

腾讯云对象存储

JSON 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.....	4
API概览.....	5
错误码.....	6
签名算法.....	16
使用示例.....	23
调用方式.....	29
请求结构.....	29
返回结构.....	31
错误码.....	33
目录接口.....	46
创建目录.....	46
列出目录.....	49
查询目录属性.....	56
删除目录.....	58
文件接口.....	60
简单上传文件.....	60
初始化分片上传.....	64
逐个上传分片.....	68
结束上传分片.....	72
移动文件.....	76
查询文件属性.....	79
查询上传分片.....	83
更新文件属性.....	89
删除文件.....	92
下载文件.....	94
复制文件.....	96

## 简介

### 描述

对象存储服务的 RESTful API 是一种轻量级的，无连接状态的，可以直接通过 http/https 的请求和响应，来和腾讯云对象存储（COS）后台进行交互操作的 Web 服务。

在查阅其他的 API 文档之前，首先请仔细阅读 调用方式 和 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 服务时每个项目拥有的唯一资源标识。
SecretID	SecretID 是开发者拥有的项目身份识别 ID。
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。

### 快速入门

要使用对象存储 API，需要先执行以下步骤：

1. 在控制台 创建一个 Bucket
2. 获取 Appid、SecretID、SecretKey 内容
3. 编写一个 签名鉴权 算法程序（或使用任何一种服务端 SDK）
4. 计算签名，调用 API 执行操作

## API概览

操作类型	功能	请求方式	详细说明
目录操作	创建目录	POST	<a href="#">创建目录API说明</a>
目录操作	列出目录	GET	<a href="#">列出目录API说明</a>
目录操作	查询目录属性	GET	<a href="#">查询目录属性API说明</a>
目录操作	删除目录	POST	<a href="#">删除目录API说明</a>
文件操作	简单上传文件	POST	<a href="#">简单上传文件API说明</a>
文件操作	初始化分片上传	POST	<a href="#">初始化分片上传API说明</a>
文件操作	逐个上传分片	POST	<a href="#">逐个上传分片上传API说明</a>
文件操作	结束上传分片	POST	<a href="#">结束上传分片文件API说明</a>
文件操作	查询文件属性	GET	<a href="#">查询文件属性API说明</a>
文件操作	查询上传分片	GET	<a href="#">查询上传分片API说明</a>
文件操作	更新文件属性	POST	<a href="#">更新文件属性API说明</a>
文件操作	删除文件	POST	<a href="#">删除文件API说明</a>
文件操作	移动文件	POST	<a href="#">移动文件API说明</a>
文件操作	下载文件	GET	<a href="#">下载文件API说明</a>

历史版本API，请点击查看[历史版本文档](#)

## 错误码

错误码	描述
-5999	请求参数错误
-5997	内部错误，联系官方处理
-5996	HTTP请求方法错误，或者没有发送HTTP请求方法
-5994	url路径不符合规范
-5993	formdata body格式不正确
-5991	append接口，大于5G的追加上传
-5980	"没有op字段，用户没填写操作类型或格式错误解析不到"
-5975	请求的json格式不正确
-5971	分片上传init时没有填写文件大小
-5970	refer白名单数量必须小于等于10
-5969	cname数量必须小于等于10
-5968	refer黑名单数量必须小于等于10
-5967	密钥数量必须小于等于10
-5966	list参数非数字或者大于1000
-5965	分片上传init时未指定slice_size参数
-5965	分片上传init时uploadparts参数不合法
-5964	参数op填写错误，没有此操作
-5963	读取数据超时，检查网络是否过慢或上传并发数过大
-5962	XML解析失败，请求携带的XML不符合规范
-5961	请求body体的Content-MD5不符合预期
-5957	操作不存在的bucket
-5949	copy文件时查询源文件信息失败
-5947	biz_attr最大4K
-5946	缺少头域CONTENT_MD5，有些接口必须校验内容
-5945	内部错误，联系官方处理
-5939	内部错误，联系官方处理
-5938	内部错误，联系官方处理
-5937	内部错误，联系官方处理
-4020	文件已存在，但未完成上传，新传文件sha值和该未完成文件不一致
-4019	文件已存在，但未完成上传，新传文件sha值和该未完

错误码	描述
	成文件一致，可断点续传
-4018	文件已存在，并且已完成上传
-4016	上传文件数据sha与请求sha校验失败
-288	内部错误，联系官方处理
-287	内部错误，联系官方处理
-285	上传文件大小大于限制
-284	上传文件数据请求每片数据大小与初始化的分片大小不同
-283	分片上传分片小于限制，或者append文件小于4K
-281	分片上传分片大于限制
-197	该文件不存在
-196	内部错误，联系官方处理
-195	内部错误，联系官方处理
-194	内部错误，联系官方处理
-185	内部错误，联系官方处理
-184	内部错误，联系官方处理
-183	内部错误，联系官方处理
-181	内部错误，联系官方处理
-180	路径错误，可能是url超过长度限制，非法字符等原因
-178	路径冲突，要上传的文件或者目录已经存在
-177	文件已存在，但是和新要上传的文件sha值不一致，可用覆盖上传覆盖掉原来的文件
-176	创建的bucket已经存在
-175	上传的文件大小和指定的content-length不一致
-173	删除非空的bucket
-172	bucket数量超过200个限制
-134	bucket不支持跨域访问
-133	bucket不存在
-121	"匹配跨域规则失败，可能 Origin / request method / Access-Control-Request-Method 或 Access-Control-Request-Headers不在跨域规则配置里面"
-120	uploadid不存在
-119	内部错误，联系官方处理

错误码	描述
-118	上传块编号找不到
-117	分块上传finish时块编号不连续
-116	内部错误，联系官方处理
-101	"不需要拷贝,可能上次更新时间/ETag规则匹配问题"
-100	copy的前置条件冲突
-99	内部错误，联系官方处理
-98	内部错误，联系官方处理
-97	签名校验失败
-96	签名过期
-85	内部错误，联系官方处理
-83	内部错误，联系官方处理
-81	签名为空（需鉴权的操作提供签名为空）
-79	secret id不存在
-77	单次性签名已被使用过
-76	单次签名没有指定f参数
-75	内部错误，联系官方处理
-74	多次签名-过期时间为0
-73	单次签名-过期时间不为0
-71	"操作太频繁，请稍后再试"
-70	请求的appid和签名中的appid不匹配
-66	内部错误，联系官方处理 )
-64	secretid中的uid和签名中的appid不匹配
-63	请求中的uri与签名中的f不匹配
-62	secretid可能填错或联系官方处理
-59	s3签名过期
-58	内部错误，联系官方处理
-57	签名缺失必填字段
-56	签名中的header未排序
-55	用户没有此操作权限
-53	签名有效期区间设置过长，最多设置一个月
-29175	内部错误，联系官方处理
-29173	分块上传不应指定存储介质
-29172	内部错误，联系官方处理



