

腾讯云对象存储

XML API文档

产品文档



腾讯云

【版权声明】

©2015-2016 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

文档声明.....	2
简介	5
公共请求头部	7
公共返回头部	9
错误码	10
签名算法	15
Service接口	28
Get Service	28
Bucket接口	32
Get Bucket	32
Get Bucket ACL	38
Get Bucket CORS	42
Get Bucket Location	46
Get Buket Lifecycle	48
Get Bucket Tagging	53
Put Bucket	56
Put Bucket ACL	59
Put Bucket CORS	64
Put Bucket Lifecycle	68
Put Bucket Tagging	73
Delete Bucket	76
Delete Bucket CORS	78
Delete Bucket Lifecycle	80
Delete Bucket Tagging	82
Head Bucket	84
List Multipart Uploads	86
Object接口	94
Append Object	94
Get Object	99
Get Object ACL	102
Put Object	106
Put Object ACL	110

Delete Object	115
Delete Multiple Object	117
Head Object	121
Options Object	124
Initiate Multipart Upload	127
Upload Part	131
List Parts	134
Complete Multipart Upload	139
Abort Multipart Upload	143

简介

描述

对象存储的 XML API 是一种轻量级的，无连接状态的接口。您可以调用此套接口直接通过 http/https 发出请求和接受响应，从而实现与腾讯云对象存储（COS）后台进行交互操作。此套API的请求和响应内容都为XML格式。

在查阅其他的 API 文档之前，首先请详细阅读签名鉴权。

基本信息

本部分介绍是为了您更有效的使用 COS，而必须要了解的主要概念和术语。

名称	描述
AppID	开发者访问 COS 服务时拥有的用户维度唯一资源标识，用以标示资源。
SecretID	SecretID 是开发者拥有的项目身份识别 ID，用以身份认证
SecretKey	SecretKey 是开发者拥有的项目身份密钥。
Bucket	存储桶是 COS 中用于存储数据的容器，是用户存储在 Appid 下的第一级目录，每个对象都存储在一个存储桶中。
Object	对象是 COS 中存储的具体文件，是存储的基本实体。
Region	域名中的地域信息，枚举值：cn-east（华东），cn-north（华北），cn-south（华南），sg（新加坡）

快速入门

要使用对象存储 API，需要先执行以下步骤：

1. 获取 Appid、SecretID、SecretKey 内容
2. 编写一个 签名鉴权 算法程序（或使用任何一种服务端 SDK）
3. 计算签名，调用 API 执行操作

其他版本API

JSON API

JSON API即2016年10月之后，用户接入COS使用的API，上传域名为[Region].file.myqcloud.com。JSON API接口日后将保持维护状态，可以正常使用但是不发展新特性。其与标准XML的API底层架构相同，数据互通，可以交叉使用，但是接口不兼容，域名不一致。[文档链接](#)

历史版本API

历史版本API即2016年10月之前，用户接入COS使用的API，上传域名为web.file.myqcloud.com。使用该版本接口的用户，COS将主动帮助用户缓慢迁移数据，并转发请求，用户若无新功能使用诉求，无需主动更换接口。其与标准XML的API 底层架构不同，数据不互通，接口不兼容。[文档链接](#)

公共请求头部

描述

此篇文档将为您介绍在使用API时候会使用到的公共请求头部（Request Header），下文提到的头部会在之后的具体API文档中不再赘述。

请求头部列表

Header名称	描述	类型	必选
Authorization	携带鉴权信息，用以验证请求合法性的签名信息。针对公有读的文件，无需携带此头部。	String	否
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节），常用于PUT类型的API的操作。	String	否
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型（MIME），例如text/plain	String	是
Content-MD5	RFC 1864 中定义的经过Base64编码的128-bit 内容MD5 校验值。此头部用来校验文件内容是否发生变化。	String	否
Date	RFC 1123 中定义的 GMT 时间，例如 Wed, 30 Mar. 2016 23:00:00 GMT。	String	否
Expect	当使用 Expect: 100-continue 时，在收到服务端确认后，才会发送请求内容。该选项可以被用于验证头部是否有效，而无需	String	否

Header名称	描述	类型	必选
	发送数据内容。 有效值：100-continue。		
Host	请求的主机，形式为[BucketName]-[AppID].[Region].myqcloud.com	String	是
x-cos-content-sha1	自定义的经过SHA1编码的160-bit 内容校验值。此头部用来校验文件内容是否发生变化，会在签名中使用。	String	否

公共返回头部

描述

此篇文档将为您介绍在使用API时候会出现的公共返回头部（Response Header），下文提到的头部会在之后的具体API文档中不再赘述。

返回头部列表

Header名称	描述	类型
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）	String
Content-Type	RFC 2616 中定义的 HTTP 请求内容类型（MIME）	String
Connection	声明客户端与服务端之间的通信状态。 枚举值：keep-alive，close	Enum
Date	服务器端的响应时间，根据 RFC 1123 中定义的 GMT 时间。	String
Etag	ETag 全称 Entity Tag 是 Object 被创建时用于标识 Object 内容的信息标签。此参数并不一定返回 MD5 值，请根据不同请求的情况参考。ETag 的值可以用于检查 Object 的内容是否发生变化。	String
Server	创建请求的服务器的名称，默认值：tencent-cos	String
x-cos-request-id	每次请求发送时，服务端将会自动为请求生成一个ID。	String
x-cos-trace-id	每次请求出错时，服务端将会自动为这个错误生成一个ID。	String

