有该Object的访问权限，即使该Bucket是私有Bucket。如果某个Object未设置过ACL，则访问权限遵循Bucket ACL。
- Object的读操作包括Get Object、HeadObject、CopyObject和UploadPart Copy中的对原Object的读；Object的写操作包括Put Object、Post Object、AppendObject、DeleteObject、DeleteMultipleObjects、CompleteMultipart Upload以及CopyObject对新Object的写。
- 您还可以在进行Object的写操作时，在请求头中带上x-oss-object-acl来设置Object ACL，效果与Put ObjectACL等同。例如Put Object时在请求头中带上x-oss-object-acl可以在上传一个Object的同时设置此Object的ACL。

请求语法

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

请求头

名称	类型	是否必选	示例值	描述
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
x-oss-object-acl	字符串	否	public-read	指定OSS创建Object时的访问权限。 取值： <i>default</i>（默认）：Object遵循所在存储空间的访问权限。 <i>private</i>：Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户没有权限操作该Object。 <i>public-read</i>：Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户只有该Object的读权限。请谨慎使用该权限。 <i>public-read-write</i>：Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息，请参见基于读写权限ACL的权限控制。

此接口还需要包含Host、Date等公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

示例

- 未开启版本控制

请求示例

```
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:KU5h8YMUC78M30dXqf3JxrTZ****
```

返回示例

```
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
```

- 已启用版本控制

请求示例

```
PUT /example?acl&versionId=CAEQMhiBgIC3rpSD0BYiIDBjYTk5MmlzN2JlNjQxZTFiNGlzM2E3OTliODA0****
HTTP/1.1
x-oss-object-acl: public-read
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:30:11 GMT
Authorization: OSS qctg2ns3l8u51iu:UTsv3F7L34v+ECq52vURdCSv****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-version-id: CAEQMhiBgIC3rpSD0BYiIDBjYTk5MmlzN2JlNjQxZTFiNGlzM2E3OTliODA0****
x-oss-request-id: 5CAC3BF3B7AEADE017000624
Date: Tue, 09 Apr 2019 06:30:11 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [.NET](#)
- [Node.js](#)
- [Browser.js](#)

错误码

错误码	HTTP状态码	描述
AccessDenied	403	接口使用者不是Bucket Owner或者Bucket Owner对Object没有读写权限。
InvalidArgument	400	指定的x-oss-object-acl值无效。

7.3.2. GetObjectACL

调用GetObjectACL接口获取存储空间（Bucket）下某个文件（Object）的访问权限（ACL）。

版本控制

调用GetObjectACL接口时，默认只能获取Object当前版本的ACL。您可以通过指定versionId参数来获取指定Object版本的ACL。如果Object的对应版本为删除标记（Delete Marker），则OSS将返回404 Not Found。

 **说明** 如果一个Object从未设置过ACL，则调用GetObjectACL时，返回的ObjectACL为default，表示该Object的ACL遵循Bucket ACL。即如果Bucket的访问权限是private，则该Object的访问权限也是private。

请求语法

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

响应元素

名称	类型	描述
AccessControlList	容器	存储ACL信息的容器。 父节点：AccessControlPolicy
AccessControlPolicy	容器	保存Get Object ACL结果的容器。 父节点：None
DisplayName	字符串	Bucket拥有者的名称（目前和用户ID一致）。 父节点：AccessControlPolicy.Owner
Grant	枚举字符串	Object的ACL权限。 有效值： <i>private</i> , <i>public-read</i> , <i>public-read-write</i> 父节点：AccessControlPolicy.AccessControlList
ID	字符串	Bucket拥有者的用户ID。 父节点：AccessControlPolicy.Owner
Owner	容器	保存Bucket拥有者信息的容器。 父节点：AccessControlPolicy

示例

- 未开启版本控制

请求示例

```
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+WwIfNm0Fmgb****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 253
Content-Type: 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>
```

- 已启用版本控制

请求示例

```
GET /example?acl&versionId=CAEQMhiBgMC1qpSD0BYiIGQ0ZmI5ZDEyYWVvKNTQwMjBiNTliY2NjNmY3ZTVk
**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:30:10 GMT
Authorization: OSS qctg2ns3l8u51iu:w4DK66Kb/0M9GJKdsrpNs8l1****
```

返回示例