错误码	描述
-29171	内部错误，联系官方处理
-29170	内部错误，联系官方处理
-29169	内部错误，联系官方处理
-29168	内部错误，联系官方处理
-29166	list分块请求没有携带UploadID
-29165	内部错误，联系官方处理
-29164	内部错误，联系官方处理
-29163	内部错误，联系官方处理
-29162	内部错误，联系官方处理
-29161	分片上传完成接口指定的Etag长度错误
-29159	结束上传分片请求body为空
-29156	内部错误，联系官方处理
-29155	内部错误，联系官方处理
-29150	内部错误，联系官方处理
-29148	内部错误，联系官方处理
-29147	JSON API接口md5并发上传时，读取分片的内容和上传内容不一致
-29146	内部错误，联系官方处理
-29145	内部错误，联系官方处理
-29143	上传文件MD5校验失败
-29141	分片上传init指定的filesize错误
-29140	内部错误，联系官方处理
-29138	v4分片上传，sha值校验失败
-29137	内部错误，联系官方处理
-29136	相同的文件请使用断点续传，不要删除后重传
-29134	初始化分片上传指定的分片大小不符合规范
-29133	内部错误，联系官方处理
-29132	内部错误，联系官方处理
-29131	内部错误，联系官方处理
-29130	内部错误，联系官方处理
-29129	下载文件时文件尚未完成上传
-29128	内部错误，联系官方处理
-29126	初始化分片上传未指定filesize参数

错误码	描述
-29125	内部错误，联系官方处理
-29124	内部错误，联系官方处理
-29123	内部错误，联系官方处理
-29119	内部错误，联系官方处理
-29118	内部错误，联系官方处理
-29117	内部错误，联系官方处理
-29115	内部错误，联系官方处理
-29114	内部错误，联系官方处理
-29112	内部错误，联系官方处理
-29111	内部错误，联系官方处理
-29110	请求json格式错误
-29109	内部错误，联系官方处理
-29108	内部错误，联系官方处理
-29107	内部错误，联系官方处理
-29106	内部错误，联系官方处理
-29105	JSON API上传的文件超过系统限制，限制为64G
-29104	结束上传分片未指定filesize参数
-29103	上传时sessionid传错
-29102	内部错误，联系官方处理
-29101	内部错误，联系官方处理
-29100	内部错误，联系官方处理
-29099	内部错误，联系官方处理
-29098	用户上传数据偏移量不是每块的整数倍错误
-29096	传的sessionid不合法
-29094	传的sessionid不合法
-29093	传的sessionid不合法
-29088	内部错误，联系官方处理
-29087	内部错误，联系官方处理
-29086	内部错误，联系官方处理
-29085	内部错误，联系官方处理
-29084	内部错误，联系官方处理
-29083	内部错误，联系官方处理
-29082	内部错误，联系官方处理

错误码	描述
-29081	内部错误，联系官方处理
-29080	内部错误，联系官方处理
-29070	内部错误，联系官方处理
-29069	内部错误，联系官方处理
-29066	内部错误，联系官方处理
-29063	分片md5不一致
-29044	内部错误，联系官方处理
-29043	文件不存在
-29037	If-Modified-Since用户头部If-Modified-Since时间格式错误
-29036	文件未上传完成，不允许下载
-29034	下载的offset大于文件的fsize
-29033	下载请求没有有效的range
-29031	内部错误
-29027	内部错误，联系官方处理
-29026	内部错误，联系官方处理
-29021	上传文件sessionid传输错误
-29015	HEADER字段中没有host或者host错误
-29012	内部错误，联系官方处理
-29011	内部错误，联系官方处理
-29010	内部错误，联系官方处理
-29003	内部错误，联系官方处理
-29002	内部错误，联系官方处理
-29001	内部错误，联系官方处理
-29000	内部错误，联系官方处理
101	内部错误，联系官方处理
100	内部错误，联系官方处理
-29993	内部错误，联系官方处理
-29992	内部错误，联系官方处理
-29991	内部错误，联系官方处理
-46642	appid不存在
-46641	多版本下的文件不存在
-46640	内部错误，联系官方处理

错误码	描述
-46639	内部错误，联系官方处理
-46638	内部错误，联系官方处理
-46637	内部错误，联系官方处理
-46636	内部错误，联系官方处理
-46635	内部错误，联系官方处理
-46634	内部错误，联系官方处理
-46632	追加文件长度为0
-46627	用户在黑名单里面
-46626	内部错误，联系官方处理
-46622	非法分片大小
-46621	文件被封禁，可能涉黄政治敏感信息被封禁
-46620	bucket被封禁，可能qps过高，恶意攻击被后台封禁
-46619	签名过期，可能生成签名机器的时间跟北京时间不一致，或者用户拿着签名过期串访问
-46618	签名校验失败，用户签名算法错误，密钥被禁用，appid和密钥对不上等错误
-46617	bucket的refer黑名单配置了访问的refer域名
-46616	bucket的refer白名单没有配置访问的refer域名
-46615	bucket禁止访问，可能您的bucket访问量过大或欠费或违反了政策法规导致的封禁
-46614	文件没有完成上传，只传了一部分，检查上传时候有无报错
-46613	自定义的biz_attr过大，目前最大4K
-46607	内部错误，联系官方处理
-46601	内部错误，联系官方处理
-46313	内部错误，联系官方处理
-46312	内部错误，联系官方处理
-46311	内部错误，联系官方处理
-46310	内部错误，联系官方处理
-46309	内部错误，联系官方处理
-46308	内部错误，联系官方处理
-46307	内部错误，联系官方处理
-46306	内部错误，联系官方处理

错误码	描述
-46305	内部错误，联系官方处理
-46304	内部错误，联系官方处理
-46303	内部错误，联系官方处理
-46302	内部错误，联系官方处理
-46301	内部错误，联系官方处理
-46212	内部错误，联系官方处理
-46211	内部错误，联系官方处理
-46210	内部错误，联系官方处理
-46209	上传时versionid不正确
-46208	相同的文件名已存在，并且是目录
-46207	内部错误，联系官方处理
-46206	内部错误，联系官方处理
-46205	内部错误，联系官方处理
-46204	跨域设置时候规则超长，最大100KB
-46203	跨域设置时候规则超长，最大100KB
-46202	内部错误，联系官方处理
-46201	内部错误，联系官方处理
-46200	批量删除时，数量过大，最多1000个
-46199	分块上传finish时，传入的各个分块的Etag值与系统对不上
-46198	在非追加类型的文件上追加数据
-46197	追加上传时，offset和cos上文件大小不一致
-46196	内部错误，联系官方处理
-46195	内部错误，联系官方处理
-46194	内部错误，联系官方处理
-46193	内部错误，联系官方处理
-46192	内部错误，联系官方处理
-46191	内部错误，联系官方处理
-46190	内部错误，联系官方处理
-46189	内部错误，联系官方处理
-46188	跨域设置时候规则超长，最大100KB
-46187	跨域设置的规则参数有误
-46186	设置跨域时标志有误