错误码

概述

此文将为您介绍请求出错时返回的错误码和对应错误信息。

错误信息返回格式

返回头部

Content-Type : application/xml

对应HTTP状态码：3XX，4XX，5XX

返回内容

语法格式

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>[错误码]</Code>
  <Message>[错误信息]</Message>
  <Resource>[资源地址]</Resource>
  <RequestId>[请求ID]</RequestId>
  <TraceId>[错误ID]</TraceId>
</Error>
```

元素说明

元素名称	描述	类型
Error	包含所有的错误信息。	Container
Code	错误码用来定位唯一的错误条件，用来确定错误场景，具体错误码见下文	String

元素名称	描述	类型
	。	
Message	包含具体的错误信息。	String
Resource	资源地址：Bucket地址或者Object地址。	String
RequestId	当请求发送时，服务端将会自动为请求生成一个唯一的 ID。使用遇到问题时，request-id能更快地协助COS 定位问题。	String
TraceId	当请求出错时，服务端将会自动为这个错误生成一个唯一的 ID。使用遇到问题时，trace-id能更快地协助COS 定位问题。当请求出错时，trace-id与request-id一一对应。	String

错误码列表

3XX类型错误

错误码	描述	HTTP状态码
PermanentRedirect	该资源已经被永久改变了位置，请使用HTTP Location来重定向到正确的新位置	301 Moved Permanently
TemporaryRedirect	该资源已经被临时改变了位置，请使用HTTP Location来重定向到正确的新位置	302 Moved Temporarily
Redirect	临时重定向	307 Moved Temporarily
TemporaryRedirect	在DNS更新期间，您将被临时重定向	307 Moved Temporarily

4XX类型错误

错误码	描述	HTTP状态码
BadDigest	提供的x-cos-SHA-1值与服务端收到的文件SHA-1值不符合	400 Bad Request

错误码	描述	HTTP状态码
EntityTooSmall	上传的文件大小 不足要求的最小值，常见于分片上传	400 Bad Request
EntityTooLarge	上传的文件大小超过要求的最大值	400 Bad Request
IncompleteBody	请求的实际内容长度和指定的Content-Length不符	400 Bad Request
IncorrectNumberOfFilesInPostRequest	Post请求每次只允许上传一个文件	400 Bad Request
InlineDataTooLarge	内链数据大小高于要求的最大值	400 Bad Request
InvalidArgument	请求参数不合法	400 Bad Request
InvalidBucketName	Bucket名称不合法	400 Bad Request
InvalidDigest	x-cos-SHA-1值不合法	400 Bad Request
InvalidPart	分片缺失或者SectionID出错	400 Bad Request
InvalidPolicyDocument	策略配置文件不合法	400 Bad Request
InvalidURI	URI不合法	400 Bad Request
KeyTooLong	文件路径过长	400 Bad Request
MalformedACLError	描述的ACL策略不符合XML语法	400 Bad Request
MalformedPOSTRequest	该POST请求的Body内容不合法	400 Bad Request
MalformedXML	body的XML格式不符合XML语法	400 Bad Request
MaxMessageLengthExceeded	请求过长	400 Bad Request
MaxPostPreDataLengthExceededError	该POST请求的数据前缀过长，常见于分片上传	400 Bad Request
MatadataTooLarge	元数据大小超过要求的最大值	400 Bad Request
MissingRequestBodyError	请求Body缺失	400 Bad Request
MissingSecurityHeader	必要Header缺失	400 Bad Request
MissingContentMD5	请求头中缺少Content-MD5	400 Bad Request
MissingAppid	请求头中缺少Appid	400 Bad Request
MissingHost	请求头中缺少Host	400 Bad Request
RequestIsNotMultiPartContent	Post请求 Content-Type不合法	400 Bad Request
RequestTimeout	读取数据超时，检查网络是否过慢或上传并发数过大	400 Bad Request
TooManyBucket	bucket数量超过200限制	400 Bad Request
UnexpectedContent	请求不支持相关内容	400 Bad Request
UnresolvableGrantByUID	提供的UID不存在	400 Bad Request

错误码	描述	HTTP状态码
UserKeyMustBeSpecified	针对Bucket的Post操作必须指定明确路径	400 Bad Request
AccessDenied	签名或者权限不正确，拒绝访问	403 Forbidden
AccooountProblem	您的账号拒绝了此次操作	403 Forbidden
InvaildAccessKeyId	AccessKey不存在	403 Forbidden
InvalidObjectState	请求内容与Object属性相冲突	403 Forbidden
InvalidSecurity	签名串不合法	403 Forbidden
RequestTimeTooSkewed	请求时间超过权限有效时间	403 Forbidden
SignatureDoesNotMatch	提供的签名不符合规则	403 Forbidden
NoSuchBucket	指定的Bucket不存在	404 Not Found
NoSuchUpload	指定的分片上传不存在	404 Not Found
NoSuchBucket	指定的Bucket策略不存在	404 Not Found
MethodMotAllowed	此资源不支持该HTTP方法	405 Method Not Allowed
BucketAlreadyExists	CreateBucket指定的BucketName已经使用，请选择新的BucketName	409 Conflict
BucketNotEmpty	DeleteBucket前请先删除文件和未完成的分片上传任务	409 Conflict
InvalidBucketState	bucket状态与操作请求冲突，比如多版本管理与跨区域复制的冲突	409 Conflict
OperationAborted	指定资源不支持此类操作	409 Conflict
MissingContentLength	Header Content-Length缺失	411 Length Required
PreconditionFailed	前置条件匹配失败	412 Precondition
InvalidRange	请求的文件范围不合法	416 Requested Range Not Satisfiable
InvalidSHA1Digest	请求内容sha1校验不合法	400 Bad Request
NoSuchUpload	分块上传时指定的uploadid不存在	400 Bad Request
InvalidPart	分块缺失	400 Bad Request
InvalidPartOrder	分块上传编号不连续	400 Bad Request
ObjectNotAppendable	指定的文件不能追加	400 Bad Request
AppendPositionErr	Append:文件长度和position不一致	400 Bad Request
NoSuchVersion	指定版本不存在	400 Bad Request
NoLifecycle	生命周期不存在	400 Bad Request
PreconditionFailed	前置条件匹配失败	400 Bad Request

错误码	描述	HTTP状态码
UnexpectedContent	请求不支持相关内容	400 Bad Request
MultiBucketNotSupport	跨区域复制只能设一个目的bucket	400 Bad Request
NotSupportedStorageClass	指定的存储类型不合法	400 Bad Request
InvalidAccessKeyId	AccessKey不存在	403 Forbidden
ExpiredToken	签名串已过期	403 Forbidden

5XX类型错误

错误码	描述	HTTP状态码
InternalServerError	服务端内部错误	500 Internal Server
NotImplemented	Header中存在无法处理的方法	501 Not Implemented
ServiceUnavailable	服务器内部错误，请重试	503 Service Unavailable
SlowDown	请降低访问频率	503 Slow Down

其他类型错误

错误码	描述	HTTP状态码
InvaildAddressingHeader	必须使用匿名角色访问	N/A

签名算法

签名描述

使用对象存储服务 COS 时，发起的 HTTP 的请求分为 匿名请求 和 签名请求。签名请求根据腾讯云提供的密钥，并结合请求的内容计算出一串密文，通过 HTTP 请求传入。

- 匿名请求：请求不携带任何身份标识和鉴权信息，直接通过对 COS 的地址进行 PUT/GET/POST/DELET 等 HTTP 操作。
- 签名请求：通过在 HTTP Headers 中的 Authorization 字段或通过 URL 参数的方式传入加密后的签名信息，COS 服务端将会判断该签名请求是否被许可。

通常在部署公共访问服务时，可以将存储设置为允许所有匿名访问，而针对写入请求一般都需要通过签名保护。COS 可以根据用户指定的 ACL 策略，与传入的签名进行判断后，决定应当许可何种请求、如何控制请求内容、控制请求有效期等。

限制请求并使用签名访问，常见于以下场景：

校验使用者身份：计算签名的过程依赖腾讯云提供的密钥，包含了 SecretID 和 SecretKey 等元素。

校验传输的数据：数据的校验信息包含在签名的密文信息中，若传入数据的校验值与密文中的校验值不匹配，则请求将返回失败。通常这能保障请求内容被劫持时，错误的数据内容不会被记录。

保护签名不被二次使用：通过在签名中加密请求的有效时间，可以确保在签名过期后客户端无法发起请求，这也用于确保网络被监听时，第三方无法介入重复使用签名以破坏数据安全性。

签名方法

COS 的签名请求使用以下两种方式包含在 HTTP 请求中：

- HTTP Authorization Header：通过 HTTP 请求头部中的 Authorization 字段传入。

- URL 参数：通过 URL 后携带参数的方式传入签名。

签名算法 - HTTP 头部

概述

除了 POST 操作以外，其他的签名都可以通过 HTTP 请求头部中的 Authorization 字段传入，这也是 HTTP 标准定义中最常见的请求验证方式。常见的请求如下：

PUT ?versioning HTTP/1.1

Host: testbucket-125000000.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=QmFzZTY0IGlzIGZ2VuZXJp&q-sign-time=1480932292;1481012292&q-key-time=1480932292;1481012292&q-header-list=host&q-url-param-list=versioning&q-signature=43803ef4a4207299d87bb75d1c739c06cc9406bb

以下表格中，列出了 Authorization 字段需要传入的信息：

名称	描述
q-sign-algorithm	描述该签名使用的加密方式，目前腾讯云使用的是 HMAC-SHA1 的方式加密签名。 该字段请保持默认值：sha1
q-ak	用于标识用户身份，SecretID 的字段，在腾讯云的 API 密钥页面中可以查看。
q-sign-time	签名的有效起止时间，其使用 10 位 Unix 时间戳来表示，有效效力精确到秒。 该字段通过分号区分起止，起时在前、止时在后。
q-key-time	用户可以自定义 SigningKey 有效时间，使用 10 位 Unix 时间戳来表示，有效效力精确到秒。 该字段通过分号区分起止，起始时间在前、终止时间在后。 一般 q-key-time 的时间范围大于等于 q-sign-time。
q-header-list	提供密文中包含需要校验的 Headers 列表，必须是小写字符，且需要按字典序排序，以";"分隔

名称	描述
q-url-param-list	提供密文中包含需要校验的 Parameters 列表，必须是小写字符，以";"分隔
q-signature	经过 HMAC-SHA1 算法加密的请求校验信息。

计算签名

在构造签名前，必须先获取腾讯云账号的 API 密钥对，其中包含 SecretID 和 SecretKey，详情参看 COS 控制台页面左侧的「密钥管理」或腾讯云控制台中的「监控与管理 - 云 API 密钥」。

签名构成

- SignKey：携带有效时间，并通过 SecretKey 进行 HMAC-SHA1 加密的密钥串。
- FormatString：将请求经过一定规范格式化后的字符串。
- StringToSign：包含校验算法、请求有效时间和 Hash 校验后的 FormatString 的字符串。
- Signature：加密后的签名，使用 SignKey 与 StringToSign 通过 HMAC-SHA1 加密的字符串，填入 q-signature。

计算 SignKey

为了保障 SecretKey 的安全，请求需要对 SecretKey 进行加密后传输，签名允许用户通过携带 Unix 时间戳的方式限制 SecretKey 的有效使用时间。

为了让用户可以将计算签名所需的密钥信息下发到不信任的客户端，SecretKey 支持与用户指定的有效时间一起，一并加密生成一个不可逆的密钥串，下发给不被信任的客户端使用。计算 SignKey 的格式如下：

\$SignKey =

HMAC-SHA1(\$SecretKey,"<q-key-time>")

- SecretKey：来自腾讯云提供的 API 密钥对中的 SecretKey，例如

AKIDZfbOA78asKUYBcXFrJD0a1ICvR98JM

。

- q-key-time：包含 SignKey 的起始和终止有效时间，前后两个时间戳用分号分开，使用 10 位 Unix 时间戳，精确到秒。例如

1480932292;1481012292

。

构成 FormatString

该字符串将 HTTP 请求中的关键信息进行格式化处理，并将用于作为加密签名校验的主要部分。这可以保障 HTTP 请求在被传输的过程中，信息不会被第三方篡改。

为了让 COS 的服务端可以按照固定格式来校验请求，HTTP 请求中的关键数据都需要包含在 Format String 中，采用换行的方法陈列数据，每行一个关键要素。生成的方法如下：

\$FormatString =

<FormatMethod>\n

<FormatURI>\n

<FormatParameters>\n

<FormatHeaders>\n

- FormatMethod：指该请求的 HTTP 操作行为，例如 PUT/GET/DELETE，必须转为小写字符。
 - 例如发起

GET http://testbucket-125000000.cn-north.myqcloud.com

其 FormatMethod 为

get

- FormatURI：指该请求中的 URI 部分，即除去 http:// 协议和域名的部分（通常以 / 开始），并且不包含 URL 中的参数部分（通常以 ? 开始）。
 - 例如访问地址

`http://testbucket-1250000000.cn-north.myqcloud.com/testfile`

其 Format URI 为

`/testfile`

- FormatParameters：指该请求中的参数部分（以 ? 开始的部分），用

`key=value`

的方式表达。参数的 key 和 value 都必须经过 URL Encode，如果有多个参数对可使用 & 连接，Key和Value必须转为小写字符，且key值按字典序排序。

- 例如访问

`http://testbucket-1250000000.cn-north.myqcloud.com/testfile?versioning`

其 FormatParameters 为

`versioning=`

- 例如访问

`http://testbucket-1250000000.cn-north.myqcloud.com/?prefix=abc&max-keys=20`

其 FormatParameters 为

`max-keys=20&prefix=abc`

- FormatHeaders：指请求中的 HTTP 头部信息，用

key=value

的方式表达。头部的 key 必须全部小写，value 必须经过 URL Encode。如果有多个参数对可使用 & 连接。key值按字典序排序

- 例如头部

Host: testbucket-125000000.cn-north.myqcloud.com

其 FormatHeaders 为

host=testbucket-125000000.cn-north.myqcloud.com

计算 StringToSign

该字符串包含了签名的算法名称，签名的有效时间，和 SHA-1 哈希后的 FormatString，因此需要计算 StringToSign 必须先生成好 FormatString。

StringToSign 中使用到的签名有效时间与 SignKey 中的 SecretKey 有效使用时间概念不同，这里的有效时间仅用于校验请求是否在有效的时间内被发起。

- 如果发起请求的客户端是可信的，一般可以直接将 SecretKey 保存在客户端，并且可以在 SignKey 和 StringToSign 中使用相同的有效起止时间来保障请求的有效性。
- 如果发起请求的客户端默认不可信，则需要保护 SecretKey 不能直接将其保存在客户端，此时便可以通过对 SecretKey 进行起止时间限制和加密后，下发 SignKey 给客户端以用于计算签名，而 StringToSign 中的有效时间应当在请求发起时由客户端生成。

生成 StringToSign 的格式如下：

\$StringToSign =

<q-sign-algorithm>\n

<q-sign-time>\n

SHA1Hash(\$FormatString)\n

- q-sign-algorithm：签名使用的加密算法，默认填

sha1

。

- q-sign-time：该请求的起始和终止有效时间，前后两个时间戳用分号分开，使用 10 位 Unix 时间戳，精确到秒。例如

1480932292;1481012292

。

- SHA1Hash(\$FormatString)：将构成 FormatString 部分的字符串，使用 SHA-1 算法哈希得到一个不可逆的字符串，以用于标识请求的关键内容。

计算签名 Signature

此步生成的 Signature 将被放置在 q-signature 字段，用于校验请求的内容是否合法。其使用到 HMAC-SHA1 算法，使用 SignKey 作为密钥对 StringToSign 进行加密计算。生成 Signature 的格式如下：

\$Signature =

HMAC-SHA1(\$SignKey,\$StringToSign)

生成 Authorization

将计算签名步骤生成的内容与请求中需要明文标识的内容，用

key=value

的方式表达，多个参数对使用 & 连接。

生成 Authorization 的格式如下（一个长字符串，不含换行符）：

q-sign-algorithm=sha1&
q-ak= <SecretID> &
q-sign-time= <SignTime> &
q-key-time= <KeyTime> &
q-header-list= <SignedHeaderList> &
q-url-param-list= <SignedParameterList> &
q-signature= <Signature>

示例

COS 将提供一些示例来演示签名生成的过程和最终结果，您也可以示例中的案例来验证您的签名代码是否正确地计算了结果。示例使用了如下密钥对：

SecretID	SecretKey
QmFzZTY0IGlzIGEgZ2VuZXJp	AKIDZfbOA78asKUYBcXFrJD0a1ICvR98JM

需要生成的密钥和请求起止时间均为：1480932292 和 1481012292。

测试使用的 Bucket 是在中国华北地区的 testbucket，建立在开发商 AppID 为 125000000 的账户下。

因此示例中发起的 Host 均为

testbucket-125000000.cn-north.myqcloud.com

。

GET Object 示例

以下示例内容展示了一个 GET Object 操作的签名示例，地址是

http://testbucket-125000000.cn-north.myqcloud.com/testfile

并通过 Range 参数请求该文件的前 4 个字节。

GET /testfile HTTP/1.1

Host: testbucket-125000000.cn-north.myqcloud.com

Range: bytes=0-3

步骤一：计算 SignKey

HMAC-SHA1(AKIDZfbOA78asKUYBcXFrJD0a1ICvR98JM,"1480932292;1481012292")

\$SignKey = 95d110a8ead64cac52083100db75b7e3f369e72f

步骤二：构成 FormatString

注意尾行仍然需要使用 \n 换行。

get

/testfile

host=testbucket-125000000.cn-north.myqcloud.com&range=bytes%3d0-3

为便于步骤三中需要的 SHA1Hash(\$FormatString) 使用，这里先对内容做哈希，得到结果：

c92f7246e3f922fe4abae5d6d5ebcd2397dc88cb

步骤三：计算 StringToSign

注意尾行仍然需要使用 \n 换行。

sha1

1480932292;1481012292

c92f7246e3f922fe4abae5d6d5ebcd2397dc88cb

步骤四：计算 Signature

HMAC-SHA1(95d110a8ead64cac52083100db75b7e3f369e72f,\$StringToSign)

得到结果

29b2f454bb9d8a629e7cad61227bd5fd0dd11a2d

步骤五：构造 Authorization

q-sign-

algorithm=sha1&q-

ak=QmFzZTY0IGlzIGegZ2VuZXJp&q-sign-time=1480932292;1481012292&q-key-time=1480932292;1481012292&q-header-list=host;range&q-url-param-list=&q-signature=29b2f454bb9d8a629e7cad61227bd5fd0dd11a2d

最终的请求示例

GET /testfile HTTP/1.1

Host: testbucket-125000000.cn-north.myqcloud.com

Range: bytes=0-3

Authorization: q-sign-algorithm=sha1&q-ak=QmFzZTY0IGlzIGegZ2VuZXJp&q-sign-time=1480932292;1481012292&q-key-time=1480932292;1481012292&q-header-list=host;range&q-url-param-list=&q-signature=29b2f454bb9d8a629e7cad61227bd5fd0dd11a2d

PUT Object 示例

以下示例内容展示了一个 PUT Object 操作的签名示例，地址是

`http://testbucket-125000000.cn-north.myqcloud.com/testfile2`

, 并通过 SHA-1 校验文件内容 "HelloWorld" 哈希为

`db8ac1c259eb89d4a131b253bacfca5f319d54f2`

, 指定该文件放在近线存储介质。

`PUT /testfile2 HTTP/1.1`

`Host: testbucket-125000000.cn-north.myqcloud.com`

`x-cos-content-sha1: db8ac1c259eb89d4a131b253bacfca5f319d54f2`

`x-cos-stroage-class: nearline`

`<Payload>`

步骤一：计算 SignKey

`HMAC-SHA1(AKIDZfbOA78asKUYBcXFrJD0a1ICvR98JM,"1480932292;1481012292")`

`$SignKey = 95d110a8ead64cac52083100db75b7e3f369e72f`

步骤二：构成 FormatString

注意尾行仍然需要使用 \n 换行。

`put`

`/testfile2`

`host=testbucket-125000000.cn-north.myqcloud.com&x-cos-content-`

`sha1=db8ac1c259eb89d4a131b253bacfca5f319d54f2&x-cos-stroage-class=nearline`

为便于步骤三中需要的 SHA1Hash(\$FormatString) 使用，这里先对内容做哈希，得到结果：

```
c3aa791042f601c81e8453dbb05472de8242576d
```

步骤三：计算 StringToSign

注意尾行仍然需要使用 \n 换行。