```
HTTP/1.1 200 OK
x-oss-version-id: CAEQMhiBgMC1qpSD0BYiIGQ0ZmI5ZDEyYWVvKNTQwMjBiNTliY2NjNmY3ZTVk****
x-oss-request-id: 5CAC3BF2B7AEADE017000621
Date: Tue, 09 Apr 2019 06:30:10 GMT
Content-Length: 261
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy>
  <Owner>
    <ID>1234513715092****</ID>
    <DisplayName>1234513715092****</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>public-read</Grant>
  </AccessControlList>
</AccessControlPolicy>
```

SDK

此接口所对应的各语言SDK如下：

- [Java](#)

- [Python](#)
- [PHP](#)
- [Go](#)
- [.NET](#)

错误码

错误码	HTTP状态码	错误信息	描述
AccessDenied	403	You do not have read acl permission on this object.	没有操作权限。只有Bucket的拥有者才能使用GetObjectACL接口来获取该Bucket下某个Object的ACL。

7.4. 软链接（Symlink）

7.4.1. PutSymlink

调用PutSymlink接口用于为OSS的目标文件（Target Object）创建软链接（Symlink），您可以通过该软链接访问Target Object。

注意事项

- 使用PutSymlink接口创建软链接时不会检查目标文件是否存在、目标文件类型是否合法以及目标文件是否有访问权限。
- Symlink自身的访问权限（ACL）以及目标文件的ACL检查仅会在GetObject等需要访问目标文件的API中进行。
- 使用PutSymlink接口时，携带以x-oss-meta-为前缀的参数，则被视为user meta，例如x-oss-meta-location。一个Object可以有多个类似的参数，但所有的user meta总大小不能超过8 KB。
- 默认情况下，如果试图添加的文件已经存在，并且有访问权限，则新添加的文件将覆盖原来的文件，成功添加后将返回200 OK。

版本控制

您可以通过Target Object创建的软链接指向Target Object的当前版本。

软链接本身也可以有多个版本，每个不同的版本可以指向不同的Target Object，版本ID由OSS自动生成，在响应Header中返回x-oss-version-id。

请求语法

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

请求头

名称	类型	是否必选	描述
x-oss-forbid-overwrite	字符串	否	指定PutSymlink操作时是否覆盖同名Object。 - 不指定x-oss-forbid-overwrite或者指定x-oss-forbid-overwrite为<code>false</code>时，表示允许覆盖同名Object。 - 指定x-oss-forbid-overwrite为<code>true</code>时，表示禁止覆盖同名Object。 设置x-oss-forbid-overwrite请求Header会导致QPS处理性能下降，如果您有大量的操作需要使用x-oss-forbid-overwrite请求Header（QPS>1000），请联系技术支持，避免影响您的业务。  说明 当目标Bucket处于已开启或已暂停版本控制状态时，x-oss-forbid-overwrite请求Header设置无效，即允许覆盖同名Object。
x-oss-symlink-target	字符串	是	软链接指向的目标文件。 合法值：命名规范同Object - TargetObjectName同ObjectName一样，需要对其进行URL编码。 - 软链接的目标文件类型不能为软链接。
x-oss-object-acl	字符串	否	指定OSS创建Object时的访问权限。 取值： <code>default</code>（默认）：Object遵循所在存储空间的访问权限。 <code>private</code>：Object是私有资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户没有权限操作该Object。 <code>public-read</code>：Object是公共读资源。只有Object的拥有者和授权用户有该Object的读写权限，其他用户只有该Object的读权限。请谨慎使用该权限。 <code>public-read-write</code>：Object是公共读写资源。所有用户都有该Object的读写权限。请谨慎使用该权限。 关于访问权限的更多信息，请参见基于读写权限ACL的权限控制。

名称	类型	是否必选	描述
x-oss-storage-class	字符串	否	指定Object的存储类型。 对于任意存储类型的Bucket，如果上传Object时指定此参数，则此次上传的Object将存储为指定的类型。例如在IA类型的Bucket中上传Object时，如果指定x-oss-storage-class为Standard，则该Object直接存储为Standard。 取值： • <i>Standard</i>: 标准存储 • <i>IA</i>: 低频访问 • <i>Archive</i>: 归档存储 IA与Archive类型的单个Object大小如果不足64 KB，则会按64 KB计量计费。建议在使用PutSymlink接口时不要将Object的存储类型指定为IA或Archive。 关于存储类型的更多信息，请参见存储类型介绍。

此接口还需要包含Host、Date等公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅包含公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

● 请求示例

```
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+dcGKw5x2****= x-oss-symlink-target: oss****
x-oss-storage-class: Standard
```

返回示例

```
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: "0A477B89B4602AA8DECB8E19BFD4****"
```

● 版本控制请求示例

```
PUT /link-to-oss.jpg?symlink HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:50:48 GMT
Authorization: OSS o3shiyktjw16xw1:NVXXKiyUJ2tg07PxINinU0eO****
x-oss-symlink-target: oss.jpg
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 09 Apr 2019 06:50:48 GMT
Content-Length: 0
Connection: keep-alive
x-oss-version-id: CAEQNRiBgMClj7qD0BYiIDQ5Y2QyMjc3NGZkODRlMTU5M2VkY2U3MWRiNGRh****
x-oss-request-id: 5CAC40C8B7AEADE01700064B
ETag: "136A5E127272200EDAB170DD84DE****"
```

SDK

PutSymlink接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
InvalidArgument	400	StorageClass的值不合法。
FileAlreadyExists	409	当请求Header中携带x-oss-forbid-overwrite=true时，表示禁止覆盖同名文件。如果文件已存在，则返回此错误。
FileImmutable	409	Bucket内的数据处于被保护状态时，若您尝试删除或修改这些数据，将返回此错误码。

7.4.2. GetSymlink

调用GetSymlink接口获取软链接。此操作需要您对该软链接有读权限。

版本控制

GetSymlink接口默认获取软链接的当前版本。允许通过指定versionId来获取指定版本。如果软链接的当前版本为删除标记，OSS会返回404 Not Found，在响应header中返回x-oss-delete-marker = true以及版本ID：x-oss-version-id。删除标记没有关联数据，因此也没有软链接指向的Target Object。

请求语法

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

响应头

名称	类型	描述
x-oss-symlink-target	字符串	软链接指向的目标文件。

示例

- 请求示例

```
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:UNQDb7GapEgJCZkcde6OhZ9J****
```

返回示例

```
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: "A797938C31D59EDD08D86188F6D5****"
```

- 指定versionId获取软链接的请求示例

```
GET /link-to-oss.jpg?symlink&versionId=CAEQNRiBgMClj7qD0BYiIDQ5Y2QyMjc3NGZkODRIMTU5M2VkY2U3MWRiNGRh**** HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 06:50:48 GMT
Authorization: OSS o3shiyktjw16xw1:LfKjDsA81MCTcBKV6h+Z/95f****
```

返回示例

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 09 Apr 2019 06:50:48 GMT
Last-Modified: Tue, 09 Apr 2019 06:50:48 GMT
Content-Length: 0
Connection: keep-alive
x-oss-version-id: CAEQNRiBgMClj7qD0BYiIDQ5Y2QyMjc3NGZkODRlMTU5M2VkY2U3MWRiNGRh****
x-oss-request-id: 5CAC40C8B7AEADE01700064D
x-oss-symlink-target: oss.jpg
ETag: "40CF4D450730DCCD1A78566FAE35*****"
```

SDK

GetSymlink接口所对应的各语言SDK如下：

- [Java](#)
- [Python](#)
- [PHP](#)
- [Go](#)
- [C](#)
- [C++](#)
- [.NET](#)
- [Android](#)
- [iOS](#)
- [Node.js](#)

错误码

错误码	HTTP状态码	描述
NoSuchKey	404	文件不存在。
NotSymlink	400	软链接不存在。

7.5. 标签（Tagging）

7.5.1. PutObjectTagging

调用PutObjectTagging接口设置或更新对象（Object）的标签（Tagging）信息。对象标签使用一组键值对（Key-Value）标记对象。

注意事项

- 单个Object最多能设置10个标签，Key不能重复。
- 每个Key长度不能超过128字符，每个Value长度不能超过256字符。
- Key和Value区分大小写。
- 标签的合法字符集包括大小写字母、数字、空格和以下符号：