错误码	描述
-46185	自定义header超长，最大4KB
-46184	内部错误，联系官方处理
-46183	内部错误，联系官方处理
-46182	内部错误，联系官方处理
-46181	refer长度过长，最大3K
-46180	签名过长，最大2K
-46179	封禁文件时，标志传错
-46178	查询文件列表时，delimiter长度大于1
-46177	list时填的offset长度超过path长度限制
-46176	biz attr长度非法，最大4K
-46175	内部错误，联系官方处理
-46174	内部错误，联系官方处理
-46173	内部错误，联系官方处理
-46172	内部错误，联系官方处理
-46171	内部错误，联系官方处理
-46170	下载文件时，host值传的不对
-46169	内部错误，联系官方处理
-46168	内部错误，联系官方处理
-46158	内部错误，联系官方处理
-46105	内部错误，联系官方处理
-46104	内部错误，联系官方处理
-46103	内部错误，联系官方处理
-46102	内部错误，联系官方处理
-46101	内部错误，联系官方处理
-46031	内部错误，联系官方处理
-46030	内部错误，联系官方处理
-46026	内部错误，联系官方处理
-46025	内部错误，联系官方处理
-46024	内部错误，联系官方处理
-46023	内部错误，联系官方处理
-46022	内部错误，联系官方处理
-46021	内部错误，联系官方处理
-46020	内部错误，联系官方处理

错误码	描述
-46007	内部错误，联系官方处理
-46005	内部错误，联系官方处理
-46004	内部错误，联系官方处理
-46003	内部错误，联系官方处理
-46002	内部错误，联系官方处理
-46001	内部错误，联系官方处理
-50122	内部错误，联系官方处理

## 签名算法

### 签名鉴权

#### 为什么需要鉴权？

业务端购买腾讯云（对象存储）服务，来作为其业务组成的一部分（云存储或内容分发加速），并最终提供给用户。所以当用户访问腾讯云服务时，需要先征得业务端的“许可”。那么，用户如何才能向腾讯云证明自己已经得到“访问许可”了呢？

简单的说，腾讯云和业务端约定了一套规则，通过这套规则，业务端可以产生一个“钥匙”。当用户需要访问业务端购买的腾讯云服务时，业务端会将这个钥匙传递给用户，然后用户拿着这个钥匙去向腾讯云“证明”自己已经拿到了“访问许可”。腾讯云收到用户的请求后，会先鉴别“钥匙”是否有效，若有效，则允许访问云服务，若无效，则返回 Authorization 错误消息。

这套规则，就是腾讯云的“鉴权系统”，这把“钥匙”，就是下文要解释的“签名”。

#### 签名有哪几种？

签名本质上是一个加密的字符串，目前在腾讯云中，分为多次有效签名和单次有效签名：

**多次有效签名：**签名中部分不绑定文件 fileid，部分绑定文件 fileid，开启token防盗链的下载、文件简单上传、文件分片上传可以绑定fileid，同时支持前缀匹配。多次有效签名需要设置一个大于当前时间的有效期，有效期内此签名可多次使用，有效期最长可设置三个月。

**单次有效签名：**签名中绑定文件 fileid

，有效期必须设置为0，此签名只可使用一次，且只能应用于被绑定的文件或文件夹操作。

#### 哪些场景需要用到签名？

腾讯云对象存储（COS）对签名的适用场景做了如下限制：

场景	适用签名	签名中是否需要绑定 fileid
下载（不开启 token 防盗链）	不验证签名	——



场景	适用签名	签名中是否需要绑定 fileid
下载（开启 token 防盗链）	多次有效签名	可选
简单上传文件	多次有效签名	可选
分片上传文件	多次有效签名	可选
查询目录、文件属性	多次有效签名	否
创建目录	多次有效签名	否
删除目录、文件	单次有效签名	是
更新文件属性	单次有效签名	是
移动（重命名）文件	单次有效签名	是

## 如何使用签名来完成鉴权？

腾讯云对象存储通过签名来验证请求的合法性。业务端通过将签名授权给用户端，使其具备上传下载及管理指定资源的权限。

当用户使用 RESTful API 对腾讯云对象存储资源进行操作时，必须要在每个 Http/Https 请求的 Header 中填写 Host 和 Authorization，其中的 Authorization 参数，即为“签名算法”最终得到的 sign 字符串。

参数名称	必选	类型	描述
Host	是	String	文件云服务器域名，固定为 [region].file.myqcloud.com
Authorization	是	String	用户的有效签名，用于鉴权

## 签名算法

### 腾讯云的各类 SDK

已经提供了标准的签名计算方法，业务端只需填写相关参数，即可得到签名字符串。用户也可以根据 RESTful API 自行计算签名。

对于签名的计算过程，可简单分为如下三个步骤：

获取签名所需信息，拼接明文字符串，将明文字符串转化为签名。

### 获取签名所需信息

生成签名所需信息包括项目 ID ( App Id ) , 空间名称 ( Bucket , 文件资源的组织管理单元 ) , 项目的 Secret ID 和 Secret Key。获取这些信息的方法如下：

- 1) 登录腾讯云对象存储，进入对象存储空间；
- 2) 如开发者未创建空间，可添加空间，空间名称 ( Bucket ) 由用户自行输入；
- 3) 点击【获取 API 密钥】，获取 App Id , Secret ID 和 Secret Key。

### 拼接明文字符串 Original

明文字符串 Original 按照签名的类型可划分为“多次”和“单次”有效签名。

拼接多次有效签名串 multi\_effect\_signature：

Original = "a=[appid]&b=[bucket]&k=[SecretID]&e=[expiredTime]&t=[currentTime]&r=[rand]&f="

拼接单次有效签名串 once\_signature：

Original =

"a=[appid]&b=[bucket]&k=[SecretID]&e=[expiredTime]&t=[currentTime]&r=[rand]&f=[fileid]"