```
sha1
1480932292;1481012292
c3aa791042f601c81e8453dbb05472de8242576d
```

步骤四：计算 Signature

```
HMAC-SHA1(95d110a8ead64cac52083100db75b7e3f369e72f,$StringToSign)
```

得到结果

```
b237c36c5495b048519b82b17a200840594c0339
```

步骤五：构造 Authorization

```
q-sign-
algorithm=sha1&q-
ak=QmFzZTY0IGlzIGZ2VuZXJp&q-sign-time=1480932292;1481012292&q-key-
time=1480932292;1481012292&q-header-list=host;x-cos-content-sha1;x-cos-storage-class&q-url-
param-list=&q-signature=b237c36c5495b048519b82b17a200840594c0339
```

最终的请求示例

PUT /testfile2 HTTP/1.1

Host: testbucket-125000000.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=QmFzZTY0IGlzIGegZ2VuZXJp&q-sign-time=1480932292;1481012292&q-key-time=1480932292;1481012292&q-header-list=host;x-cos-content-sha1;x-cos-storage-class&q-url-param-list=&q-signature=b237c36c5495b048519b82b17a200840594c0339

x-cos-content-sha1: db8ac1c259eb89d4a131b253bacfca5f319d54f2

x-cos-stroage-class: nearline

HelloWorld

Service接口

Get Service

功能描述

Get Service接口实现获取该用户下所有Bucket列表。该API接口需要使用Authorization签名认证，且只能获取签名中AccessID所属账户的Bucket列表。

请求

请求语法

GET / HTTP 1.1

Host:service.cos.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
ListAllMyBucketsResult	说明本次返回的所有信息	Container
Owner	说明Bucket所有者的信息 父节点：ListAllMyBucketsResult	Container
UIN	Bucket所有者的UIN 父节点：ListAllMyBucketsResult.Owner	String
Buckets	说明本次返回的Bucket列表的所有信息 父节点：ListAllMyBucketsResult	Container
Bucket	单个Bucket的信息 父节点：ListAllMyBucketsResult.Buckets	Container
Name	Bucket名称 父节点：ListAllMyBucketsResult.Buckets.Bucket	String
Location	Bucket所在区域，枚举值：china-east, china-south, china-north, china-southwest	String
CreateDate	Bucket创建时间，ISO8601格式，例如 2016-11-09T08:46:32.000Z 父节点：ListAllMyBucketsResult.Buckets.Bucket	Date

<ListAllMyBucketsResult>

<Owner>

<UIN> </UIN>

</Owner>

<Buckets>

<Bucket>

<Name> </Name>

```
<Location> </Location>
<CreateDate> </CreateDate>
</Bucket>
...
</Buckets>
</ListAllMyBucketsResult>
```

示例

请求

GET / HTTP/1.1

Host:service.cos.myqcloud.com

Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1489110340;32468694340&q-key-time=1489110340;32562006340&q-header-list=host&q-url-param-list=&q-signature=cb46d5ce6daed2d3dc0db7130a57193497605620

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 19935

Connection: keep-alive

Date: Fri Mar 10 09:45:46 2017

Server: tencent-cos

x-cos-request-id: NThjMjA1NGFfNTViMjM1XzI0NWRfMjA4OGIx

```
<ListAllMyBucketsResult>
  <Owner>
    <uin>2779643970</uin>
  </Owner>
  <Buckets>
```

```
<Bucket>
<Name>01</Name>
<Location>china-south</Location>
<CreateDate>2016-09-13 15:20:15</CreateDate>
</Bucket>
<Bucket>
<Name>0111</Name>
<Location>china-south</Location>
<CreateDate>2017-01-11 17:23:51</CreateDate>
</Bucket>
<Bucket>
<Name>1201new</Name>
<Location>china-south</Location>
<CreateDate>2016-12-01 09:45:02</CreateDate>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

Bucket接口

Get Bucket

功能描述

Get Bucket请求等同于 List

Object请求，可以列出该Bucket下部分或者所有Object，发起该请求需要拥有Read权限。

请求

请求语法

GET / HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: auth

请求参数

名称	描述	类型
prefix	前缀匹配，用来规定返回的文件前缀地址	String
delimiter	定界符为一个符号，如果有Prefix，则将Prefix到delimiter之间的相同路径归为一类，定义为Common Prefix，然后列出所有Common Prefix。如果没有Prefix，则从路径起点开始	String
encoding-type	规定返回值的编码方式	String
marker	默认以UTF-8二进制顺序列出条目，所有列出条目从marker开始	String
max-keys	单次返回最大的条目数量，默认1000	String

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
Name	Bucket名字 父节点：ListBucketResult	String
Prefix	前缀匹配，用来规定返回的文件前缀地址 父节点：ListBucketResult	String
Marker	默认以UTF-8二进制顺序列出条目，所有列出条目从marker开始 父节点：ListBucketResult	String
Maxkeys	单次返回最大的条目数量 父节点：ListBucketResult	String
IsTruncated	返回条目是否被截断，布尔值：True， False 父节点：ListBucketResult	Boolean
NextMarker	假如返回条目被截断，则返回Next Marker就是下一个条目的起点 父节点：ListBucketResult	String

名称	描述	类型
CommonPrefixes	将Prefix到delimiter之间的相同路径归为一类，定义为Common Prefix 父节点：ListBucketResult	String
Encoding-Type	编码类型，作用于Delimiter，Marker，Prefix，NextMarker，Key 父节点：ListBucketResult	String
Content	元数据信息 父节点：ListBucketResult	Container
Key	Object名称 父节点：ListBucketResult.Contents	String
LastModified	Object最后修改时间 父节点：ListBucketResult.Contents	Date
Etag	文件的 SHA-1 算法校验值 父节点：ListBucketResult.Contents	String
Size	文件大小，单位Byte 父节点：ListBucketResult.Contents	String
Owner	Bucket所有者信息 父节点：ListBucketResult.Contents	Container
StorageClass	Object的存储级别，枚举值：Standard，Standard_IA，Nearline 父节点：ListBucketResult.Contents.Owner	String
ID	Bucket的AppID 父节点：ListBucketResult.Contents.Owner	String
CommonPrefixes.Prefix	单条Common Prefix 父节点：CommonPrefixes	String

<ListBucketResult>

```
<Name> </Name>
<Prefix> </Prefix>
<Marker> </Marker>
<MaxKeys> </MaxKeys>
<IsTruncated> </IsTruncated>
<Contents>
<Key> </Key>
<LastModified> </LastModified>
<ETag> </ETag>
<Size> </Size>
<Owner>
<ID> </ID>
</Owner>
<StorageClass> </StorageClass>
</Contents>
<CommonPrefixes>
<Prefix> </Prefix>
</CommonPrefixes>
</ListBucketResult>
```

示例

请求

GET / HTTP/1.1

Host: zuhaotestnorth-1251668577.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484213451;32557109451&q-key-time=1484213451;32557109451&q-header-list=host&q-url-param-list=&q-signature=0336a1fc8350c74b6c081d4dff8e7a2db9007dce

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 1132

Connection: keep-alive

Vary: Accept-Encoding

Date: Thu Jan 12 17:30:54 2017

Server: tencent-cos

x-cos-request-id: NTg3NzRjY2VfYmRjMzVfMTc5M182MmIyNg==

<?xml version='1.0' encoding='utf-8' ?>

<ListBucketResult>

<Name>zuhaotestnorth</Name>

<Prefix/>

<Marker/>

<MaxKeys>1000</MaxKeys>

<Delimiter>/</Delimiter>

<IsTruncated>>false</IsTruncated>

<Contents>

<Key>testL</Key>

<LastModified>Wed Jan 11 18:57:06 2017</LastModified>

<ETag>"79f2a852fac7e826c9f4dbe037f8a63b"</ETag>

<Size>10485760</Size>

<StorageClass>Standard</StorageClass>

</Contents>

<Contents>

<Key>testL1</Key>

<LastModified>Wed Jan 11 19:02:17 2017</LastModified>

<ETag>"3f9a5dbff88b25b769fa6304902b5d9d"</ETag>

<Size>10485760</Size>

<StorageClass>Standard</StorageClass>

</Contents>

<Contents>

<Key>testLLL</Key>

<LastModified>Wed Jan 11 16:36:08 2017</LastModified>

```
<ETag>"39bfb88c11c65ed6424d2e1cd4db1826"</ETag>
<Size>10485760</Size>
<StorageClass>Standard</StorageClass>
</Contents>
<Contents>
<Key>testLOL</Key>
<LastModified>Wed Jan 11 17:24:10 2017</LastModified>
<ETag>"fb31459ad10289ff49327fd91a3e1f6a"</ETag>
<Size>4</Size>
<StorageClass>Standard</StorageClass>
</Contents>
<Contents>
<Key>tet</Key>
<LastModified>Wed Jan 11 15:54:02 2017</LastModified>
<ETag>"83b3ec25cc19626ac073297eba30fbc4"</ETag>
<Size>10485760</Size>
<StorageClass>Standard</StorageClass>
</Contents>
</ListBucketResult>
```

Get Bucket ACL

功能描述

使用API读取Bucket的ACL表，只有所有者有权操作。

请求

请求语法

GET /?acl Http/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

必选头部

参数名称	描述	类型	必选
Authorization	签名串	String	是

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

参数名称	描述	类型
AccessControlPolicy	一条独立的ACL记录	Container
Owner	标识资源的所有者	Container
uin	用户QQ号	String
Subaccount	子账户QQ账号	String
AccessControlList	被授权者信息与权限信息	Container
Grant	单条授权信息，一个AccessControlList可以拥有100条Grant	Container
Grantee	被授权者资源信息，type类型可以为RootAccount，SubAccount；当type类型为RootAccount时，可以在UIN中填写QQ，也可以填写anonymous（指代所有类型用户）。当type类型为RootAccount时，UIN代表根账户账号，SubAccount代表子账户账号	Container
Permission	权限信息，枚举值：READ，WRITE，FULL_CONTROL	String

```

<AccessControlPolicy>
  <Owner>
    <uin>ID</uin>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee type="SubAccount">
        <uin>ID</uin>
        <Subaccount> SUBID </Subaccount>
      </Grantee>
      <Permission>Permission</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>

```

```
<Grant>
<Grantee type="RootAccount">
<uin>ID</uin>
</Grantee>
<Permission>Permission</Permission>
</Grant>
<Grant>
...
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

示例

请求

GET /?acl HTTP/1.1

Host: zuhaotestnorth-1251668577.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484213027;32557109027&q-key-time=1484213027;32557109027&q-header-list=host&q-url-param-list=acl&q-signature=dcc1eb2022b79cb2a780bf062d3a40e120b40652

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 266

Connection: keep-alive

Date: Thu Jan 12 17:23:49 2017

Server: tencent-cos

x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==


```
<AccessControlPolicy>
  <Owner>
    <uin>2779643970</uin>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee type="RootAccount">
        <uin>2779643970</uin>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Get Bucket CORS

功能描述

Get Bucket CORS实现跨域访问配置读取。

请求

请求语法

GET /?cors HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型	必选
CORSConfiguration	说明跨域配置的所有信息，最多可以包含100条CORSRule	Container	是
CORSRule	单条配置的信息 父节点：CORSRule	Container	是
ID	规则名称，可选填 父节点：CORSRule	String	否
AllowedMethod	允许的HTTP操作，枚举值：Get，Put，Head，Post，Delete 父节点：CORSRule	Enum	是
AllowedOrigin	允许的访问来源，支持『*』通配符 父节点：CORSRule	String	是
AllowedHeader	在发送 OPTIONS 请求时告知服务端，接下来的请求可以使用哪些自定义的 HTTP 请求头部 父节点：CORSRule	String	否
MaxAgeSeconds	设置 OPTIONS 请求得到结果的有效期 父节点：CORSRule	Integer	否
ExposeHeadr	设置浏览器可以接收到的来自服务器端的自定义头部信息 父节点：CORSRule	String	否

<CORSConfiguration>

<CORSRule>

<ID> </ID>

<AllowedOrigin> </AllowedOrigin>

<AllowedMethod> </AllowedMethod>

```
<AllowedHeader> </AllowedHeader>
<MaxAgeSeconds> </MaxAgeSeconds>
<ExposeHeader> </ExposeHeader>
</CORSRule>
<CORSRule>
...
</CORSRule>
...
</CORSConfiguration>
```

示例

请求

```
GET /?cors HTTP/1.1
Host:arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-
time=1484815944;32557711944&q-key-time=1484815944;32557711944&q-header-list=host&q-url-
param-list=cors&q-signature=a2d28e1b9023d09f9277982775a4b3b705d0e23e
```

返回

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 345
Connection: keep-alive
Date: Thu Jan 19 16:52:31 2017
Server: tencent-cos
x-cos-request-id: NTg4MDdlNGZfNDYyMDRIXzM0YWFFZTBh
```