+ - = . _ : /

通过HTTP Header的方式设置标签且标签中包含任意字符时，您需要对标签的Key和Value进行URL编码。

- 更改标签信息不会更新Object的Last-Modified时间。

关于对象标签的更多信息，请参见[对象标签](#)。

版本控制

调用Put Object Tagging接口时，默认设置Object当前版本的标签信息。您可以通过指定versionId参数来设置指定Object版本的标签信息。如果Object的对应版本为删除标记（Delete Marker），则OSS将返回404 Not Found。

请求语法

```
PUT /objectname?tagging
Content-Length: 114
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Mon, 18 Mar 2019 08:25:17 GMT
Authorization: SignatureValue
<Tagging>
<TagSet>
<Tag>
  <Key>Key</Key>
  <Value>Value</Value>
</Tag>
</TagSet>
</Tagging>
```

请求元素

名称	类型	是否必需	描述
Tagging	容器	是	标签集合。 子节点：TagSet
TagSet	容器	是	标签集合。 父节点：Tagging 子节点：Tag
Tag	容器	否	标签集合。 父节点：TagSet 子节点：Key、Value
Key	字符串	否	标签键。 父节点：Tag 子节点：无

名称	类型	是否必需	描述
Value	字符串	否	标签值。 父节点：Tag 子节点：无

示例

- 未开启版本控制

在未开启版本控制情况下，针对存储空间bucketname中的对象objectname，通过PUT请求设置{a:1}和{b:2}两个标签。标签设置成功后返回200 (OK)。

请求示例

```
PUT /objectname?tagging
Content-Length: 114
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Mon, 18 Mar 2019 08:25:17 GMT
Authorization: OSS *****
<Tagging>
<TagSet>
<Tag>
<Key>a</Key>
<Value>1</Value>
</Tag>
<Tag>
<Key>b</Key>
<Value>2</Value>
</Tag>
</TagSet>
</Tagging>
```

返回示例

```
200 (OK)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5C8F55ED461FB4A64C00****
date: Mon, 18 Mar 2019 08:25:17 GMT
```

- 已启用版本控制

在启用了版本控制的情况下，针对存储空间bucketname中的对象objectname的指定版本（即请求示例中的versionId），通过PUT请求设置{age:18}标签。标签设置成功后返回200 (OK)。

请求示例

```

PUT /objectname?tagging&versionId=CAEQExiBgID.jlmWlxcilDQ2ZjgwODlyNDk5MTRhNzBiYmQwYTZkMTYzZjM0****
Content-Length: 90
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 24 Jun 2020 08:58:15 GMT
Authorization: OSS *****.*****
<Tagging>
  <TagSet>
    <Tag>
      <Key>age</Key>
      <Value>18</Value>
    </Tag>
  </TagSet>
</Tagging>

```

返回示例

```

200 (OK)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5EF315A7FBD3EC3232B4****
date: Wed, 24 Jun 2020 08:58:15 GMT
x-oss-version-id: CAEQExiBgID.jlmWlxcilDQ2ZjgwODlyNDk5MTRhNzBiYmQwYTZkMTYzZjM0****

```

SDK

PutObjectTagging接口对应的各语言SDK示例如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

7.5.2. GetObjectTagging

调用GetObjectTagging接口获取对象（Object）的标签（Tagging）信息。

版本控制

调用GetObjectTagging接口时，默认只能获取Object当前版本的标签信息。您可以通过指定versionId参数来获取指定Object版本的标签信息。如果Object的对应版本为删除标记（Delete Marker），则OSS将返回404 Not Found。

请求语法

```

GET /objectname?tagging
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 20 Mar 2019 02:02:36 GMT
Authorization: SignatureValue

```


请求头

此接口仅包含公共请求头。关于公共请求头的更多信息，请参见公共请求头（Common Request Headers）。

响应头

此接口仅包含公共响应头。关于公共响应头的更多信息，请参见公共响应头（Common Response Headers）。

响应元素

名称	类型	描述
Tagging	容器	标签集合。 子节点：TagSet
TagSet	容器	标签集合。 父节点：Tagging 子节点：Tag
Tag	容器	标签集合。 父节点：TagSet 子节点：Key、Value
Key	字符串	标签键。 父节点：Tag 子节点：无
Value	字符串	标签值。 父节点：Tag 子节点：无

示例

- 未开启版本控制

在未开启版本控制的情况下，针对存储空间bucket name中的对象object name发起GET 请求时，获取到 {a:1}和{b:2}的标签信息。标签获取成功后返回200 (OK)。

请求示例

```
GET /objectname?tagging
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 20 Mar 2019 02:02:36 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:kZoYNv66bsmc10+dcGKw5x2P****
```

返回示例

```

200 (OK)
content-length: 209
server: AliyunOSS
x-oss-request-id: 5C919F38461FB4282600****
date: Wed, 20 Mar 2019 02:02:32 GMT
content-type: application/xml
<?xml version="1.0" encoding="UTF-8"?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>a</Key>
      <Value>1</Value>
    </Tag>
    <Tag>
      <Key>b</Key>
      <Value>2</Value>
    </Tag>
  </TagSet>
</Tagging>

```

- 已启用版本控制

在启用版本控制的情况下，针对存储空间bucket name中的对象object name的指定版本（即请求示例中的versionId）发起GET请求时，获取到{age:18}的标签信息。标签获取成功后返回200 (OK)。

请求示例

```

GET /objectname?tagging&versionId=CAEQExiBgID.jlmWlxcIDQ2ZjgwODlyNDk5MTRhNzBiYmQwYTZkMTYzZjM0****
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 24 Jun 2020 08:50:28 GMT
Authorization: OSS *****.*****

```

返回示例

```

200 (OK)
content-length: 161
server: AliyunOSS
x-oss-request-id: 5EF313D44506783438F3****
date: Wed, 24 Jun 2020 08:50:28 GMT
content-type: application/xml
x-oss-version-id: CAEQExiBgID.jlmWlxcIDQ2ZjgwODlyNDk5MTRhNzBiYmQwYTZkMTYzZjM0****
<?xml version="1.0" encoding="UTF-8"?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>age</Key>
      <Value>18</Value>
    </Tag>
  </TagSet>
</Tagging>