签名串中各字段含义如下：

字段	解释
a	开发者的项目 ID，接入对象存储服务创建空间时系统生成的唯一标识项目的 ID，即 App Id
b	空间名称 Bucket
k	项目的 Secret ID
t	当前时间戳，是一个符合 Unix Epoch 时间戳规范的数值，单位为秒
e	多次有效签名时，e 为签名的失效时刻，是一个符合 Unix Epoch 时间戳规范的数值，单位为秒。e 的计算方式为

字段	解释
	$e = t +$ 签名有效时长。签名有效时长最大取值为7776000 (90天)；单次签名时， $e$ 必须设置为0
r	随机串，无符号10进制整数，用户需自行生成，最长10位
f	fileid，唯一标识存储资源的相对路径。格式为 /appid/ bucketname/dir name/[filename]，并且需要对其中非 '/' 字符进行 UrlEncode 编码。当操作对象为文件夹时，filename 缺省。filename 中要包含文件后缀名。

注意：

当签名类型需要绑定 fileid 时，f 字段请按要求赋值。

拼接单次有效签名的签名串时，过期时间  $e$  必须设置为0，以保证此签名只能针对固定资源使用一次；

拼接多次有效签名的签名串时，过期时间  $e$  的单位为秒，不同编程语言获得的系统 Unix 时间戳单位可能有所差异（比如 java 是毫秒），但都请转化为秒。

多次/单次有效签名的具体适用场景参见 [哪些场景需要用到签名？](#)

## 将明文字符串转化为签名

拼接好签名的明文字符串 Original 后，用已经获取的 SecretKey 对明文串进行 HMAC-SHA1 加密，得到 SignTmp：

$SignTmp = \text{HMAC-SHA1}(\text{SecretKey}, \text{Original})$

将密文串 SignTmp 放在明文串 Origin 前面，拼接后进行 Base64Encode 算法，得到最终的签名 Sign：

$Sign = \text{Base64}(\text{append}(\text{SignTmp}, \text{Original}))$

注 1：此处使用的是标准的 Base64 编码，不是 urlsafe 的 Base64 编码。

注 2：由于使用了 HMAC 算法，计算 SignTmp 的结果为二进制字符串，因此建议将算法写在同一函数中实现。单独输出 SignTmp 可能导致拼接后的字符串有误。

## 示例

本节介绍生成签名的算法实例，实例中使用 PHP 语言（若开发者使用其他开发语言，请使用对应的函数）。

### 获取签名所需信息

获取得到的签名所需信息如下

项目 ID：200001

空间名称 Bucket：newbucket

Secret ID：AKIDUfLUEUigQiXqm7CVSspKJnuaiIKtxqAv

Secret Key：bLcPnl88WU30VY57ipRhSePfPdOfSruK

### 拼接签名串

拼接的多次有效签名串 multi\_effect\_signature 如下：

```
a=200001&b=newbucket&k=AKIDUfLUEUigQiXqm7CVSspKJnuaiIKtxqAv&e=1437995704&t=1437995644&r=2081660421&f=
```

拼接的单次有效签名串 once\_signature 如下：

```
a=200001&b=newbucket&k=AKIDUfLUEUigQiXqm7CVSspKJnuaiIKtxqAv&e=0&t=1437995645&r=166710792&f=/200001/newbucket/tencent_test.jpg
```

```
$appid = "200001";
$bucket = "newbucket";
$secret_id = "AKIDUfLUEUigQiXqm7CVSspKJnuaiIKtxqAv";
$secret_key = "bLcPnl88WU30VY57ipRhSePfPdOfSruK";
$expired = time() + 60;
$onceExpired = 0;
$current = time();
$rdm = rand();
$fileid = "/200001/newbucket/tencent_test.jpg";

$multi_effect_signature = 'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$expired.'&t='
.$current.'&r='.$rdm.'&f=';

$once_signature=
'a='.$appid.'&b='.$bucket.'&k='.$secret_id.'&e='.$onceExpired.'&t='.$current.'&r='.$rdm.'&f='
.$fileid;
```

## 生成签名

```
$multi_effect_signature = base64_encode(hash_hmac('SHA1', $multi_effect_signature, $secret_key,
true).$multi_effect_signature);

$once_signature = base64_encode(hash_hmac('SHA1', $once_signature, $secret_key,
true).$once_signature);

echo $multi_effect_signature."\n";

echo $once_signature."\n";
```

最终得到的多次有效签名为：

```
v6+um3VE3lxGz97PmnSg6+/V9PZhPTIwMDAwMSZiPW5ld2J1Y2tldCZrPUFLSURVZkxVRVVPzZ1FpWH
```

FtN0NWU3NwS0pudWFpSUt0eHFBdiZIPTE0NzA3MzcwMDAmdD0xNDcwNzM2OTQwJnI9NDkwMjU4OTQzJmY9

单次有效签名为：

CkZ0/gWkHy3f76ER7k6yXgzq7w1hPTIwMDAwMSZiPW5ld2J1Y2tldCZrPUFLSURVZkxVRVpZ1FpWHFtN0NWU3NwS0pudWFpSUt0eHFBdiZIPTAmdD0xNDcwNzM2OTQwJnI9NDkwMjU4OTQzJmY9LzIwMDAwMS9uZXdidWNrZXQvdGVuY2VudF90ZXN0LmpwZw==