```
<CORSConfiguration>
  <CORSRule>
```

```
<ID>1234</ID>  
<AllowedOrigin>http://www.qq.com</AllowedOrigin>  
<AllowedMethod>PUT</AllowedMethod>  
<AllowedHeader>x-cos-meta-test</AllowedHeader>  
<ExposeHeader>x-cos-meta-test1</ExposeHeader>  
<MaxAgeSeconds>500</MaxAgeSeconds>  
</CORSRule>  
</CORSConfiguration>
```

Get Bucket Location

功能描述

Get Bucket Location接口获取Bucket所在地域信息，只有Bucket所有者有权限读取信息。

请求

请求语法

GET /?location HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
LocationConstraint	Bucket所在区域，枚举值：china-east, china-south, china-north, china-west, singapore	String

示例

请求

GET /?location HTTP/1.1

Host:arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484817522;32557713522&q-key-time=1484817522;32557713522&q-header-list=host&q-url-param-list=location&q-signature=ceb96fc929663dd4d2e6dc0aeb304cdde6761ed0

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 92

Connection: keep-alive

Date: Thu Jan 19 17:18:49 2017

Server: tencent-cos

x-cos-request-id: NTg4MDg0NzlfNDYyMDRIXzM0OWFfZjFk

<LocationConstraint>singapore</LocationConstraint>

Get Buket Lifecycle

功能描述

Get Bucket Lifecycle请求实现读取生命周期管理的配置。当配置不存在时，返回404 Not Found。

（目前只支持华南园区）

请求

请求语法

GET /?lifecycle HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
LifecycleConfiguration	描述此生命周期管理的所有配置信息，最大支持1000条规则。	Container
Rule	描述单条规则的具体信息，最大支持1000条规则 父节点：LifecycleConfiguration	Container
Prefix	前缀匹配策略，根目录则填空；多个Rule不能同时指定相同前缀路径；父目录指定某条Rule时，子目录不能指定相同类型的Rule 父节点：LifecycleConfiguration	String
Status	是否生效此条策略，枚举值：Enabled, Disabled 父节点：LifecycleConfiguration	String
ID	标示策略名称 父节点：LifecycleConfiguration	String
Transition	配置文件的生命周期管理配置 父节点：Rule	Container
Expiration	配置文件的定期删除规则，过期时间必须大于转换时间 父节点：Rule	Container
Days	在文件创建指定天数以后执行操作。Transition操作请用自然数，Expiration操作请用正整数，不同的Rule下可使用不同的时间格式，相同Rule下冲突 父节点：Transition，Expiration	Integer
Date	在指定时间点执行操作，时间格式为ISO 8601格式，须指定东八区触发当天午夜零点零分，可以指定历史时	Integer

名称	描述	类型
	间，例如2016-10-31T00:00:00+08:00；当前时间大于等于Date便执行操作，长期有效；不同的Rule下可使用不同的时间格式，相同Rule下冲突父节点：Transition，Expiration	
StorageClass	指定存储级别，枚举值：Standard，Standard_IA，Nearline 父节点：Transition	String
AbortIncompleteMultipartUpload	配置未完成分块上传的定期删除规则 父节点：Rule	Container
DaysAfterInitiation	在初始化分块上传之后指定天数以后Abort操作	正整数

配置文件的冷热沉降规则

<LifecycleConfiguration>

<Rule>

<ID> </ID>

<Prefix> </Prefix>

<Status> </Status>

<Transition>

<Date> </Date>

<StorageClass> </StorageClass>

</Transition>

</Rule>

<Rule>

<ID> </ID>

<Prefix> </Prefix>

<Status> </Status>

<Transition>

<Days> </Days>

<StorageClass> </StorageClass>

</Transition>

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

配置文件的定期删除规则

```
<LifecycleConfiguration>
  <Rule>
    <ID> </ID>
    <Prefix> </Prefix>
    <Status> </Status>
    <Expiration>
      <Date> </Date>
    </Expiration>
  </Rule>
  <Rule>
    <ID> </ID>
    <Prefix> </Prefix>
    <Status> </Status>
    <Expiration>
      <Days> </Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

配置未完成分块上传的定期删除规则

```
<LifecycleConfiguration>
  <Rule>
    <ID> </ID>
    <Prefix> </Prefix>
    <Status> </Status>
    <AbortIncompleteMultipartUpload>
```

```
<DaysAfterInititation> </DaysAfterInititation>  
</AbortIncompleteMultipartUpload>  
</Rule>  
</LifecycleConfiguration>
```

Get Bucket Tagging

功能描述

Get Bucket Tagging接口实现获取指定Bucket的标签。

请求

请求语法

GET /?tagging HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
Tagging	说明所有TagSet和Tag的信息	Container
TagSet	说明一系列的Tag信息 父节点：Tagging	Container
Tag	说明单个的Tag信息 父节点：TagSet	Container
Key	Tag的类别名称 父节点：Tag	String
Value	Tag的值 父节点：Tag	String

```
<Tagging>
  <TagSet>
    <Tag>
      <Key> </Key>
      <Value> </Value>
    </Tag>
    <Tag>
      ...
    </Tag>
  </TagSet>
</Tagging>
```

示例

请求

GET /?tagging HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484817283;32557713283&q-key-time=1484817283;32557713283&q-header-list=host&q-url-param-list=tagging&q-signature=b1da7bf83c43fd06fc4c5664ecb832b98966b193

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 96

Connection: keep-alive

Date: Thu Jan 19 17:14:51 2017

Server: tencent-cos

x-cos-request-id: NTg4MDgzOGJfOWExZjRlXzQ2YTBfZTY0

<Tagging>

<TagSet>

<Tag>

<Key>1</Key>

<Value>2</Value>

</Tag>

</TagSet>

</Tagging>

Put Bucket

功能描述

Put Bucket请求可以在指定账号下创建一个Bucket。

请求

请求语法

PUT / HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: auth

请求参数

无特殊请求参数

请求头部

推荐使用头部

名称	描述	类型	必选
x-cos-acl	允许用户自定义文件权限。 有效值：private，public-read 默认值：private	String	否
x-cos-grant-read	赋予被授权者读的权限 格式x-cos-grant-read: uin=" ",uin=" " 当需要给子账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAco	String	否

名称	描述	类型	必选
	untID"		
x-cos-grant-write	赋予被授权者写的权限 格式x-cos-grant-write: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-full-control	赋予被授权者读写权限 格式x-cos-grant-full-control: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

PUT / HTTP/1.1

Host: arlenhuangtestsgnoversion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484708728;32557604728&q-key-time=1484708728;32557604728&q-header-list=host&q-url-param-list=&q-signature=b394a86624cbcc705b11bc6fc505843c5e2dd9c9

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Wed Jan 18 11:05:42 2017

Server: tencent-cos

x-cos-request-id: NTg3ZWRiODJfOWIxZjRlXzZmNDBfMTUz

Put Bucket ACL

功能描述

使用API写入Bucket的ACL表，您可以通过Header："x-cos-acl","x-cos-grant-read","x-cos-grant-write","x-cos-grant-full-control"传入ACL信息，也可以通过body以XML格式传入ACL信息，但是只能选择

Header

和

Body

其中一种，否则返回冲突。

Put Bucket ACL是一个覆盖操作，传入新的ACL将覆盖原有ACL。只有所有者有权操作。

"x-cos-acl"：枚举值为public-read，private；public-read意味这个Bucket有公有读私有写的权限，private意味这个Bucket有私有读写的权限。

"x-cos-grant-read"：意味被赋予权限的用户拥有该Bucket的读权限

"x-cos-grant-write"：意味被赋予权限的用户拥有该Bucket的写权限

"x-cos-grant-full-control"：意味被赋予权限的用户拥有该Bucket的读写权限

请求

请求语法

PUT /?acl Http/1.1

Host:<BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Content-Type:application/xml

Content-MD5:MD5

x-cos-acl: [对应权限]

x-cos-grant-read: uin="",uin=""

x-cos-grant-write: uin="",uin=""

x-cos-grant-full-control: uin="",uin=""

Authorization: Auth String

请求参数

无特殊请求参数

请求头部

权限相关头部

参数名称	描述	类型	必选
x-cos-acl	定义Bucket的ACL属性，有效值：private, public-read，默认值：private	String	否
x-cos-grant-read	赋予被授权者读的权限，格式x-cos-grant-read: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-write	赋予被授权者写的权限，格式x-cos-grant-write: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否

参数名称	描述	类型	必选
	untID"		
x-cos-grant-full-control	赋予被授权者读写权限，格式x-cos-grant-full-control: uin=" ",uin=" " 当需要给予子账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否

请求内容

参数名称	描述	类型
AccessControlPolicy	一条独立的ACL记录	Container
Owner	标识资源的所有者	Container
uin	用户QQ号	String
Subaccount	子账户QQ账号	String
AccessControlList	被授权者信息与权限信息	Container
Grant	单条授权信息，每个AccessControlList可以拥有100条Grant	Container
Grantee	被授权者资源信息，type类型可以为RootAccount，SubAccount；当type类型为RootAccount时，可以在UIN中填写QQ，也可以填写anonymous（指代所有类型用户）。当type类型为RootAccount时，UIN代表根账户账号，SubAccount代表子账户账号	Container
Permission	权限信息，枚举值：READ，WRITE，FULL_CONTROL	String

<AccessControlPolicy>

<Owner>

<uin>ID</uin>

</Owner>

```
<AccessControlList>
<Grant>
<Grantee type="SubAccount">
<uin>ID</uin>
<Subaccount> SUBID </Subaccount>
</Grantee>
<Permission>Permission</Permission>
</Grant>
<Grant>
<Grantee type="RootAccount">
<uin>ID</uin>
</Grantee>
<Permission>Permission</Permission>
</Grant>
<Grant>
...
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

返回值

返回头部

无特殊返回头部，其他返回值请参见公共返回头部

返回内容

无返回内容

示例

请求

PUT /?acl HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484724784;32557620784&q-key-time=1484724784;32557620784&q-header-list=host&q-url-param-list=acl&q-signature=785d9075b8154119e6a075713c1b9e56ff0bddfc

Content-Length: 229

Content-Type: application/x-www-form-urlencoded

<AccessControlPolicy>

<Owner>

<uin>2779643970</uin>

</Owner>

<AccessControlList>

<Grant>

<Grantee type="RootAccount">

<uin>2779643970</uin>

</Grantee>

<Permission>FULL_CONTROL</Permission>

</Grant>

</AccessControlList>

</AccessControlPolicy>

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Wed Jan 18 15:41:31 2017

Server: tencent-cos

x-cos-request-id: NTg3ZjFjMmJfOWIxZjRlXzMNDhfMjIw

Put Bucket CORS

功能描述

Put Bucket CORS实现跨域访问设置，您可以通过传入XML格式的配置文件实现配置，文件大小限制为64 KB。

请求

请求语法

PUT /?cors HTTP 1.1

Host: <Bucketname>-<AppID>.<Region>.myqcloud.com

Date: date

Content-Length: length

Content-Type: application/xml

Content-MD5: MD5

Authorization: Auth

<XML 文件>

请求参数

无特殊请求参数

请求头部

必选头部

名称	描述	类型	必选
Content-MD5	RFC 1864 中定义的 128位 内容 MD5 算法校验值。	String	是

请求内容

名称	描述	类型	必选
CORSConfiguration	说明跨域配置的所有信息，最多可以包含100条CORSRule	Container	是
CORSRule	单条配置的信息 父节点：CORSRule	Container	是
ID	规则名称，可选填 父节点：CORSRule	String	否
AllowedMethod	允许的HTTP操作，枚举值：Get, Put, Head, Post, Delete 父节点：CORSRule	Enum	是
AllowedOrigin	允许的访问来源，支持『*』通配符，协议，端口和域名必须一致 父节点：CORSRule	String	是
AllowedHeader	在发送 OPTIONS 请求时告知服务端，接下来的请求可以使用哪些自定义的 HTTP 请求头部，支持『*』通配符 父节点：CORSRule	String	否
MaxAgeSeconds	设置 OPTIONS 请求得到结果的有效期 父节点：CORSRule	Integer	否
ExposeHeader	设置浏览器可以接收到的来自服务器端的自定义头部信息 父节点：CORSRule	String	否

<CORSConfiguration>

<CORSRule>

<ID> </ID>

<AllowedOrigin> </AllowedOrigin>

...