```

SDK

Get Object Tagging接口对应的各语言SDK示例如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

7.5.3. DeleteObjectTagging

调用DeleteObjectTagging接口删除指定对象（Object）的标签（Tagging）信息。

版本控制

调用DeleteObjectTagging接口时，默认只能删除Object当前版本的标签信息。您可以通过指定versionId参数来删除指定Object版本的标签信息。如果Object的对应版本为删除标记（Delete Marker），则OSS将返回404 Not Found。

请求语法

```
DELETE /objectname?tagging
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Mon, 18 Mar 2019 08:25:17 GMT
Authorization: SignatureValue
```

请求头

此接口仅包含公共请求头。关于公共请求头的更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅包含公共响应头。关于公共响应头的更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

- 未开启版本控制

在未开启版本控制的情况下，通过DELETE请求删除了存储空间bucket name中对象objectname已有的标签信息，OSS解析此请求后删除此对象中的所有标签。

请求示例

```
DELETE /objectname?tagging
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Tue, 09 Apr 2019 03:00:33 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:kZoYNv66bsmc10+dcGKw5x2P****
```

返回示例

```
204 (No Content)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5CAC0AD16D0232E2051B****
date: Tue, 09 Apr 2019 03:00:33 GMT
```

- 已启用版本控制

在启用版本控制的情况下，通过DELETE请求删除存储空间bucketname中对象objectname已有的标签信息，删除时指定了对象版本号（即请求示例中的versionId），OSS解析此请求后删除此版本对象中的所有标签。

请求示例

```
DELETE /objectname?tagging&versionId=CAEQExiBgID.jlmWlxcilDQ2ZjgwODIyNDk5MTRhNzBiYmQwYTZkMTYzZjM0****
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 24 Jun 2020 09:01:09 GMT
Authorization: OSS qn6qrrqxo2oawuk53otf****:kZoYNv66bsmc10+dcGKw5x2P****
```

返回示例

```
204 (No Content)
content-length: 0
server: AliyunOSS
x-oss-request-id: 5EF3165525D95C3338E8****
date: Wed, 24 Jun 2020 09:01:09 GMT
x-oss-version-id: CAEQExiBgID.jlmWlxcilDQ2ZjgwODIyNDk5MTRhNzBiYmQwYTZkMTYzZjM0****
```

SDK

DeleteObjectTagging接口对应的各语言SDK示例如下：

- [Java](#)
- [Python](#)
- [Go](#)
- [C++](#)
- [.NET](#)
- [Node.js](#)

8.关于LiveChannel的操作

8.1. LiveChannel简介

您可以使用RTMP协议将音视频数据上传到OSS，转储为指定格式的音视频文件。上传前需要先创建一个LiveChannel，以获取对应的推流地址。

通过RTMP协议上传音视频数据目前有以下限制：

- 只能使用RTMP推流的方式，不支持拉流。
- 必须包含视频流，且视频流格式为H264。
- 音频流是可选的，并且只支持AAC格式，其他格式的音频流会被丢弃。
- 转储只支持HLS协议。
- 一个LiveChannel同时只能有一个客户端向其推流。

8.2. RTMP推流地址及签名

本文介绍RTMP推流地址及其签名规则。

 **说明** 仅当Bucket ACL为非public-read-write时，推流地址需要签名后才可以使⽤。签名方法与OSS的URL签名类似。

RTMP推流地址

RTMP推流地址组成规则为 `rtmp://{bucket}.{host}/live/{channel}?{params}`，例如 `rtmp://examplebucket.oss-cn-hangzhou.aliyuncs.com/live/test-channel`。

- **bucket**：Bucket名称，例如 `examplebucket`。有关Bucket命名规范的更多信息，请参见[存储空间（Bucket）](#)。
- **host**：填写地域节点Endpoint，例如 `oss-cn-hangzhou.aliyuncs.com`。有关Endpoint的更多信息，请参见[访问域名和数据中心](#)。
- **live**：RTMP协议的App名称，OSS固定使用live。
- **channel**：channel名称。例如 `test-channel`。有关channel命名规范的更多信息，请参见[PutLiveChannel](#)。
- **params**：推流参数，与HTTP请求的query string相同，格式为 `varA=valueA&varB=valueB`。

RTMP推流支持的URL参数

RTMP推流支持的URL参数及描述如下表所示。

名称	描述
playlistName	指定生成的m3u8文件名称。  说明 生成的m3u8文件名称仍被添加 <code>\${channel_name}/</code> 前缀。

推流地址的签名规则

带签名的推流地址格式为 `rtmp://$bucket.$host/live/$channel?OSSAccessKeyId=xxx&Expires=yyy&Signature=zzz&${params}`。

推流地址的签名规则中包含的参数及描述如下表所示。

参数名称	描述
OSSAccessKeyId	与OSS HTTP签名的AccessKeyId相同。
Expires	Unix时间戳。
Signature	签名字符串。
params	其他参数。所有的参数都需要经过签名。

Signature的计算规则如下。

```
base64(hmac-sha1(AccessKeySecret,
+ Expires + "\n"
+ CanonicalizedParams
+ CanonicalizedResource))
```

Signature计算规则中涉及的参数及描述如下表所示。

名称	描述
CanonicalizedParams	按照param key字典序拼接所有参数，格式为 <code>key:value\n</code>。 ② 说明 - 如果参数个数为0，那此项为空。 - 参数中不包含SecurityToken、OSSAccessKeyId、Expire以及Signature。 - 每个param key只能出现一次。
CanonicalizedResource	格式为 <code>/BucketName/ChannelName</code>，例如 <code>examplebucket/test-channel</code>。

8.3. PutLiveChannel

通过RTMP协议上传音视频数据前，必须先调用该接口创建一个LiveChannel。调用PutLiveChannel接口会返回RTMP推流地址，以及对应的播放地址。