## 使用示例

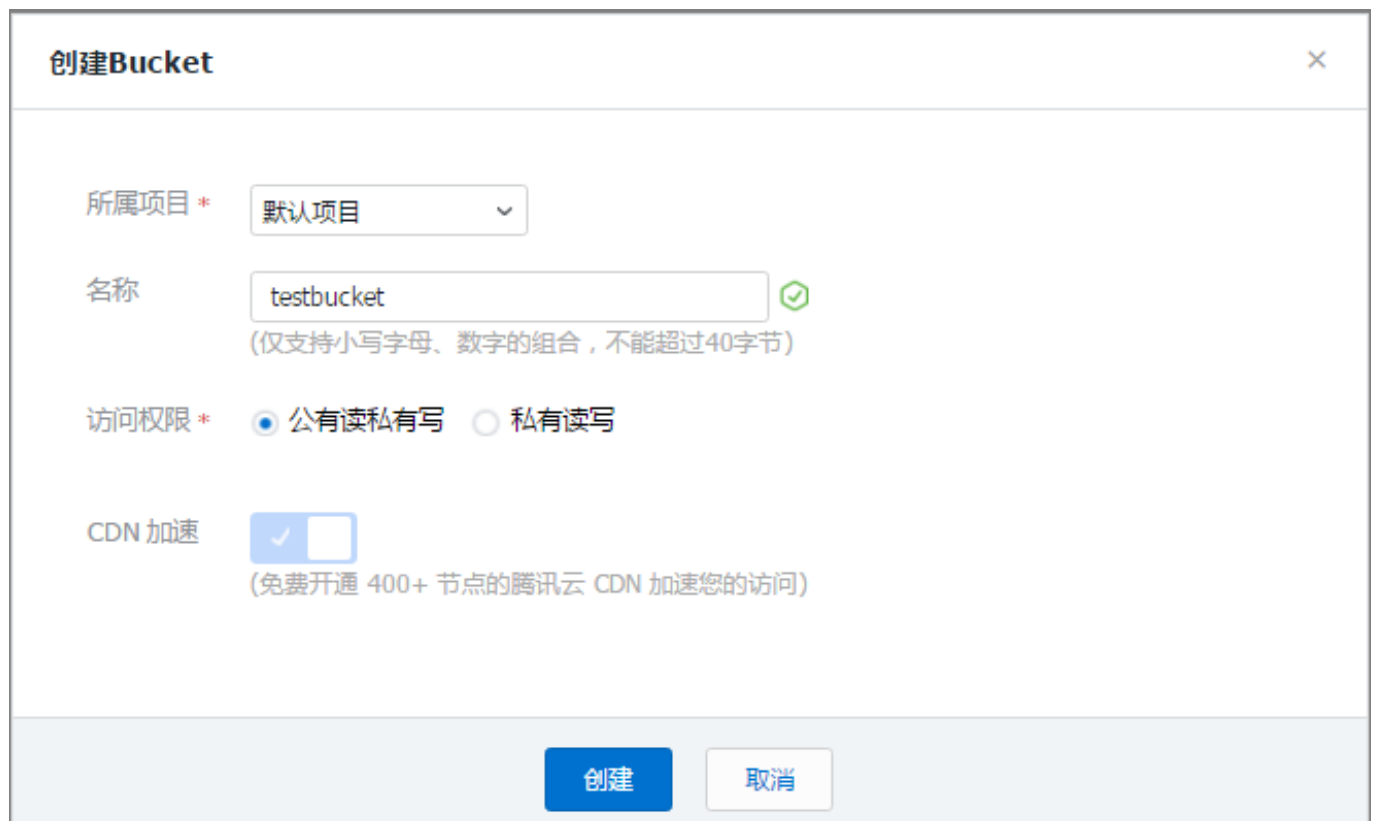
### 描述

本篇示例将实现三个简单的对象存储接口：创建目录，简单上传文件，删除文件。

### 准备工作


在任何文件上传至 COS 之前，都需要创建 Bucket 来存放文件。

进入 COS 管理控制台，点击 创建 Bucket ，弹出如下对话框：



创建Bucket

所属项目 \* 默认项目

名称 testbucket   
(仅支持小写字母、数字的组合, 不能超过40字节)

访问权限 \*  公有读私有写  私有读写

CDN 加速   
(免费开通 400+ 节点的腾讯云 CDN 加速您的访问)

创建 取消

点击 创建 ，即可在页面 Bucket 列表中看到创建的 Bucket ：



## 使用 RESTful API

您至少需要一个支持发起 HTTP RESTful 请求的客户端，例如 curl 或其他相关函数来调用 API。

## 示例

说明：以华东地区的请求为例

### 创建目录

在 buckettest 创建一个 foldertest 目录。

请求

POST /files/v2/10000202/buckettest/foldertest/ HTTP/1.1

Host: sh.file.myqcloud.com

Content-Type: application/json

Content-Length: 15

Authorization: 1k2i3/EXIiTDirFg9DoKgWNHc4JhPTEwMDAwMjAyJms9QUtJRFBOUHVyNUIyN3FjdVJh  
akNFbXpLVjkzVTdrOFZjZXFjXmU9MTQ2NTg3NTU0OSZ0PTE0NjU4NzUzNjkmcj03MTI5NDYyMzQmZj  
0mYj1qb25ueHU1

```
{
  "op": "create"
}
```



返回

HTTP/1.1 200 OK

Server: nginx

Date: Tue, 14 Jun 2016 03:36:12 GMT

Content-Length: 109

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "ctime": "1465875372",
    "resource_path": "/10000202/buckettest/foldertest/"
  }
}
```

## 简单上传文件

在 buckettest 的 foldertest 目录下上传一个 test\_upload.pptx 文件。

请求

POST /files/v2/10000202/buckettest/foldertest/test\_upload.pptx HTTP/1.1

Host: sh.file.myqcloud.com

Accept: /

Authorization: 5iMIxdTSSxBlikBTnUr+cIUZcTZhPTEwMDAwMjAyJms9QUtJRFB0UHVyNUIyN3FjdVJha  
kNFbXpLVjkzVTdrOFZjZXFXJmU9MTQ2NTg3NTU1MyZ0PTE0NjU4NzUzNzMmcj0yMDEwOTAxNjkwJm  
Y9JmI9am9ubnh1NQ==

Content-Type: multipart/form-data; boundary=-----8d3944816ef2585

Content-Length: 78963

-----8d3944816ef2585

Content-Disposition: form-data; name="op"

upload

-----8d3944816ef2585

Content-Disposition: form-data; name="sha"

D80DFA67880831C3691AA1458589C6BED4423736

-----8d3944816ef2585

Content-Disposition: form-data; name="insertOnly"

0

-----8d3944816ef2585

Content-Disposition: form-data; name="fileContent"; filename="test\_upload.pptx"

Content-Type: application/octet-stream

返回

HTTP/1.1 200 OK

Server: nginx

Date: Tue, 14 Jun 2016 03:36:13 GMT

Content-Type: /

Content-Length: 330

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "access_url": "http://buckettest-10000202.file.myqcloud.com/foldertest/test_upload.pptx",
```

```
"resource_path": "/foldertest/test_upload.pptx",
"source_url": "http://buckettest-10000202.cos.myqcloud.com/foldertest/test_upload.pptx",
"url": "http://region.file.myqcloud.com/files/v2/foldertest/test_upload.pptx"
}
}
```

## 删除文件

### 请求

```
POST /files/v2/10000202/buckettest/foldertest/test_upload.pptx HTTP/1.1
Host: sh.file.myqcloud.com
Content-Type: application/json
Content-Length: 15
Authorization: g4THINbztAH/beX6Pmf00q8T++thPTEwMDAwMjAyJms9QUtJRFBOUHVyNUIyN3FjdVJhakNFbXpLVjkzVTdrOFZjZXFJmU9MCZ0PTE0NjU4NzU0MzMzEmcj0yNTk2NTAyNDkmZj0vMTAwMDAyMDIvbm9ubnh1NS9mb2xkZXJ0ZXN0L3Rlc3RfdXBsb2FkLnBwdHgmYj1qb25ueHU1
```

```
{
  "op": "delete"
}
```