```
<AllowedMethod> </AllowedMethod>
...
<AllowedHeader> </AllowedHeader>
...
<MaxAgeSeconds> </MaxAgeSeconds>
<ExposeHeader> </ExposeHeader>
...
</CORSRule>
<CORSRule>
...
</CORSRule>
...
</CORSConfiguration>
```

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

PUT /?cors HTTP/1.1

Host:arlenhuangtestsgnoverversion-1251668577.sg.myqcloud.com

Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484814927;32557710927&q-key-time=1484814927;32557710927&q-header-list=host&q-url-param-list=cors&q-signature=8b9f05dabce2578f3a79d732386e7cbade9033e3

Content-Type:application/xml

Content-Length: 280

```
<CORSConfiguration>
  <CORSRule>
    <ID>1234</ID>
    <AllowedOrigin>http://www.qq.com</AllowedOrigin>
    <AllowedMethod>PUT</AllowedMethod>
    <AllowedHeader>x-cos-meta-test</AllowedHeader>
    <MaxAgeSeconds>500</MaxAgeSeconds>
    <ExposeHeader>x-cos-meta-test1</ExposeHeader>
  </CORSRule>
</CORSConfiguration>
```

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Thu Jan 19 16:42:21 2017

Server: tencent-cos

x-cos-request-id: NTg4MDdiZWRFOWExZjRlXzQ2OWVfZGY0

Put Bucket Lifecycle

功能描述

Put Bucket Lifecycle请求实现设置生命周期管理的功能。您可以通过该请求实现数据的生命周期管理配置和定期删除。此请求为覆盖操作，上传新的配置文件将覆盖之前的配置文件。生命周期管理对文件和文件夹同时生效。

（目前只支持华南园区）

请求

请求语法

PUT /?lifecycle HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Content-Type:application/xml

Content-MD5:MD5

Authorization: Auth

请求参数

无特殊请求参数

请求头部

必选头部

名称	描述	类型	必选
Content-MD5	RFC 1864 中定义的 128位内容 MD5 算法校验值。	String	是

请求内容

名称	描述	类型	必选
LifecycleConfiguration	描述此生命周期管理的所有配置信息，最大支持1000条规则。	Container	是
Rule	描述单条规则的具体信息，最大支持1000条规则 父节点：LifecycleConfiguration	Container	是
Prefix	前缀匹配策略，根目录则填空；多个Rule不能同时指定相同前缀路径；父目录指定某条Rule时，子目录不能指定相同类型的Rule 父节点：LifecycleConfiguration	String	是
Status	是否生效此条策略，枚举值：Enabled, Disabled 父节点：LifecycleConfiguration	String	是
ID	标示策略名称 父节点：LifecycleConfiguration	String	否
Transition	配置文件的生命周期管理配置 父节点：Rule	Container	是
Expiration	配置文件的定期删除规则，过期时间必须大于转换时间 父节点：Rule	Container	是
Days	在文件创建指定天数以后执行操作。Transition操作请用自然数，Expiration操作请用正整数，不同的Rule下可使用不同的时间格式，相同Rule下冲突	Intger	Days与Date二选一

名称	描述	类型	必选
	父节点：Transition，Expiration		
Date	在指定时间点执行操作，时间格式为ISO 8601格式，须指定东八区触发当天午夜零点零分，可以指定历史时间，例如2016-10-31T00:00:00+08:00；当前时间大于等于Date便执行操作，长期有效；不同的Rule下可使用不同的时间格式，相同Rule下冲突 父节点：Transition，Expiration	Integer	Days与Date二选一
StorageClass	指定存储级别，枚举值：Standard，Standard_IA，Nearline。 父节点：Transition	String	Transition中必须有
AbortIncompleteMultipartUpload	配置未完成分块上传的定期删除规则 父节点：Rule	Container	是
DaysAfterInitiation	在初始化分块上传之后指定天数以后Abort操作	正整数	是

配置文件的生命周期管理配置规则

<LifecycleConfiguration>

<Rule>

<ID> </ID>

<Prefix> </Prefix>

<Status> </Status>

<Transition>

<Date> </Date>

```
<StorageClass> </StorageClass>
</Transition>
</Rule>
<Rule>
<ID> </ID>
<Prefix> </Prefix>
<Status> </Status>
<Transition>
<Days> </Days>
<StorageClass> </StorageClass>
</Transition>
</Rule>
</LifecycleConfiguration>
```

配置文件的定期删除规则

```
<LifecycleConfiguration>
<Rule>
<ID> </ID>
<Prefix> </Prefix>
<Status> </Status>
<Expiration>
<Date> </Date>
</Expiration>
</Rule>
<Rule>
<ID> </ID>
<Prefix> </Prefix>
<Status> </Status>
<Expiration>
<Days> </Days>
</Expiration>
</Rule>
```

</LifecycleConfiguration>

配置未完成分块上传的定期删除规则

<LifecycleConfiguration>

<Rule>

<ID> </ID>

<Prefix> </Prefix>

<Status> </Status>

<AbortIncompleteMultipartUpload>

<DaysAfterInitiation> </DaysAfterInitiation>

</AbortIncompleteMultipartUpload>

</Rule>

</LifecycleConfiguration>

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

Put Bucket Tagging

功能描述

Put Bucket Tagging接口实现给用指定Bucket打标签。用来组织和管理相关Bucket。当该请求设置相同Key名称，不同Value时，会返回400。请求成功，则返回204。

请求

请求语法

PUT /?tagging HTTP 1.1

Host: <Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Content-Type:application/xml

Authorization: Auth

[XML]

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

名称	描述	类型	必选
Tagging	说明所有TagSet和Tag的信息	Container	是
TagSet	说明一系列的Tag信息	Container	是

名称	描述	类型	必选
	父节点：Tagging		
Tag	说明单个的Tag信息 父节点：TagSet	Container	是
Key	Tag的类别名称 父节点：Tag	String	是
Value	Tag的值 父节点：Tag	String	是

```

<Tagging>
  <TagSet>
    <Tag>
      <Key> </Key>
      <Value> </Value>
    </Tag>
    <Tag>
      ...
    </Tag>
  </TagSet>
</Tagging>

```

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

PUT /?tagging HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484817032;32557713032&q-key-time=1484817032;32557713032&q-header-list=host&q-url-param-list=tagging&q-signature=0a99e59b753c26b807e4b372560b0d026d26af26

Content-Type: application/xml

Content-Length: 75

<Tagging>

<TagSet>

<Tag>

<Key>1</Key>

<Value>2</Value>

</Tag>

</TagSet>

</Tagging>

返回

HTTP/1.1 204

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Thu Jan 19 17:12:29 2017

Server: tencent-cos

x-cos-request-id: NTg4MDgyZmRfOTkxZjRlXzEwNjRfZWl2

Delete Bucket

功能描述

Delete Bucket请求可以在指定账号下删除Bucket，删除之前要求Bucket为空。

请求

请求语法

DELETE / HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: authorization string

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

DELETE / HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-

ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj

fGFO&q-sign-time=1484708950;32557604950&q-key-time=1484708950;32557604950&q-header-list=host&q-url-param-list=&q-signature=2b27b72ad2540ff2dde341dc7579a66ee8cb2afc

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Wed Jan 18 11:09:21 2017

Server: tencent-cos

x-cos-request-id: NTg3ZWRjNjBfOTgxZjRlXzZhYjlfMTg0

Delete Bucket CORS

功能描述

Delete Bucket CORS实现跨域访问配置删除。

请求

请求语法

DELETE /?cors HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

DELETE /?cors HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-

ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj

fGFO&q-sign-time=1484816036;32557712036&q-key-time=1484816036;32557712036&q-header-list=host&q-url-param-list=cors&q-signature=e92eecbf0022fe7e5fd39b2c500b22da062be50a

返回

HTTP/1.1 204 No Content

Content-Type: application/xml

Content-Length: 405

Connection: keep-alive

Date: Thu Jan 19 16:54:04 2017

Server: tencent-cos

x-cos-request-id: NTg4MDdlYWVfOTgxZjRlXzZhYTlfZjAz

x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVvMDAxODczNTBmNjMwZmQ0MTZkMjg0NjlkNTYyNmY4ZTRkZTk0N2M2MTdkZGZlMGNhOWQyYjk3MWNmNWNkYzFhMjQzNzRiZTE1NjgzNzFhOGI5M2EwZDMYNGM4Y2ZmMzhiNTIIMjk=

Delete Bucket Lifecycle

功能描述

Delete Bucket Lifecycle请求实现删除生命周期管理。

(目前只支持华南园区)

请求

请求语法

DELETE /?lifecycle HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

Delete Bucket Tagging

功能描述

Delete Bucket Tagging接口实现删除指定Bucket的标签。

请求

请求语法

DELETE /?tagging HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Date:date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

DELETE /?tagging HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-

ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj

fGFO&q-sign-time=1484817338;32557713338&q-key-time=1484817338;32557713338&q-header-list=host&q-url-param-list=tagging&q-signature=fa13dedef474fe2034d2bb5b93b3afeffb225e5a

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Thu Jan 19 17:15:46 2017

Server: tencent-cos

x-cos-request-id: NTg4MDgzYzJfOWIxZjRlXzZmM2JfZWUw

Head Bucket

功能描述

Head

Bucket请求可以确认是否存在该Bucket，是否有权限访问，Head的权限与Read一致。当其存在时，返回 HTTP 状态码200；当无权限时，返回 HTTP 状态码403；当不存在时，返回 HTTP 状态码404。

请求

请求语法

HEAD / Http/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部参见公共返回头部

返回内容

无返回内容

示例

请求

HEAD / HTTP/1.1

Host: zuhaotestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484640517;32557536517&q-key-time=1484640517;32557536517&q-header-list=host&q-url-param-list=&q-signature=7bedc2f84a0a3d29df85fe727d0c1e388c410376

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Date: Tue Jan 17 16:09:39 2017

x-cos-request-id: NTg3ZGQxNDNfNDUyMDRIXzUyOWNfMjY5

List Multipart Uploads

功能描述

List Multipart Uploads用来查询正在进行中的分块上传。单次最多列出1000个正在进行中的分块上传。

请求

请求语法

GET /?uploads HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: auth

请求参数

名称	描述	类型	必选
delimiter	定界符为一个符号，如果有Prefix，则将Prefix到delimiter之间的相同路径归为一类，定义为Common Prefix，然后列出所有Common Prefix。如果没有Prefix，则从路径起点开始	String	否
encoding-type	规定返回值的编码方式	String	否
Prefix	前缀匹配，用来规定返回的文件前缀地址	String	否
max-uploads	单次返回最大的条目数量，默认1000	String	否
key-marker	与upload-id-marker一起使用 当upload-id-marker未被指定时，ObjectName字母	String	否

名称	描述	类型	必选
	顺序大于key-marker的条目将被列出 当upload-id-marker被指定时，ObjectName字母顺序大于key-marker的条目将被列出，ObjectName字母顺序等于key-marker同时UploadID大于upload-id-marker的条目将被列出。		
upload-id-marker	与key-marker一起使用 当key-marker未被指定时，upload-id-marker将被忽略 当key-marker被指定时，ObjectName字母顺序大于key-marker的条目将被列出，ObjectName字母顺序等于key-marker同时UploadID大于upload-id-marker的条目将被列出。	String	否

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
ListMultipartUploadsResult	用来表述所有分块上传的信息	Container
Bucket	分块上传的目标Bucket 父节点：ListMultipartUploadsResult	String
Encoding-type	规定返回值的编码方式 父节点：ListMultipartUploadsResult	String
KeyMarker	列出条目从该key值开始 父节点：ListMultipartUploadsResult	String
UploadIdMarker	列出条目从该UploadId值开始 父节点：ListMultipartUploadsResult	String
NextKeyMarker	假如返回条目被截断，则返回NextKeyMarker就是下一个条目的起点 父节点：ListMultipartUploadsResult	String
NextUploadIdMarker	假如返回条目被截断，则返回UploadId就是下一个条目的起点 父节点：ListMultipartUploadsResult	String
MaxUploads	单次返回最大的条目数量 父节点：ListMultipartUploadsResult	String
IsTruncated	返回条目是否被截断，布尔值：True, False 父节点：ListMultipartUploadsResult	Boolean
Upload	每个Upload的信息	Container

名称	描述	类型
	父节点：ListMultipartUploadsResult	
Key	Object的名称 父节点：Upload	Integer
UploadID	标示本次分块上传的ID 父节点：Upload	Integer
StorageClass	用来表示分块的存储级别，枚举值：Standard, Standard_IA, Nearline 父节点：Upload	String
Initiator	用来表示本次上传发起者的信息，子节点包括UID 父节点：Upload	Container
UID	开发商APPID 父节点：Initiator, Owner	String
Owner	用来表示这些分块所有者的信息，子节点包括UID 父节点：Upload	Container
Initiated	分块上传的起始时间 父节点：Upload	Date
ListMultipartUploadsResult.Prefix	前缀匹配，用来规定返回的文件前缀地址 父节点：ListMultipartUploadsResult	String
delimiter	定界符为一个符号，如果有Prefix，则将Prefix到delimiter之间的相同路径归为一类，定义为Common Prefix，然后列出所有Common Prefix。如果没有Prefix，则从路径起点开始 父节点：ListMultipartUploadsResult	String
CommonPrefixes	将Prefix到delimiter之间的相同路径归为一类，定义为Common Prefix	Container

名称	描述	类型
	父节点：ListMultipartUploadsResult	
CommonPrefixes.Prefix	显示具体的CommonPrefixes 父节点：CommonPrefixes	String