② 说明 您可以使用返回的地址进行推流、播放，您还可以根据该LiveChannel的名称来发起相关的操作，如查询推流状态、查询推流记录、禁止推流等。

请求语法

```
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>
```

请求头

名称	类型	是否必选	描述
ChannelName	字符串	是	必须符合ObjectName的命名规范，另外，ChannelName不能包含斜杠（/）。

请求元素

名称	类型	是否必选	描述
LiveChannelConfiguration	容器	是	保存LiveChannel配置的容器。 子节点：Description、Status、Target 父节点：无
Description	字符串	否	LiveChannel的描述信息，最长128字节。 子节点：无 父节点：LiveChannelConfiguration

名称	类型	是否必选	描述
Status	枚举字符串	否	指定LiveChannel的状态。 子节点：无 父节点：LiveChannelConfiguration 有效值： <i>enabled</i> 、 <i>disabled</i> 默认值： <i>enabled</i>
Target	容器	是	保存转储配置的容器。 子节点：Type、FragDuration、FragCount、PlaylistName 父节点：LiveChannelConfiguration
Type	枚举字符串	是	指定转储的类型。 子节点：无 父节点：Target 有效值： <i>HLS</i>  说明 - 转储类型为HLS时，OSS会在生成每个ts文件后更新m3u8文件。m3u8文件中最多包含最近的FragCount个ts文件。 - 转储类型为HLS时，写入当前ts文件的音视频数据时长达到FragDuration指定的时长后，OSS会在收到下一个关键帧的时切换到下一个ts文件；如果$\max(2 * \text{FragDuration}, 60\text{s})$后仍未收到下一个关键帧，OSS将强制切换文件，此时可能引起播放时卡顿。

名称	类型	是否必选	描述
FragDuration	字符串	否	当Type为HLS时，指定每个ts文件的时长。 单位：秒 子节点：无 父节点：Target 取值范围：[1, 100] 默认值：5 ② 说明 FragDuration和FragCount的默认值只有在两者都未指定时才会生效；指定了其中一个，则另一个的值也必须指定。
FragCount	字符串	否	当Type为HLS时，指定m3u8文件中包含ts文件的个数。 子节点：无 父节点：Target 取值范围：[1, 100] 默认值：3 ② 说明 FragDuration和FragCount的默认值只有在两者都未指定时才会生效；指定了其中一个，则另一个的值也必须指定。
PlaylistName	字符串	否	当Type为HLS时，指定生成的m3u8文件的名称。必须以“.m3u8”结尾，长度范围为[6, 128]。 子节点：无 父节点：Target 默认值：playlist.m3u8 取值范围：[6, 128]
Snapshot	容器	否	保存高频截图操作Snapshot 选项的容器。 子节点：RoleName、DestBucket、NotifyTopic、Interval、PornRec 父节点：Snapshot

名称	类型	是否必选	描述
RoleName	字符串	否	用于高频截图操作的角色名称，要求有DestBucket的写权限和向NotifyTopic发消息的权限。 子节点：无 父节点：Snapshot
DestBucket	字符串	否	保存高频截图目标Bucket，要求与当前Bucket是同一个Owner。 子节点：无 父节点：Snapshot
NotifyTopic	字符串	否	用于通知用户高频截图操作结果的MNS的Topic。 子节点：无 父节点：Snapshot
Interval	数字	否	高频截图的间隔长度。如果该段间隔时间内没有关键帧（I 帧），那么该间隔时间不截图。 单位：秒 子节点：无 父节点：Snapshot 取值范围：[1, 100]

响应元素

名称	类型	描述
CreateLiveChannelResult	容器	保存CreateLiveChannel请求结果的容器。 子节点：PublishUrls, PlayUrls 父节点：无
PublishUrls	容器	保存推流地址的容器。 子节点：Url 父节点：CreateLiveChannelResult

名称	类型	描述
Url	字符串	推流地址。 子节点：无 父节点：PublishUrls  说明 - 推流地址是未加签名的URL，如Bucket ACL非 public-read-write，则需先进行签名才可访问。 - 播放地址是未加签名的URL，如Bucket ACL为 private，则需先进行签名才可访问。
PlayUrls	容器	保存播放地址的容器。 子节点：Url 父节点：CreateLiveChannelResult
Url	字符串	播放地址。 子节点：无 父节点：PlayUrls

示例

请求示例

```
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+a****
<?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>
```

返回示例

```
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>
```

SDK

Java

8.4. ListLiveChannel

ListLiveChannel接口用于列举指定的LiveChannel。

请求语法

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

请求元素

名称	类型	是否必选	描述
Marker	字符串	否	设定结果从marker之后按字母排序的第一个开始返回。
Max-keys	字符串	否	限定此次返回LiveChannel的最大数。max-keys取值不能大于1000。 默认值：100
Prefix	字符串	否	限定返回的LiveChannel必须以prefix作为前缀。注意使用prefix查询时，返回的key中仍会包含prefix。

响应元素

名称	类型	描述
ListLiveChannelResult	容器	保存ListLiveChannel请求结果的容器。 子节点：Prefix、Marker、MaxKeys、IsTruncated、NextMarker、LiveChannel 父节点：无
Prefix	字符串	本次查询结果的开始前缀。 子节点：无 父节点：ListLiveChannelResult
Marker	字符串	本次ListLiveChannel的起点。 子节点：无 父节点：ListLiveChannelResult
MaxKeys	字符串	响应请求内返回结果的最大数目。 子节点：无 父节点：ListLiveChannelResult
IsTruncated	字符串	是否已返回所有的结果。 • true：表示本次请求已返回全部结果。 • false：表示本次请求未返回全部结果。 子节点：无 父节点：ListLiveChannelResult
NextMarker	字符串	如果本次没有返回全部结果，响应请求中将包含NextMarker元素，用于标明接下来请求的Marker值。 子节点：无 父节点：ListLiveChannelResult
LiveChannel	容器	保存返回每个LiveChannel信息的容器。 子节点：Name、Description、Status、LastModified、PublishUrls、PlayUrls 父节点：ListLiveChannelResult
Name	字符串	LiveChannel的名称。 子节点：无 父节点：LiveChannel