### 返回

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 14 Jun 2016 03:37:11 GMT
Content-Type: /
Content-Length: 30
```

```
{
  "code": 0,
  "message": "SUCCESS"
}
```

```
}
```

## 调用方式

### 请求结构

### 描述

腾讯云对象存储的 RESTful API 请求结构如下表：

参数名称	必选	类型	描述
Version	是	String	HTTP 版本
URL	是	String	HTTP 请求地址，总长度不得超过 1024 字节
Header	是	数据集合	HTTP 请求头部
Http Method	是	String	HTTP 请求方法，常用 GET 或 POST
Request Body	否	json	HTTP 请求正文

URL 请求地址格式举例：

```
<Region>.file.myqcloud.com/files/v2/<appid>/<bucket_name>[/dir_name/<file_name>]
```

说明：

Region：sh（上海），gz（广州），tj（天津）

appid：数字，在腾讯云对象存储 Web 控制台中，创建项目后得到 appid，用于唯一标识某个项目。

bucket\_name：字母和数字的组合串。对应某个项目下的 bucket 名称。

<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数，具体要求以各 API 文档为准。

Header 结构：

参数名称	必选	类型	描述
Host	是	String	文件云服务器域名，固定填 web.file.myqcloud.com
Authorization	是	String	用户的有效签名，用于鉴权。对应签名鉴权中得到的 sign 字符串
Content-Type	是	String	按要求填 application/json 或 multipart/form-data
Content-Length	是	Int	请求长度，单位：字节

## 示例

请求

POST /files/v2/10000202/buckettest/foldertest/ HTTP/1.1

Host: sh.file.myqcloud.com

Content-Type: application/json

Content-Length: 15

Authorization: 1k2i3/EXIiTDirFg9DoKgWNHc4JhPTEwMDAwMjAyJms9QUtJRFBOUHVyNUIyN3FjdVJh  
akNFbXpLVjkzVTdrOFZjZXFXJmU9MTQ2NTg3NTU0OSZ0PTE0NjU4NzUzNjkmcj03MTI5NDYyMzQmZj  
0mYj1qb25ueHU1

```
{
  "op": "create"
}
```

## 返回结构

### 描述

腾讯云对象存储的 RESTful API 统一返回 HTTP 标准的状态码，HTTP 标准的头部信息以及 json 格式的返回包内容。

常见的返回格式如下：

参数名称	类型	描述
HTTP Status Code	Int	HTTP 状态码
HTTP Headers	String	HTTP 头部信息
Response Body	数据集合	Http 响应正文

### Response Body 数据集合

参数名称	类型	描述
code	Int	服务端返回码
message	String	服务端提示内容
data	数据集合	服务端返回的应答数据

## 示例

### 返回

HTTP/1.1 200 OK

Server: nginx

Date: Tue, 14 Jun 2016 03:36:12 GMT

Content-Length: 109