```

<ListMultipartUploadsResult>
  <Bucket> </Bucket>
  <Encoding-type> </Encoding-type>
  <KeyMarker> </KeyMarker>
  <UploadIdMarker> </UploadIdMarker>
  <NextKeyMarker> </NextKeyMarker>
  <NextUploadIdMarker> </NextUploadIdMarker>
  <MaxUploads> </MaxUploads>
  <IsTruncated> </IsTruncated>
  <Prefix> </Prefix>
  <delimiter> </delimiter>
  <Upload>
    <Key> </Key>
    <UploadID> </UploadID>
    <StorageClass> </StorageClass>
    <Initiator>
      <UID> </UID>
    </Initiator>
    <Owner>
      <UID> </UID>
    </Owner>
    <Initiated> </Initiated>
  </Upload>
  <CommonPrefixes>
    <Prefix> </Prefix>
  </CommonPrefixes>
</ListMultipartUploadsResult>

```

示例

请求

GET /?uploads HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484727508;32557623508&q-key-time=1484727508;32557623508&q-header-list=host&q-url-param-list=uploads&q-signature=5bd4759a7309f7da9a0550c224d8c61589c9dbbf

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 1203

Date: Wed Jan 18 16:18:37 2017

Server: tencent-cos

x-cos-request-id: NTg3ZjI0ZGRfNDQyMDRIXzNhZmRfMjRI

<ListMultipartUploadsResult>

<Bucket>arlenhuangtestsgnoverion</Bucket>

<Encoding-Type/>

<KeyMarker/>

<UploadIdMarker/>

<MaxUploads>1000</MaxUploads>

<Prefix/>

<Delimiter>/</Delimiter>

<IsTruncated>>false</IsTruncated>

<Upload>

<Key>Object</Key>

<UploadID>

1484726657932bcb5b17f7a98a8cad9fc36a340ff204c79bd2f51e7dddf0b6d1da6220520c</UploadID>

<Initiator>

```
<UID/>
</Initiator>
<Owner>
<UID>1251668577</UID>
</Owner>
<StorageClass>Standard</StorageClass>
<Initiated>Wed Jan 18 16:04:17 2017</Initiated>
</Upload>
<Upload>
<Key>Object</Key>
<UploadID>
1484727158f2b8034e5407d18cbf28e84f754b791ecab607d25a2e52de9fee641e5f60707c</UploadID>
<Initiator>
<UID/>
</Initiator>
<Owner>
<UID>1251668577</UID>
</Owner>
<StorageClass>Standard</StorageClass>
<Initiated>Wed Jan 18 16:12:38 2017</Initiated>
</Upload>
<Upload>
<Key>ObjectName</Key>
<UploadID>
148472727
0323ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e</UploadID>
<Initiator>
<UID/>
</Initiator>
<Owner>
<UID>1251668577</UID>
</Owner>
<StorageClass>Standard</StorageClass>
<Initiated>Wed Jan 18 16:14:30 2017</Initiated>
```

</Upload>

</ListMultipartUploadsResult>

Object接口

Append Object

功能描述

Append请求可以将一个文件（Object）以分块追加的方式上传至 Bucket 中。使用Append Upload的文件必须事前被设定为Appendable。当Appendable的文件被执行Put Object的操作以后，文件被覆盖，属性改变为Normal。

文件属性可以在Head Object操作中被查询到，当您发起Head Object请求时，会返回自定义Header『x-cos-object-type』，该Header只有两个枚举值：Normal或者Appendable。

追加上传建议文件大小1M - 5G。如果position的值和当前Object的长度不致，COS会返回409错误。如果Append一个Normal的Object，COS会返回409 ObjectNotAppendable。

Appendable的文件不可以被复制，不参与版本管理，不参与生命周期管理，不可跨区域复制。

请求

请求语法

POST /ObjectName?append&position=*position* HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Content-Length: size

Content-Type: ContentType

Date: date

Authorization: auth

请求参数

名称	描述	必选
Position	追加操作的起始点，单位：字节。首次追加Position=0，后续追加，Posi	

名称	描述	必选
	tion=当前文件content-length	

请求头部

必选头部

名称	描述	类型	必选
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）。	String	是

推荐使用头部

名称	描述	类型	必选
Cache-Control	RFC 2616 中定义的缓存策略，将作为 Object 元数据返回。	String	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为 Object 元数据返回。	String	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为 Object 元数据返回。	String	否
Content-MD5	RFC 1864 中定义的 128-bit 内容 MD5 算法校验值。	String	否
Content-Type	RFC 2616 中定义的内容类型（ MIME ），将作为 Object 元数据返回。	String	否
Expect	当使用 Expect: 100-continue 时，在收到服务端确认后，才会发送请求内容。	String	否
Expires	RFC 2616 中定义的过期时间，将作为 Object 元数据返回。	String	否

名称	描述	类型	必选
x-cos-content-sha1	RFC 3174 中定义的 160-bit 内容 SHA-1 算法校验值。	String	否
x-cos-meta-*	允许用户自定义的头部信息，将作为 Object 元数据返回。大小限制2K。	String	否

权限相关头部

名称	描述	类型	必选
x-cos-acl	允许用户自定义文件权限。 有效值：private，public-read，public-read-write	String	否
x-cos-grant-read	赋予被授权者读的权限 格式x-cos-grant-read: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-write	赋予被授权者写的权限 格式x-cos-grant-write: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-full-control	赋予被授权者读写权限 格式x-cos-grant-full-control: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"	String	否

名称	描述	类型	必选
	countID"，当需要给根账户授权时，uin="RootAccountID"		

请求内容

追加的文件内容

返回值

返回头部

名称	描述	类型
x-cos-next-append-position	下一次追加操作的起始点，单位：字节	String
x-cos-content-sha1	分段的校验值	String
ETag	文件的唯一标识	String

返回内容

无返回内容

示例

请求

POST /coss3/app?append&position=0 HTTP/1.1

Host: zuhaotestnorth-1251668577.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3MqQCn&q-sign-time=1484208848;32557104848&q-key-time=1484208848;32557104848&q-header-list=host&q-url-param-list=append;position&q-signature=855fe6b833fadf20570f7f650e2120e17ce8a2fe

Content-Length: 4096

[Object]

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Thu Jan 12 16:14:24 2017

ETag: 1ce5b469b7d6600ecc2fd112e570917b

Server: tencent-cos

x-cos-content-sha1: 1ceaf73df40e531df3bfb26b4fb7cd95fb7bff1d

x-cos-next-append-position: 4096

x-cos-request-id: NTg3NzNhZGZfMmM4OGY3X2I2Zl8xMTBm

Get Object

功能描述

Get Object 请求可以将一个文件（Object）下载至本地。该操作需要对目标 Object 具有读权限或目标 Object 对所有人都开放了读权限（公有读）。

请求

请求语法

GET /ObjectName Http/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: authorization string(公有读无需此头部，若携带也无效)

Range: bytes=byte_range

请求参数

名称	描述	类型	必选
response-content-type	设置返回头部中的 Content-Type 参数。	String	否
response-content-language	设置返回头部中的 Content-Language 参数。	String	否
response-expires	设置返回头部中的 Content-Expires 参数。	String	否
response-cache-control	设置返回头部中的 Cache-Control 参数。	String	否
response-content-disposition	设置返回头部中的 Content-Disposition 参数。	String	否
response-content-encoding	设置返回头部中的 Content-Encoding 参数。	String	否

请求头部

推荐使用头部

名称	描述	类型	必选
Range	RFC 2616 中定义的指定文件下载范围，以字节（bytes）为单位。	String	否
If-Modified-Since	如果文件修改时间晚于指定时间，才返回文件内容。否则返回 304 (not modified)。	String	否

请求内容

无请求内容

返回值

返回头部

名称	描述	类型
x-cos-meta-*	用户自定义的元数据	String
x-cos-object-type	用来表示object是否可以被追加上传，枚举值：normal或者appendable	string
x-cos-storage-class	Object的存储级别，枚举值：Standard, Standard_IA, Nearline	String

返回内容

文件内容

示例

请求

GET /123 HTTP/1.1

Host: zuhaotestnorth-1251668577.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484212200;32557108200&q-key-time=1484212200;32557108200&q-header-list=host&q-url-param-list=&q-signature=11522aa3346819b7e5e841507d5b7f156f34e639

返回

HTTP/1.1 200 OK

Date: Thu, 12 Jan 2017 09:10:22 GMT

Content-Type: application/octet-stream

Content-Length: 16087

Connection: keep-alive

Accept-Ranges: bytes

Content-Disposition: attachment; filename*="UTF-8''123"

Content-Range: bytes 0-16086/16087

ETag: "9a4802d5c99dafa1c04da0a8e7e166bf"

Last-Modified: Wed, 11 Jan 2017 15:30:07 GMT

x-cos-object-type: normal

x-cos-request-id: NTg3NzQ3ZmVfYmRjMzVfMzE5N182NzczMQ==

[Object]

Get Object ACL

功能描述

Get Object ACL接口实现使用API读取Object的ACL表，只有所有者有权操作。

请求

请求语法

GET /ObjectName?acl Http/1.1

Host:<BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

必选头部

参数名称	描述	类型	必选
Authorization	签名串	String	是

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

参数名称	描述	类型
AccessControlPolicy	一条独立的ACL记录	Container
Owner	标识资源的所有者	Container
uin	用户QQ号	String
Subaccount	子账户QQ账号	String
AccessControlList	被授权者信息与权限信息	Container
Grant	单条授权信息，一个AccessControlList钟可以拥有100条Grant	Container
Grantee	被授权者资源信息，type类型可以为RootAccount，SubAccount；当type类型为RootAccount时，可以在UIN中填写QQ，也可以填写anonymous（指代所有类型用户）。当type类型为RootAccount时，UIN代表根账户账号，SubAccount代表子账户账号	Container
Permission	权限信息，枚举值：READ，WRITE，FULL_CONTROL	String

```

<AccessControlPolicy>
  <Owner>
    <uin>ID</uin>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee type="SubAccount">
        <uin>ID</uin>
        <Subaccount> SUBID </Subaccount>
      </Grantee>
      <Permission>Permission</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>

```

```
<Grant>
<Grantee type="RootAccount">
<uin>ID</uin>
</Grantee>
<Permission>Permission</Permission>
</Grant>
<Grant>
...
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

示例

请求

```
GET /ObjectName?acl HTTP/1.1
Host:zuhaotestsgnoverion-1251668577.sg.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484641748;32557537748&q-key-time=1484641748;32557537748&q-header-list=host&q-url-param-list=acl&q-signature=79a3cb15cd1a2a2b6d567d78c7e8f9cd895bde21
```

返回

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 266
Date: Tue Jan 17 16:30:03 2017
x-cos-request-id: NTg3ZGQ2MGJfNDQyMDRIXzE3YTdfMjk3
```