名称	类型	描述
Description	字符串	LiveChannel的描述信息。 子节点：无 父节点：LiveChannel
Status	枚举字符串	LiveChannel的状态。 子节点：无 父节点：LiveChannel 有效值： • <i>disabled</i>: 表示禁用LiveChannel。 • <i>enabled</i>: 表示启用LiveChannel。
LastModified	字符串	LiveChannel配置的最后修改时间。 格式：ISO8601 子节点：无 父节点：LiveChannel
PublishUrls	容器	保存LiveChannel对应的推流地址的容器。 子节点：Url 父节点：LiveChannel
Url	字符串	LiveChannel对应的推流地址。 子节点：无 父节点：PublishUrls
PlayUrls	容器	保存LiveChannel对应的播放地址的容器。 子节点：Url 父节点：LiveChannel
Url	字符串	LiveChannel对应的播放地址。 子节点：无 父节点：PlayUrls

示例

请求示例

```
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 YJjHKOKWDWIN****:TaX+tlc/Xsgpz6uRuqcbmUJs****
```

返回示例

```
HTTP/1.1 200
content-length: 656
server: AliyunOSS
connection: close
x-oss-request-id: 57BEA331B92475920B00****
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>
```

SDK

- [Java](#)
- [Python](#)

8.5. DeleteLiveChannel

DeleteLiveChannel接口用于删除指定的LiveChannel。

 说明

- 当有客户端正在向LiveChannel推流时，删除请求会失败。
- 本接口只会删除LiveChannel本身，不会删除推流生成的文件。

请求语法

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

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

示例

请求示例

```
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:ZbfvQ3XwmYEE8O9CX8kwVQY****
```

返回示例

```
HTTP/1.1 204
content-length: 0
server: AliyunOSS
connection: close
x-oss-request-id: 57BE9F0AB92475920B0*****
date: Thu, 25 Aug 2016 07:32:26 GMT
```

SDK

Java

8.6. PutLiveChannelStatus

LiveChannel分为启用（enabled）和禁用（disabled）两种状态。您可以使用PutLiveChannelStatus接口在两种状态之间进行切换。

注意事项

LiveChannel有如下注意事项：

- LiveChannel处于disabled状态时，OSS会禁止您向该LiveChannel进行推流操作。如果您正在向该LiveChannel推流，那么推流的客户端会被强制断开（会有10s左右的延迟）。
- 当没有客户端向该LiveChannel推流时，调用PutLiveChannelStatus重新创建LiveChannel也可以达到修改Status的目的。
- 当有客户端向该LiveChannel推流时，只能将LiveChannel的状态修改为disabled，无法调用PutLiveChannelStatus重新创建LiveChannel。

请求语法

```
PUT /ChannelName?live&status=NewStatus HTTP/1.1
Date: Tue, 25 Dec 2018 17:35:24 GMT
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

请求头

名称	类型	是否必选	描述
NewStatus	字符串	是	设置LiveChannel的Status。 有效值： - enabled：启用LiveChannel - disabled：禁用LiveChannel

有关此接口涉及的其他公共请求头，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头，详情请参见[公共响应头（Common Response Headers）](#)。

示例

请求示例

```
PUT /test-channel?live&status=disabled HTTP/1.1
Date: Tue, 25 Dec 2018 17:35:24 GMT
Host: test-bucket.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS YJHKOKWDWIN****:X/mBrSbkNoqM/JoAfRC0ytyQ****
```

返回示例

```
HTTP/1.1 200
Content-Length: 0
Server: AliyunOSS
Connection: close
x-oss-request-id: 57BE8422B92475920B00****
Date: Tue, 25 Dec 2018 17:35:24 GMT
```

SDK

- [Java](#)
- [Python](#)

8.7. GetLiveChannelInfo

GetLiveChannelInfo接口用于获取指定LiveChannel的配置信息。

请求语法

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

请求头

此接口仅涉及公共请求头。更多信息，请参见[公共请求头（Common Request Headers）](#)。

响应头

此接口仅涉及公共响应头。更多信息，请参见[公共响应头（Common Response Headers）](#)。

响应元素

名称	类型	示例值	描述
LiveChannelConfiguration	容器	不涉及	保存GetLiveChannelInfo返回结果的容器。 子节点：Description、Status、Target 父节点：无
Description	字符串	test	LiveChannel的描述信息。 子节点：无 父节点：LiveChannelConfiguration
Status	枚举字符串	enabled	LiveChannel的状态信息。 子节点：无 父节点：LiveChannelConfiguration 有效值： • <i>enabled</i>: 启用状态 • <i>disabled</i>: 禁用状态

名称	类型	示例值	描述
Target	容器	不涉及	保存LiveChannel转储配置的容器。 子节点：Type、FragDuration、FragCount、PlaylistName  说明 FragDuration、FragCount、PlaylistName只有当Type取值为HLS时才会返回。 父节点：LiveChannelConfiguration
Type	枚举字符串	HLS	当Type为HLS时，指定推流时转储文件类型。 子节点：无 父节点：Target 有效值：HLS
FragDuration	字符串	2	当Type为HLS时，指定每个ts文件的时长。 单位：秒 子节点：无 父节点：Target
FragCount	字符串	3	当Type为HLS时，指定m3u8文件中包含ts文件的个数。 子节点：无 父节点：Target
PlaylistName	字符串	playlist.m3u8	当Type为HLS时，指定生成的m3u8文件的名称。 子节点：无 父节点：Target

示例

请求示例

```
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 YJjHKOKWDWIN****:D6bDCRXKht58hin1BL83wxyG****
```

返回示例

```
HTTP/1.1 200
content-length: 475
server: AliyunOSS
connection: close
x-oss-request-id: 57BE87A8B92475920B00****
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>
```

SDK

Java

8.8. GetLiveChannelStat

GetLiveChannelStat接口用于获取指定LiveChannel的推流状态信息。

请求语法

```
GET /ChannelName?live&comp=stat HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