```
{  
  "code": 0,  
  "message": "SUCCESS",  
  "data": {  
  }  
}
```

```
}
```



## 错误码

### 服务端 SDK 错误码

- 1 指定的上传文件不存在
- 2 网络请求失败（一般是用户主机和我们的服务无法连通）
- 3 参数错误
- 4 分片大小非法

### 下载状态码

- 200 正常
- 206 (部分内容) 服务器成功处理了部分 GET 请求
- 403 鉴权错误或者命中了refer黑名单
- 404 文件不存在
- 405 请求方法不能被用于请求相应的资源。
- 500 内部服务器错误
- 6101 文件不存在
- 45062 文件未上传完成

### 后端HTTP返回码

- 400 HTTP\_BAD\_REQUEST 请求不合法，包体格式错误

401 HTTP\_UNAUTHORIZED 权限验证失败

403 HTTP\_FORBIDDEN 鉴权信息不合法，禁止访问

413 HTTP\_REQUEST\_ENTITY\_TOO\_LARGE 文件大小超长

500 HTTP\_INTERNAL\_SERVER\_ERROR 服务内部错误

503 HTTP\_SERVICE\_UNAVAILABLE 服务不可用

504 HTTP\_GATEWAY\_TIME\_OUT 后端服务超时

599 HTTP\_UNKOWN\_INTERNAL\_SERVER\_ERROR 未知服务器内部错误

612 HTTP\_FILE\_NOTEXIST 查无此文件

631 HTTP\_INVALID\_APPID 非法的业务ID

## 业务返回码

\_SUCCESS 0 成功

-9999 UNKOWN 未知错误

## PROXY 错误码

-99 ERROR\_PROXY\_GET\_CONFIG proxy读取配置失败

-98 ERROR\_PROXY\_AUTH\_INVOKE 调用签名服务失败

-97 ERROR\_PROXY\_AUTH\_FAILED 非法签名

-96 ERROR\_PROXY\_AUTH\_EXPIRED 签名过期

- 95 ERROR\_PROXY\_NO\_SESSION 消息缺少session信息
- 94 ERROR\_PROXY\_WRONG\_SESSION 携带错误session信息
- 89 ERROR\_PROXY\_ROUTE\_CMD proxy转发cmd服务失败
- 88 ERROR\_PROXY\_ENCODE\_CMD 编码cmd服务消息失败
- 87 ERROR\_PROXY\_DECODE\_CMD 解析cmd服务消息失败
- 86 ERROR\_PROXY\_ROUTE\_PROCESS proxy转发process服务失败
- 85 ERROR\_PROXY\_DECODE\_PROCESS\_RSP 解析process服务应答失败
- 84 ERROR\_PROXY\_GET\_PROCESS\_L5 获取processL5失败
- 83 ERROR\_PROXY\_AUTH\_UNPACK 签名服务解包失败
- 82 ERROR\_PROXY\_AUTH\_APPID\_NOEXIST 不存在此appid
- 81 ERROR\_PROXY\_AUTH\_SIGN\_EMPTY 签名为空
- 80 ERROR\_PROXY\_INVALID\_APPID 非法的业务ID
- 79 ERROR\_PROXY\_AUTH\_SECRETID\_NOEXIST secretid不存在
- 78 ERROR\_PROXY\_INVALID\_PROTOCOL SDK协议不匹配,请升级
- 77 ERROR\_PROXY\_AUTH\_REPLAY\_ATTACH 一次性签名已不可用
- 76 ERROR\_PROXY\_AUTH\_ONCESIGN\_NOFILEID 单次签名没有url
- 75 ERROR\_PROXY\_AUTH\_UNSUPPORTED\_OP 不支持此操作

- 74 ERROR\_PROXY\_AUTH\_MULTISIGN\_EXPIREEMPTY 多次签名-过期时间为0
- 73 ERROR\_PROXY\_AUTH\_ONCESIGN\_EXPIREDNOTEMPTY 单次签名-过期时间不为0
- 72 ERROR\_PROXY\_APP\_SIGN\_FAILED 签名失败
- 71 ERROR\_PROXY\_FREQ\_CONTROL 操作太频繁,请稍后再试
- 70 ERROR\_PROXY\_APPID\_USERID\_NOTMATCH 输入的appid与签名不匹配
- 69 ERROR\_PROXY\_INVALID\_PARAM\_DOWNLOAD\_URL\_EMPTY 输入参数错误:download\_urlempty
- 68 ERROR\_PROXY\_ENCODE ip直通车打包失败
- 67 ERROR\_PROXY\_DECODE ip直通车解包失败
- 66 ERROR\_PROXY\_NETWORK 网络请求失败
- 65 ERROR\_PROXY\_CONTINUE\_TRANSFER\_WITH\_DATA 断点续传不支持携带数据包
- 64 ERROR\_PROXY\_FORBID\_APPID 该业务已经被屏蔽
- 63 ERROR\_PROXY\_FILEID\_NOTMATCH fileid与签名不匹配
- 62 ERROR\_PROXY\_AUTH\_APPID appid与secretid不匹配
- 61 ERROR\_PROXY\_SIGN\_BUCKET\_NOTMATCH bucket与签名中的bucket不匹配
- 61 ERROR\_PROXY\_INVALID\_CMD\_REQ 非法的CMD请求
- 60 ERROR\_PROXY\_AUTH\_INVALID\_TOKEN 非法token值

## CMD 错误码

- 199 ERROR\_CMD\_FILE\_MOVE\_ERROR 文件移动失败
- 198 ERROR\_CMD\_REDIRECT\_ERROR 重定向错误
- 197 ERROR\_CMD\_FILE\_NOTEXIST 查无此文件
- 196 ERROR\_CMD\_BACKEND\_NETWORK 网络请求失败
- 195 ERROR\_CMD\_BACKEND\_RSP\_DECODE 后端解包失败
- 194 ERROR\_CMD\_BACKEND\_REQ\_ENCODE 后端打包失败
- 193 ERROR\_CMD\_RSP\_ENCODE 返回包打包失败
- 192 ERROR\_CMD\_REQ\_DECODE 请求包解析失败
- 191 ERROR\_CMD\_URL\_PARAMS\_NOTMATCH url参数解析不匹配
- 190 ERROR\_CMD\_FILE\_DEL\_ERROR 文件删除失败
- 189 ERROR\_CMD\_INVALID\_PARAM\_DOWNLOAD\_URL\_EMPTY 输入参数错误:download\_urlempty
- 187 ERROR\_CMD\_URL\_PARSE\_ERROR 从url中解析参数失败
- 186 ERROR\_CMD\_VIDEO\_MOVE 暂不支持视频文件复制
- 185 ERROR\_CMD\_PRE\_PROCESS 业务预处理失败
- 184 ERROR\_CMD\_POST\_PROCESS 业务后处理失败
- 183 ERROR\_CMD\_GET\_ROUTE 获取路由失败
- 182 ERROR\_CMD\_PARAM\_INVALID 参数检验失败

- 181 ERROR\_CMD\_COS\_ERROR 存储后端错误
- 180 ERROR\_CMD\_COS\_INVALID\_PATH 非法路径
- 179 ERROR\_CMD\_COS\_INVALID\_MOD\_FLAG 修改标志非法
- 178 ERROR\_CMD\_COS\_PATH\_CONFLICT 路径冲突
- 177 ERROR\_CMD\_COS\_FILE\_EXIST 文件已存在
- 176 ERROR\_CMD\_COS\_BUCKET\_EXIST Bucket已存在
- 175 ERROR\_CMD\_COS\_FILE\_SIZE\_NOT\_EQU 文件大小不一致
- 174 ERROR\_CMD\_COS\_SHA\_NOT\_EQU 文件SHA不一致
- 173 ERROR\_CMD\_COS\_DIR\_NOT\_EMPTY 目录非空
- 172 ERROR\_CMD\_COS\_BUCKET\_NUM\_LIMIT Bucket数量限制
- 171 ERROR\_CMD\_COS\_INVALID\_APP\_TYPE AppType非法
- 170 ERROR\_CMD\_COS\_INVALID\_LIST\_NUM LIST数量非法
- 169 ERROR\_CMD\_COS\_INVALID\_QUERY\_FLAG 查询标记非法
- 168 ERROR\_CMD\_COS\_INVALID\_SHA 非法SHA
- 167 ERROR\_CMD\_COS\_INVALID\_CNAME 非法的CNAME
- 166 ERROR\_CMD\_COS\_INDEX\_ERROR 索引不存在
- 165 ERROR\_CMD\_COS\_MAX\_NUM 单次拉取目录最大支持199