```
<AccessControlPolicy>
  <Owner>
```



```
<uin>2779643970</uin>
</Owner>
<AccessControlList>
  <Grant>
    <Grantee type="RootAccount">
      <uin>2779643970</uin>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>
```

Put Object

功能描述

Put Object请求可以将一个文件（Object）上传至指定Bucket。

请求

请求语法

PUT /ObjectName HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: authorization string

请求参数

无特殊请求参数

请求头部

必选头部

名称	描述	类型	必选
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）。	String	是

推荐使用头部

名称	描述	类型	必选
Cache-Control	RFC 2616 中定义的缓存策略，将作为 Object 元数据保存。	String	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为	String	否

名称	描述	类型	必选
	Object 元数据保存。		
Content-Encoding	RFC 2616 中定义的编码格式，将作为 Object 元数据保存。	String	否
Content-Type	RFC 2616 中定义的内容类型（ MIME ），将作为 Object 元数据保存。	String	否
Expect	当使用 Expect: 100-continue 时，在收到服务端确认后，才会发送请求内容。	String	否
Expires	RFC 2616 中定义的过期时间，将作为 Object 元数据保存。	String	否
x-cos-content-sha1	RFC 3174 中定义的 160-bit 内容 SHA-1 算法校验值。	String	否
x-cos-meta-*	允许用户自定义的头部信息，将作为 Object 元数据返回。大小限制2K。	String	否
x-cos-storage-class	设置Object的存储级别，枚举值：Standard，Standard_IA，Nearline，默认值：Standard（目前只支持华南园区）	String	否

权限相关头部

名称	描述	类型	必选
x-cos-acl	允许用户自定义文件权限。有效值：private，public-read	String	否
x-cos-grant-read	赋予被授权者读的权限格式x-cos-grant-read:	String	否

名称	描述	类型	必选
	uin=" ",uin=" " 当需要给子账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"		
x-cos-grant-write	赋予被授权者写的权限 格式x-cos-grant-write: uin=" ",uin=" " 当需要给子账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-full-control	赋予被授权者读写权限 格式x-cos-grant-full-control: uin=" ",uin=" " 当需要给子账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否

请求内容

文件内容

返回值

返回头部

名称	描述	类型
ETag	返回文件的 MD5	String

名称	描述	类型
	算法校验值。ETag 的值可以用于检查 Object 的内容是否发生变化。	

返回内容

无返回内容

示例

请求

PUT /ObjectName HTTP/1.1

Host: zuhaotestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484639384;32557535384&q-key-time=1484639384;32557535384&q-header-list=host&q-url-param-list=&q-signature=5c07b7c67d56497d9aacb1adc19963135b7d00dc

Content-Length: 64

[Object]

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Date: Tue Jan 17 15:50:02 2017

Etag: 020df6d63448ae38a1de7924a68ba1e2

x-cos-request-id: NTg3ZGNjYTlfNDUyMDRIXzUyOTlfMjRj

Put Object ACL

功能描述

使用API写入Object的ACL表，您可以通过Header："x-cos-acl", "x-cos-grant-read", "x-cos-grant-write", "x-cos-grant-full-control"传入ACL信息，也可以通过body以XML格式传入ACL信息，但是只能选择

Header

和

Body

其中一种，否则，返回冲突。

Put Object ACL是一个覆盖操作，传入新的ACL将覆盖原有ACL。只有所有者有权操作。

"x-cos-acl"：枚举值为public-read，private；public-read意味这个Object有公有读私有写的权限，private意味这个Object有私有读写的权限。

"x-cos-grant-read"：意味被赋予权限的用户拥有该Object的读权限

"x-cos-grant-write"：意味被赋予权限的用户拥有该Object的写权限

"x-cos-grant-full-control"：意味被赋予权限的用户拥有该Object的读写权限

请求

请求语法

PUT /ObjectName?acl Http/1.1

Host:<BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Content-Type:application/xml

Content-MD5:MD5

x-cos-acl: [对应权限]

x-cos-grant-read: uin="",uin=""

x-cos-grant-write: uin="",uin=""

x-cos-grant-full-control: uin="",uin=""

Authorization: Auth

请求参数

无特殊请求参数

请求头部

权限相关头部

参数名称	描述	类型	必选
x-cos-acl	定义Object的ACL属性，有效值：private, public-read，默认值：private	String	否
x-cos-grant-read	赋予被授权者读的权限，格式x-cos-grant-read: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-write	赋予被授权者写的权限，格式x-cos-grant-write: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否

参数名称	描述	类型	必选
	untID"		
x-cos-grant-full-control	赋予被授权者读写权限，格式x-cos-grant-full-control: uin=" ",uin=" " 当需要给予子账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否

请求内容

参数名称	描述	类型
AccessControlPolicy	一条独立的ACL记录	Container
Owner	标识资源的所有者	Container
uin	用户QQ号	String
Subaccount	子账户QQ账号	String
AccessControlList	被授权者信息与权限信息	Container
Grant	单条授权信息，一个AccessControlList钟可以拥有100条Grant	Container
Grantee	被授权者资源信息，type类型可以为RootAccount，SubAccount；当type类型为RootAccount时，可以在UIN中填写QQ，也可以填写anonymous（指代所有类型用户）。当type类型为RootAccount时，UIN代表根账户账号，SubAccount代表子账户账号	Container
Permission	权限信息，枚举值：READ，WRITE，FULL_CONTROL	String

<AccessControlPolicy>

<Owner>

<uin>ID</uin>

</Owner>


```
<AccessControlList>
<Grant>
<Grantee type="SubAccount">
<uin>ID</uin>
<Subaccount> SUBID </Subaccount>
</Grantee>
<Permission>Permission</Permission>
</Grant>
<Grant>
<Grantee type="RootAccount">
<uin>ID</uin>
</Grantee>
<Permission>Permission</Permission>
</Grant>
<Grant>
...
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

PUT /ObjectName?acl HTTP/1.1

Host: arlenhuangtestsgnoverversion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484726016;32557622016&q-key-time=1484726016;32557622016&q-header-list=host&q-url-param-list=acl&q-signature=e8b788abcd242ac20b6412205460f1de9afe0f7a

Content-Length: 229

Content-Type: application/x-www-form-urlencoded

<AccessControlPolicy>

<Owner>

<uin>2779643970</uin>

</Owner>

<AccessControlList>

<Grant>

<Grantee type="RootAccount">

<uin>2779643970</uin>

</Grantee>

<Permission>FULL_CONTROL</Permission>

</Grant>

</AccessControlList>

</AccessControlPolicy>

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Wed Jan 18 15:54:43 2017

Server: tencent-cos

x-cos-request-id: NTg3ZjFmNDNfOWIxZjRlXzMzMzlfMjEz

Delete Object

功能描述

Delete Object请求可以将一个文件（Object）删除。

请求

请求语法

DELETE /ObjectName HTTP/1.1

Host:<BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Content-Length:length

Authorization: auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

DELETE /123 HTTP/1.1

Host: zuhaotestnorth-1251668577.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-

ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj

fGFO&q-sign-time=1484213409;32557109409&q-key-time=1484213409;32557109409&q-header-list=host&q-url-param-list=&q-signature=1c24fe260ffe79b8603f932c4e916a6cbb0af44a

返回

HTTP/1.1 204 No Content

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Thu Jan 12 17:30:12 2017

Server: tencent-cos

x-cos-request-id: NTg3NzRjYTRfYmRjMzVfMzFhOF82MmM3Yg==

Delete Multiple Object

功能描述

Delete Multiple Object请求实现批量删除文件，最大支持单次删除1000个文件。对于返回结果，COS提供Verbose和Quiet两种结果模式。Verbose模式将返回每个Object的删除结果；Quiet模式只返回报错的Object信息。

此请求必须携带x-cos-sha1用来校验Body的完整性。

请求

请求语法

POST /?delete HTTP/1.1

Host: <Bucketname>-<AppID>.<Region>.myqcloud.com

Date: date

Content-Length:length

Content-Type:application/xml

Content-MD5:MD5

Authorization: authorization string

<Delete>

<Quiet> </Quiet>

<Object>

<Key> </Key>

</Object>

<Object>

<Key> </Key>

</Object>

...

</Delete>

请求参数

无特殊请求参数

请求头部

必选头部

名称	描述	类型	必须
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）。	String	是
Content-MD5	RFC 1864 中定义的 128位 内容 MD5 算法校验值。	String	是

请求内容

名称	描述	类型	必须
Delete	说明本次删除的返回结果方式和目标Object	Container	是
Quiet	布尔值，这个值决定了是否启动Quiet模式，True启动Quiet模式，False启动Verbose模式，默认False 父节点：Delete	Boolean	否
Object	说明每个将要删除的目标文件信息	Container	是
Key	目标文件名	String	是

<Delete>

<Quiet> </Quiet>

<Object>

<Key> </Key>

</Object>

<Object>

<Key> </Key>

</Object>

...

</Delete>

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

名称	描述	类型
DeleteResult	说明本次删除的返回结果	Container
Deleted	说明本次删除的成功Object信息 父节点：DeleteResult	Container
Key	Object的名称 父节点：Deleted，Error	String
Error	说明本次删除的失败Object信息 父节点：DeleteResult	Container
Code	删除失败的错误码	String
Message	删除错误信息	String

<DeleteResult>

<Deleted>

<Key> </Key>

</Deleted>

<Error>

<Key> </Key>

<Code> </Code>

<Message> </Message>

</Error>

</DeleteResult>

示例

请求

POST /coss3/?delete HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Content-MD5: 35385efb5ba5134bffb192bfa17c3d5e

Authorization: q-sign-algorithm=sha1&q-

ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3M

qQCn&q-sign-time=1487065662;32466649662&q-key-time=1487065662;32559961662&q-header-list=host&q-url-param-list=delete&q-signature=286ef48c81f1652c37c635f0fb7db7a2150aa5ba

Content-Length: 75

Content-Type: application/x-www-form-urlencoded

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 79

Connection: keep-alive

Date: Tue Feb 14 17:49:12 2017

Server: tencent-cos

x-cos-request-id: NThhMmQyOTdfMmM4OGY3XzZjZGFfY2Mx

<DeleteResult>

<Deleted>

<Key>ObjectName</Key>

</Deleted>

</DeleteResult>

Head Object

功能描述

Head Object请求可以取回对应Object的元数据，Head的权限与Get的权限一致

请求

请求语法

HEAD /ObjectName HTTP/1.1

Host:<BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: auth

请求参数

无特殊请求参数

请求头部

推荐使用头部

名称	描述	类型	必选
If-Modified-Since	当Object在指定时间后被修改，则返回对应Object元信息，否则返回304	String	否

请求内容

无请求内容

返回值

返回头部

名称	描述	类型
x-cos-meta-*	用户自定义的元数据	String
x-cos-object-type	用来表示object是否可以被追加上传 ，枚举值：normal或者appendable	string
x-cos-storage-class	Object的存储级别，枚举值：Stand ard，Standard_IA，Nearline	String

返回内容

无返回内容

示例

请求

HEAD /123 HTTP/1.1

Host: zuhaotestnorth-1251668577.cn-north.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484213210;32557109210&q-key-time=1484213210;32557109210&q-header-list=host&q-url-param-list=&q-signature=ac61b8eb61964e7e6b935e89de163a479a25c210

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 16087

Connection: keep-alive

Date: Thu Jan 12 17:26:53 2017

ETag: "9a4802d5c99dafe1c04da0a8e7e166bf"

Last-Modified: Wed, 11 Jan 2017 07:30:07 GMT

Server: tencent-cos

x-cos-object-type: normal

x-cos-request-id: NTg3NzRiZGRfYmRjMzVfM2Y2OF81N2YzNA==

Options Object

功能描述

Options Object请求实现跨域访问的预请求。即发出一个 OPTIONS 请求给服务器以确认是否可以进行跨域操作。

当CORS配置不存在时，请求返回403 Forbidden。

请求

请求语法

OPTIONS /ObjectName HTTP 1.1

Host:<Bucketname>-<AppID>.<Region>.myqcloud.com

Origin:Origin

Access-Control-Request-Method:HTTPMethod

Access-Control-Request-Headers:RequestHeader

请求参数

无特殊请求参数

请求头部

必选头部

名称	描述	类型	必选
Origin	模拟跨域访问的请求来源域名	String	是
Access-Control-Request-Method	模拟跨域访问的请求HTTP方法	String	是

推荐使用头部

名称	描述	类型	必选
Access-Control-Request-Headers	模拟跨域访问的请求头部	String	否

请求内容

无请求内容

返回值

返回头部

名称	描述	类型
Access-Control-Allow-Origin	模拟跨域访问的请求来源域名，当来源不允许的时候，此Header不返回。	String
Access-Control-Allow-Methods	模拟跨域访问的请求HTTP方法，当请求方法不允许的时候，此Header不返回。	String
Access-Control-Allow-Headers	模拟跨域访问的请求头部，当模拟任何请求头部不允许的时候，此Header不返回该请求头部。	String
Access-Control-Expose-Headers	跨域支持返回头部，用逗号区分	String
Access-Control-Max-Age	设置 OPTIONS 请求得到结果的有效期	String

返回内容

无返回内容

示例

请求

OPTIONS /coss3/ObjectName HTTP/1.1

Host:arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Origin:http://www.qq.com

Access-Control-Request-Method:PUT

Authorization:q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3MqQCn&q-sign-time=1487070734;32466654734&q-key-time=1487070734;32559966734&q-header-list=host&q-url-param-list=&q-signature=2ac3ada19910f44836ae0df72a0ec1003f34324b

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Access-Control-Allow-Headers: x-cos-meta-test

Access-Control-Allow-Methods: PUT

Access-Control-Allow-Origin: http://www.qq.com

Access-Control-Expose-Headers: x-cos-meta-test1

Access-Control-Max-Age: 500

Date: Tue Feb 14 19:13:51 2017

Server: tencent-cos

x-cos-request-id: NThhMmU2NmZfMmM4OGY3XzZjZGFfMTkzNw==

Initiate Multipart Upload

功能描述

Initiate Multipart Upload请求实现初始化分片上传，成功执行此请求以后会返回Upload ID用于后续的Upload Part请求。

请求

请求语法

POST /Object?uploads HTTP 1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: Auth

请求参数

无特殊请求参数

请求头部

推荐使用头部

名称	描述	类型	必选
Cache-Control	RFC 2616 中定义的缓存策略，将作为Object 元数据保存。	String	否
Content-Disposition	RFC 2616 中定义的文件名称，将作为Object 元数据保存。	String	否
Content-Encoding	RFC 2616 中定义的编码格式，将作为Object 元数据保存。	String	否

名称	描述	类型	必选
Content-Type	RFC 2616 中定义的内容类型（ MIME ），将作为 Object 元数据保存。	String	否
Expires	RFC 2616 中定义的过期时间，将作为 Object 元数据保存。	String	否
x-cos-meta-*	允许用户自定义的头部信息，将作为 Object 元数据返回。大小限制2K。	String	否
x-cos-storage-class	设置Object的存储级别，枚举值：Standard，Standard_IA，Nearline，默认值：Standard（目前只支持华南园区）	String	否

权限相关头部

名称	描述	类型	必选
x-cos-acl	允许用户自定义文件权限。有效值：private，public-read	String	否
x-cos-grant-read	赋予被授权者读的权限 格式x-cos-grant-read: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"，当需要给根账户授权时，uin="RootAccountID"	String	否
x-cos-grant-write	赋予被授权者写的权限 格式x-cos-grant-write: uin=" ",uin=" " 当需要给予账户授权时，uin="RootAccountID/SubAccountID"	String	否