响应元素

名称	类型	描述
----	----	----

名称	类型	描述
LiveChannelStat	容器	保存GetLiveChannelStat返回结果的容器。 子节点：Status、ConnectedTime、Video、Audio 父节点：无
Status	枚举字符串	LiveChannel当前的推流状态描述。 子节点：无 父节点：LiveChannelStat 有效值：DisabledLiveIdle
ConnectedTime	字符串	当Status为Live时，表示当前客户端开始推流的时间。此元素使用ISO8601格式表示。 子节点：无 父节点：LiveChannelStat
RemoteAddr	字符串	当Status为Live时，表示当前推流客户端的IP地址。 子节点：无 父节点：LiveChannelStat
Video	容器	当Status为Live时，保存视频流信息的容器。  说明 Video、Audio容器只有在Status为Live时才会返回，但Status为Live时不一定返回这两个容器。例如，客户端已经连接到LiveChannel，但尚未发送音视频数据，这种情况不会返回这两个容器。 子节点：Width、Height、FrameRate、Bandwidth、Codec 父节点：LiveChannelStat
Width	字符串	当前视频流的画面宽度。 单位：像素 子节点：无 父节点：Video

名称	类型	描述
Height	字符串	当前视频流的画面高度。 单位：像素 子节点：无 父节点：Video
FrameRate	字符串	当前视频流的帧率。 子节点：无 父节点：Video
Bandwidth	字符串	当前视频流的码率。 单位：B/s 子节点：无 父节点：Video
Codec	枚举字符串	当前视频流的编码格式。 子节点：无 父节点：Video
Audio	容器	当Status为Live时，保存音频流信息的容器。  说明 Video、Audio容器只有在Status为Live时才会返回，但Status为Live时不一定返回这两个容器。例如，客户端已经连接到LiveChannel，但尚未发送音视频数据，这种情况不会返回这两个容器。 子节点：SampleRate、Bandwidth、Codec 父节点：LiveChannelStat
SampleRate	字符串	当前音频流的采样率。 子节点：无 父节点：Audio

名称	类型	描述
Bandwidth	字符串	当前音频流的码率。 ? 说明 Bandwidth为音频流/视频流最近一段时间内的平均码率。LiveChannel刚切换到Live状态时，返回的Bandwidth值可能为0。 单位：B/s 子节点：无 父节点：Audio
Codec	枚举字符串	当前音频流的编码格式。 子节点：无 父节点：Audio

示例

- 请求示例 1

```
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:fOzwkAgVTVSO1VKLPInQ0JY****
```

返回示例 1

```
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>
```

- 请求示例 2

```
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:WeC5joEaRzfSSS8xK0tlo7WT****
```

返回示例 2

```
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>
```

SDK

Java

8.9. GetLiveChannelHistory

GetLiveChannelHistory接口用于获取指定LiveChannel的推流记录。使用GetLiveChannelHistory接口最多会返回指定LiveChannel最近的10次推流记录。

请求语法

```
GET /ChannelName?live&comp=history HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

响应元素

名称	类型	描述
LiveChannelHistory	容器	保存GetLiveChannelHistory返回结果的容器。 子节点：LiveRecord 父节点：无

名称	类型	描述
LiveRecord	容器	保存一次推流记录信息的容器。 子节点：StartTime、EndTime、RemoteAddr 父节点：LiveChannelHistory
StartTime	字符串	推流开始时间，使用ISO8601格式表示。 子节点：无 父节点：LiveRecord
EndTime	字符串	推流结束时间，使用ISO8601格式表示。 子节点：无 父节点：LiveRecord
RemoteAddr	字符串	推流客户端的IP地址。 子节点：无 父节点：LiveRecord

示例

请求示例

```
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:pqgDBP8JXTXAytBoXpvNoZfo****
```

返回示例

```
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>
```

SDK

Java

8.10. PostVodPlaylist

PostVodPlaylist接口用于为指定的LiveChannel生成一个点播用的播放列表。OSS会查询指定时间范围内由该LiveChannel推流生成的ts文件，并将其拼装为一个m3u8播放列表。

请求语法

```
POST /ChannelName/PlaylistName?vod&endTime=EndTime&startTime=StartTime HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

请求参数

名称	类型	是否必选	描述
ChannelName	字符串	是	指定LiveChannel名称，该LiveChannel必须存在。
PlaylistName	字符串	是	指定生成的点播播放列表的名称，必须以“.m3u8”结尾。
StartTime	整数	是	指定查询ts文件的起始时间，格式为Unix timestamp。
EndTime	整数	是	指定查询ts文件的终止时间。 格式：Unix timestamp ? 说明 EndTime必须大于StartTime，且时间跨度不能大于1天。

示例

请求示例

```
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 YJhKOKWDWINLKXv:ABligvNltCHK+7fMHLeRIOUn****
```

返回示例

```
HTTP/1.1 200
content-length: 0
server: AliyunOSS
connection: close
etag: "9C6104DD9CF1A0C4D0CFD21F4390****"
x-oss-request-id: 57BE9A96B92475920B002359
date: Thu, 25 Aug 2016 07:13:26 GMT
```

SDK

Java

8.11. GetVodPlaylist

GetVodPlaylist接口用于查看指定LiveChannel在指定时间段内推流生成的播放列表。

请求语法

```
GET /ChannelName?vod&endTime=EndTime&startTime=StartTime HTTP/1.1
Date: GMT date
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Authorization: SignatureValue
```

请求头

名称	类型	是否必选	描述
ChannelName	字符串	是	指定LiveChannel名称。该LiveChannel必须存在。
StartTime	整数	是	指定查询ts文件的起始时间，格式为Unix timestamp。
EndTime	整数	是	指定查询ts文件的终止时间，格式为Unix timestamp。  说明 EndTime必须大于StartTime，且时间跨度不能大于1天。

示例

请求示例

```
GET /test-channel?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:ABligvnLtCHK+7fMHLeRIOUn****
```

返回示例

```
HTTP/1.1 200
content-length: 312
server: AliyunOSS
connection: close
etag: "9C6104DD9CF1A0C4D0CFD21F43905D59"
x-oss-request-id: 57BE9A96B92475920B002359
date: Thu, 25 Aug 2016 07:13:26 GMT
Content-Type: application/x-mpegURL
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-TARGETDURATION:13
#EXTINF:7.120,
1543895706266.ts
#EXTINF:5.840,
1543895706323.ts
#EXTINF:6.400,
1543895706356.ts
#EXTINF:5.520,
1543895706389.ts
#EXTINF:5.240,
1543895706428.ts
#EXTINF:13.320,
1543895706468.ts
#EXTINF:5.960,
1543895706538.ts
#EXTINF:6.520,
1543895706561.ts
#EXT-X-ENDLIST
```

SDK

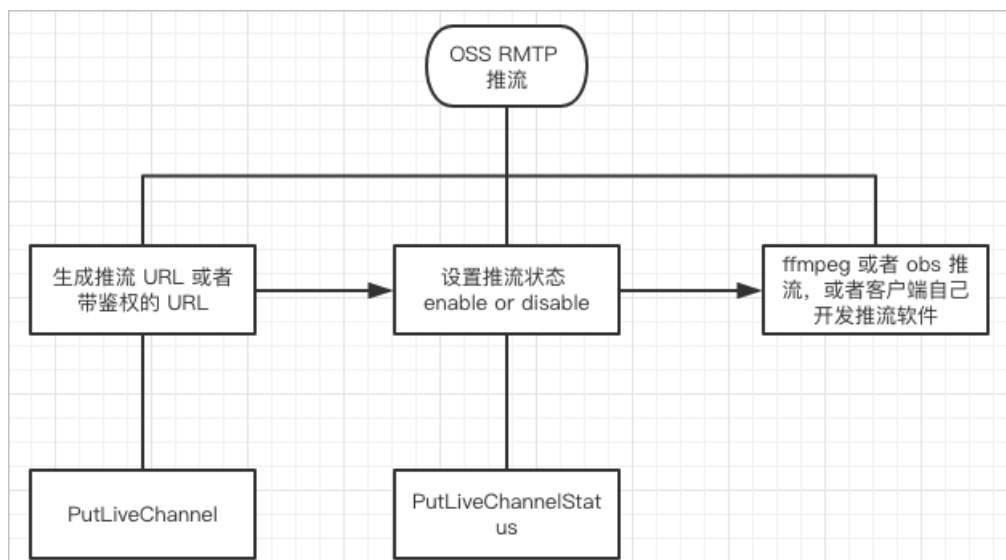
Java

8.12. LiveChannel常见问题

本文主要介绍使用LiveChannel过程中遇到的常见问题及解决方案。

OSS livechannel推流过程

下图为OSS livechannel推流过程图解，了解推流过程后，有助于排查livechannel推流过程中遇到的问题。



更多详情请参见[PutLiveChannel](#)、[PutLiveChannelStatus](#)。

案例：录制M3u8缺失

问题分析：默认录制成品M3u8只有最后3片，遵循了HLS协议的默认规则。

解决方案：使用PostVodPlaylist接口将指定时间范围内的ts文件汇聚到M3u8索引。

- EndTime 必须大于 StartTime，且时间跨度不能大于1天。
- OSS会查询指定时间范围内的所有指定LiveChannel推流生成的ts文件，并将其拼装为一个播放列表。

案例：录制的M3u8文件失败

问题分析：录制的M3u8文件成功推流到OSS才算录制成功。

解决方案：建议到客户端抓包查看是否有publish succeeds的标志，此标志表示与OSS音频包交互成功。所以出现客户端有推流记录，但是没有录制视频的情况，可以尝试抓包分析。

案例：客户端无法推流到OSS

使用ffmpeg推流不成功：

```
ffmpeg -re -i 0_20180525105430445.aac -acodec aac -strict -2 -f flv rtmp://xxx.oss-cn-beijing.aliyuncs.com/live/test_1000?Expires=1540458859&OSSAccessKeyId=LTAIujianb****&Signature=qwh31xQsanmao6ygCFJgo**%3D&playlistName=playlist.m3u8
```

解决方案：

- 使用ffmpeg推流不成功时，建议用最原始的命令推流，不添加复杂的参数。
- 推流URL中包含and符号（&）时，请使用双引号（"）将整个URL括起来。例如，`ffmpeg -re -i 0_20180525105430445.aac -acodec aac -strict -2 -f flv "rtmp://xxx.oss-cn-beijing.aliyuncs.com/live/test_1000?Expires=1540458859&OSSAccessKeyId=LTAIujianb****&Signature=qwh31xQsanmao6ygCFJgo****%3D&playlistName=playlist.m3u8"`。
- 尝试更换为OBS进行推流测试，否则无法确认是否因ffmpeg问题导致的推流失败。

如果推流后延时较长，可以在[PutLiveChannel](#)接口中调整FragDuration和FragCount参数。

案例：录制M3u8文件卡顿

当转储类型为HLS时，写入当前 ts 文件的音视频数据时长达到 `FragDuration` 指定的时长后，OSS会在收到下一个关键帧时切换到下一个 ts 文件。如果指定 `max(2*FragDuration, 60s)` 后仍未收到下一个关键帧，OSS将强制切换文件，此时可能引起播放卡顿。

案例：录制M3u8文件没有音频或视频

以下几种情况会出现音频或者视频没有录制的情况：

- 没有发送 AVC header 或者 AAC header，请通过抓包查看。
- RTMP message 长度小于2或者 sequence header 过小。
- 音频 Message 超过缓冲区大小。
- codec_ctx 是解码上下文的关键信息，如果携带的音视频数据异常，也会导致录制失败。

案例：ffmpeg推流到OSS录制没有音频

解决方案：

- 直接查看ffmpeg记录的日志，确定客户端是否已发送 aac_header。
- 在客户端进行RTMP抓包，查看客户端是否已发送 aac_header。

```
2018-10-31T00:37:13+08:00      Stream #0:0, 0, 1/1000
2018-10-31T00:37:13+08:00 : Video: h264 (High), 1 reference frame ([7][0][0][0] / 0x0007), yuv420p(progressive, left), 544x
2018-10-31T00:37:13+08:00 18 fps, 18 tbr, 1k tbn,
2018-10-31T00:37:13+08:00 1k tbc
2018-10-31T00:37:13+08:00      Stream #0:1, 0, 1/1000: Audio: aac (LC) ([10][0][0][0] / 0x000A), 48000 Hz, mono, fltp
2018-10-31T00:37:13+08:00 Stream mapping:
2018-10-31T00:37:13+08:00 (copy)
2018-10-31T00:37:13+08:00 cur_dts is invalid (this is harmless if it occurs once at the start per stream)
2018-10-31T00:37:14+08:00 Last message repeated 2 times
```