名称	描述	类型	必选
	countID" , 当需要给根账户授权时 , uin="RootAccountID"		
x-cos-grant-full-control	赋予被授权者读写权限 格式x-cos-grant-full-control: uin=" ",uin=" " 当需要给子账户授权时 , uin="RootAccountID/SubAccountID" , 当需要给根账户授权时 , uin="RootAccountID"	String	否

请求内容

无请求内容

返回值

返回头部

无特殊返回头部

返回内容

名称	描述	类型
InitiateMultipartUploadResult	说明所有返回信息	Container
Bucket	分片上传的目标Bucket	String
Key	Object的名称	String
UploadId	在后续上传中使用的ID	String

<InitiateMultipartUploadResult>

<Bucket> </Bucket>

<Key> </Key>

<UploadId> </UploadId>

</InitiateMultipartUploadResult>

示例

请求

POST /ObjectName?uploads HTTP/1.1

Host: arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484727259;32557623259&q-key-time=1484727259;32557623259&q-header-list=host&q-url-param-list=uploads&q-signature=b5f46c47379aeae74be7578380b193c01b28045

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 230

Connection: keep-alive

Date: Wed Jan 18 16:14:30 2017

Server: tencent-cos

x-cos-request-id: NTg3ZjZlZTZfOWIxZjRlXzZmMzhfMWRj

<InitiateMultipartUploadResult>

<Bucket> arlenhuangtestsgnoverion </Bucket>

<Key> ObjectName </Key>

<UploadId>

148472727

0323ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e </UploadId>

</InitiateMultipartUploadResult>

Upload Part

功能描述

Upload Part请求实现在初始化以后的分块上传，支持的块的数量为1到10000，块的大小为1 MB 到5 GB。在每次请求Upload Part时候，需要携带partNumber和uploadID，partNumber为块的编号，支持乱序上传。

当传入uploadID和partNumber都相同的时候，后传入的块将覆盖之前传入的块。当uploadID不存在时会返回404错误，NoSuchUpload。

请求

请求语法

PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP 1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Content-Length: Size

Authorization: Auth

请求参数

无特殊请求参数

请求头部

必选头部

名称	描述	类型	必选
Content-Length	RFC 2616 中定义的 HTTP 请求内容长度（字节）。	String	是

推荐使用头部

名称	描述	类型	必选
Expect	当使用 Expect: 100-continue 时，在收到服务端确认后，才会发送请求内容。	String	否
x-cos-content-sha1	RFC 3174 中定义的 160-bit 内容 SHA-1 算法校验值。	String	否

请求内容

文件内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

PUT

/ObjectName?partNumber=1&uploadId=1484727270323

ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e HTTP/1.1

Host:arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484727403;32557623403&q-key-time=1484727403;32557623403&q-header-list=host&q-url-param-list=partNumber;uploadId&q-signature=bfc54518ca8fc31b3ea287f1ed2a0dd8c8e88a1d

Content-Length: 10485760

[Object]

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Wed Jan 18 16:17:03 2017

Etag: "e1e5b4965bc7d30880ed6d226f78a5390f1c09fc"

Server: tencent-cos

x-cos-request-id: NTg3ZjI0NzlfOWIxZjRlXzZmNGJfMWYy

List Parts

功能描述

List Parts用来查询特定分块上传中的已上传的块。

请求

请求语法

GET /ObjectName?uploadId=UploadId HTTP/1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Authorization: auth

请求参数

名称	描述	类型	必选
UploadID	标示本次分块上传的ID	String	是
Encoding-type	规定返回值的编码方式	String	否
max-parts	单次返回最大的条目数量，默认1000	String	否
part-number-marker	默认以UTF-8二进制顺序列出条目，所有列出条目从marker开始	String	否

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无返回头部

返回内容

名称	描述	类型
ListPartsResult	用来表述本次分块上传的所有信息，子节点包括：Bucket，Encoding-type，Key，UploadID，Initiator，Owner，PartNumberMarker，NextPartNumberMarker，MaxParts，IsTruncated，Part	Container
Bucket	分块上传的目标Bucket 父节点：ListPartsResult	String
Encoding-type	规定返回值的编码方式 父节点：ListPartsResult	String
Key	Object的名称 父节点：ListPartsResult	String
UploadID	标示本次分块上传的ID 父节点：ListPartsResult	String
Initiator	用来表示本次上传发起者的信息，子节点包括UID 父节点：ListPartsResult	Container
UID	开发商APPID	String
Owner	用来表示这些分块所有者的信息，子节点包括UID 父节点：ListPartsResult	Container
StorageClass	用来表示这些分块的存储级别，枚举值：Standard，Standard_IA，Nearline 父节点：ListPartsResult	String

名称	描述	类型
PartNumberMarker	默认以UTF-8二进制顺序列出条目， 所有列出条目从marker开始 父节点：ListPartsResult	String
NextPartNumberMarker	假如返回条目被截断，则返回Next Marker就是下一个条目的起点 父节点：ListPartsResult	String
MaxParts	单次返回最大的条目数量 父节点：ListPartsResult	String
IsTruncated	返回条目是否被截断，布尔值：True , False 父节点：ListPartsResult	Boolean
Part	用来表示每一个块的信息 父节点：ListPartsResult	Container
PartNumber	块的编号 父节点：Part	String
LastModified	块最后修改时间 父节点：Part	Date
Etag	块的 SHA-1 算法校验值 父节点：Part	String
Size	块大小，单位Byte 父节点：Part	String

<ListPartsResult>

<Bucket> </Bucket>

<Encoding-type> </Encoding-type>

<Key> </Key>

<UploadID> </UploadID>

<Initiator>

<UID> </UID>

</Initiator>

<Owner>

<UID> </UID>

</Owner>

<StorageClass> </StorageClass>


```
<PartNumberMarker> </PartNumberMarker>
<NextPartNumberMarker> </NextPartNumberMarker>
<MaxParts> </MaxParts>
<IsTruncated> </IsTruncated>
<Part>
  <PartNumber> </PartNumber>
  <LastModified> </LastModified>
  <Etag> </Etag>
  <Size> </Size>
</Part>
</ListPartsResult>
```

示例

请求

GET

/coss3/test10M_2?uploadId=14846420620b

1f381e5d7b057692e131dd8d72dfa28f2633cfbbe4d0a9e8bd0719933545b0&max-parts=1 HTTP/1.1

Host:burning-1251668577.cn-east.myqcloud.com

Authorization:q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3MqQCn&q-sign-time=1484643123;1484646723&q-key-time=1484643123;1484646723&q-header-list=host&q-url-param-list=max-parts;uploadId&q-signature=b8b4055724e64c9ad848190a2f7625fd3f9d3e87

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 661

Connection: keep-alive

Date: Tue Jan 17 16:52:08 2017

x-cos-request-id: NTg3ZGRiMzhfMmM4OGY3XzdhY2NfYw==

```
<ListPartsResult>
  <Bucket>burning</Bucket>
  <Encoding-type/>
  <Key>test10M_2</Key>
  <UploadId>
148464206
20b1f381e5d7b057692e131dd8d72dfa28f2633cfbbe4d0a9e8bd0719933545b0</UploadId>
  <Initiator>
  <UID/>
  </Initiator>
  <Owner>
  <UID>1251668577</UID>
  </Owner>
  <PartNumberMarker>0</PartNumberMarker>
  <Part>
  <PartNumber>1</PartNumber>
  <LastModified>Tue Jan 17 16:43:37 2017</LastModified>
  <ETag>"a1f8e5e4d63ac6970a0062a6277e191fe09a1382"</ETag>
  <Size>5242880</Size>
  </Part>
  <NextPartNumberMarker>1</NextPartNumberMarker>
  <StorageClass>Standard</StorageClass>
  <MaxParts>1</MaxParts>
  <IsTruncated>true</IsTruncated>
</ListPartsResult>
```

Complete Multipart Upload

功能描述

Complete Multipart Upload用来实现完成整个分块上传。当您已经使用Upload Parts上传所有块以后，您可以使用该API完成上传。在使用该API时，您必须在Body中给出每一个块的PartNumber和ETag，用来校验块的准确性。

由于分块上传的合并需要数分钟时间，因而当合并分块开始的时候，COS就立即返回200的状态码，在合并的过程中，COS会周期性的返回空格信息来保持连接活跃，直到合并完成，COS会在Body中返回合并后块的内容。

当上传块小于1 MB的时候，在调用该请求时，会返回400

EntityTooSmall；当上传块编号不连续的时候，在调用该请求时，会返回400

InvalidPart；当请求Body中的块信息没有按序号从小到大排列的时候，在调用该请求时，会返回400

InvalidPartOrder；当UploadId不存在的时候，在调用该请求时，会返回404 NoSuchUpload。

建议您及时完成分块上传或者舍弃分块上传，因为已上传但是未终止的块会占用存储空间进而产生存储费用。

请求

请求语法

POST /ObjectName?uploadId=UploadId HTTP 1.1

Host: <BucketName>-<AppID>.<Region>.myqcloud.com

Date: date

Content-length: Size

Authorization: Auth

请求参数

无特殊请求参数

请求头部

无特殊请求头部

请求内容

名称	描述	类型	必选
CompleteMultipartUpload	用来说明本次分块上传的所有信息	Container	是
Part	用来说明本次分块上传中每个块的信息	Container	是
PartNumber	块编号	String	是
ETag	每个块文件的 SHA-1 算法校验值	String	是

<CompleteMultipartUpload>

<Part>

<PartNumber> </PartNumber>

<ETag> </ETag>

</Part>

...

</CompleteMultipartUpload>

返回值

返回头部

无特殊返回头部

返回内容

名称	描述	类型
CompleteMultipartUpload	说明所有返回信息	Container
Location	创建的Object的外网访问域名	URI

名称	描述	类型
Bucket	分块上传的目标Bucket	String
Key	Object的名称	String
ETag	合并后文件的 MD5算法校验值	String

```
<CompleteMultipartUpload>  
  <Location> </Location>  
  <Bucket> </Bucket>  
  <Key> </Key>  
  <ETag> </ETag>  
</CompleteMultipartUpload>
```

示例

请求

POST

```
/ObjectName?uploadId=1484728886e63106  
e87d8207536ae8521c89c42a436fe23bb58854a7bb5e87b7d77d4ddc48 HTTP/1.1  
Host:arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com  
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-  
time=1484729794;32557625794&q-key-time=1484729794;32557625794&q-header-list=host&q-url-  
param-list=uploadId&q-signature=23627c8fddb3823cce4257b33c663fd83f9f820d  
Content-Length: 155  
Content-Type: application/x-www-form-urlencoded
```

返回

```
HTTP/1.1 200 OK  
Content-Type: application/xml  
Content-Length: 277  
Connection: keep-alive  
Date: Wed Jan 18 16:58:13 2017
```

Server: tencent-cos

x-cos-request-id: NTg3ZjJlMjVfNDYyMDRlXzM0YzRfMjc1

<CompleteMultipartUpload>

<Location>arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com/ObjectName</Location>

<Bucket>arlenhuangtestsgnoverion</Bucket>

<Key>ObjectName</Key>

<ETag>"3a0f1fd698c235af9cf098cb74aa25bc"</ETag>

</CompleteMultipartUpload>

Abort Multipart Upload

功能描述

Abort Multipart Upload用来实现舍弃一个分块上传并删除已上传的块。当您调用Abort Multipart Upload时，如果有正在使用这个Upload Parts上传块的请求，则Upload Parts会返回失败。当该UploadID不存在时，会返回404 NoSuchUpload。

建议您及时完成分块上传或者舍弃分块上传，因为已上传但是未终止的块会占用存储空间进而产生存储费用。

请求

请求语法

```
DELETE /ObjectName?uploadId=UploadId HTTP 1.1
Host: <BucketName>-<AppID>.<Region>.myqcloud.com
Date: date
Authorization: Auth
```

请求参数

无特殊请求参数

请求头部

无特殊请求头部，其他头部请参见公共请求头部

请求内容

无请求内容

返回值

返回头部

无特殊返回头部，其他头部请参见公共返回头部

返回内容

无返回内容

示例

请求

DELETE

/ObjectName?uploadId=1484727270323

d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e HTTP/1.1

Host:arlenhuangtestsgnoverion-1251668577.sg.myqcloud.com

Authorization:q-sign-algorithm=sha1&q-

ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj

fGFO&q-sign-time=1484728626;32557624626&q-key-time=1484728626;32557624626&q-header-list=host&q-url-param-list=uploadId&q-signature=2d3036b57cade4a257b48a3a5dc922779a562b18

返回

HTTP/1.1 200 OK

Content-Type: application/xml

Content-Length: 0

Connection: keep-alive

Date: Wed Jan 18 16:37:13 2017

Server: tencent-cos

x-cos-request-id: NTg3ZjI5MzlfOTgxZjRlXzZhYjNmMjBh