- 164 ERROR\_CMD\_PARAM\_PATH\_EMPTY path为空
- 163 ERROR\_CMD\_PARAM\_PATH\_ELEMENT\_MISSING path字段缺少元素
- 162 ERROR\_CMD\_PARAM\_PATH\_ELEMENT\_NOT\_MATCH path字段元素不匹配
- 161 ERROR\_CMD\_PARAM\_PATH\_DIR\_MISSING\_SLASH 目录path字段缺少slash结尾
- 160 ERROR\_CMD\_PARAM\_PAHT\_FILE\_SURPLUS\_SLASH 文件path多余slash结尾
- 159 ERROR\_CMD\_PARAM\_CNAME\_ERROR cname数量必须小于等于10
- 158 ERROR\_CMD\_PARAM\_AUTH\_TYPE 鉴权类别错误//鉴权类别错误
- 157 ERROR\_CMD\_PARAM\_MIN\_NUM 单次拉取条目最小为1//DirListReq中拉取数据条数
- 156 ERROR\_CMD\_INVALID\_VIDEO\_NAME 非法的视频名
- 155 ERROR\_CMD\_INVALID\_FORMAT 非法视频FORMAT
- 154 ERR\_CMD\_APP\_VIDEO\_NOMATCH 文件BUCKET中设置视频参数
- 153 ERR\_CMD\_INVALID\_VIDEO\_APP 非法的视频APP
- 152 ERR\_CMD\_COS\_TIMEOUT 存储后端超时
- 151 ERROR\_CMD\_PARAM\_REFERER\_ERROR refer数量必须小于等于10
- 150 ERROR\_CMD\_PARAM\_SOURCE\_DOMAIN\_EMPTY 创建bucket时设置的回源路径不能为空
- 149 ERROR\_CMD\_INVALID\_REFERER refer中含有非法字符
- 148 ERROR\_CMD\_INVALID\_CNAME cname中含有非法字符

-147 ERROR\_CMD\_NOT\_UTF8\_ENCODING 参数含非utf8编码

-146 ERROR\_CMD\_INVALID\_HOST\_NAME 回源地址不合法

-145 ERROR\_CMD\_INVALID\_COVER 是否需要封面参数不合法

-144 ERROR\_CMD\_INDEX\_NO\_EXIST 索引不存在

-143 ERROR\_CMD\_INDEX\_ERROR 索引错误

### 文件缓存的错误码

-300 ERROR\_PROCESS\_OVERLOAD 服务过载

-299 ERROR\_PROCESS\_UNKNOWN\_CMD 命令字未知

-298 ERROR\_PROCESS\_REQ2MSG 解包失败

-297 ERROR\_PROCESS\_FRAME 框架handleprocess错误

-296 2RSP 打包失败

-295 ERROR\_PROCESS\_WRITEFILE 文件数据异常

-294 ERROR\_PROCESS\_READFILE 文件数据异常

-293 ERROR\_PROCESS\_NOTIFY 通知插件失败

-292 ERROR\_PROCESS\_CREATEFILE 文件缓存服务器错误

-291 ERROR\_PROCESS\_ENCODE\_SESSION 编码session失败

-290 ERROR\_PROCESS\_OFFSET 无效的session



- 289 ERROR\_PROCESS\_DENY 插件拒绝上传
- 288 ERROR\_PROCESS\_ENCODE process打包失败
- 287 ERROR\_PROCESS\_DECODE process解包失败
- 286 ERROR\_PROCESS\_DECODE\_SESSION 解码session失败
- 285 ERROR\_PROCESS\_FILE\_TOO\_LARGE 文件过大
- 284 ERROR\_PROCESS\_WRONG\_SLICE\_SIZE 分片大小不一致
- 283 ERROR\_PROCESS\_SLICE\_TOO\_SMALL 分片过小
- 282 ERROR\_PROCESS\_ANALYZE 图片分析失败

## 文件插件错误码

- 4000 ErrDecodeReq 请求体解包失败
- 4001 ErrEncodeRsp 回包打包失败
- 4002 ErrParamInvalid 请求参数错误
- 4003 ErrNoSetErrCode 插件处理错误
- 4004 ErrCosRspHeadCode 入库出错
- 4005 ErrCosRspInvalidOffset 入库返回非法偏移
- 4006 ErrCosRspInvalidHeadFlag 入库返回非法的头部标记
- 4007 ErrTmemRspSsdError 获取TMEM配置错误

- 4008 ErrParserShmError 解析共享内存TMEM配置错误
  
- 4010 ErrCreateControlFile 创建控制文件错误
  
- 4011 ErrReadControlFile 读取控制文件错误
  
- 4012 ErrGetInputData 获取入库数据失败
  
- 4013 ErrDecodeSession 解session失败
  
- 4014 ErrQueryOffset 查询分片错误
  
- 4015 ErrTheSameOffset 同一个offset多次返回
  
- 4016 ErrNotSameSHA sha校验失败
  
- 4017 ErrHttpSvr 文件入库错误
  
- 4018 ErrSameFileUpload 相同文件已上传过
  
- 4019 ErrOffsetToZero 入库失败，请重新上传
  
- 4020 ErrParamInvalidCheckTypeShouldBeSHA 校验类型需要SHA
  
- 4021 ErrParamInvalidSessionDecodeError 参数session解码失败
  
- 4022 ErrParamInvalidFilePathInvalid 获取bucket路径失败
  
- 4023 ErrParamInvalidCheckSumSizeInvalid 校验和大小错误
  
- 4024 ErrOffGoBack 偏移量回退，请重试
  
- 4025 ErrHttpTimeout 文件入库超时

-4026 ErrOffsetBackTooMuch 索引被删除或SHA不一致，请重新上传

-4027 ErrOffsetRedo 索引错误，请重新上传

## CGI 错误码

-5999 ERROR\_CGI\_PARAM 参数错误

-5998 ERROR\_CGI\_AUTHORIZATION 签名格式错误

-5997 ERROR\_CGI\_NETWORK 后端网络错误

-5996 ERROR\_CGI\_REQ\_METHOD HTTP请求方法错误

-5995 ERROR\_CGI\_FILESIZE 文件大小错误

-5994 ERROR\_CGI\_URL\_NOTMATCH url参数解析不匹配'

-5993 ERROR\_CGI\_FORM\_ERR multipart/formdata参数错误

-5992 ERROR\_CGI\_REQ\_PARAM 请求参数错误

-5991 ERROR\_CGI\_SLICE 分片过大

-5990 ERROR\_CGI\_MISSING\_FILECONTENT 找不到filecontent

-5989 ERROR\_CGI\_UPLOAD\_FAIL 上传失败

-5988 ERROR\_CGI\_INIT cgi初始化错误

-5987 ERROR\_CGI\_WUP\_ENCODE wup编码失败

-5986 ERROR\_CGI\_WUP\_DECODE wup解码失败

- 5985 ERROR\_CGI\_NETWORK\_L5 获取路由失败
  
- 5984 ERROR\_CGI\_SHA\_NOT\_MATCH sha1不匹配
  
- 5983 ERROR\_CGI\_WRONG\_SESSION 错误的session
  
- 5982 ERROR\_CGI SOCK\_CREATE 建立连接错误
  
- 5981 ERROR\_CGI SOCK\_CONNECT 建立连接错误
  
- 5980 ERROR\_CGI\_PARAM\_MISSING\_OP 缺少参数:op
  
- 5979 ERROR\_CGI\_PARAM\_MISSING\_SHA 缺少参数:sha
  
- 5978 ERROR\_CGI\_PDU\_ENCODE pdu编码失败
  
- 5977 ERROR\_CGI\_PDU\_DECODE pdu解码失败
  
- 5976 ERROR\_CGI\_REQ\_PARAM\_FLAG\_NOTMATCH bucketflag与实际参数不一致
  
- 5975 ERROR\_CGI\_JSON\_PARSE\_ERROR json解析失败
  
- 5974 ERROR\_CGI\_VIDEO\_FORMAT 视频码率错误
  
- 5973 ERROR\_CGI\_CROSS\_BUCKET\_COPY 不允许跨存储空间复制
  
- 5972 ERROR\_CGI\_UNSUPPORTED\_METHOD 此方法不支持于当前版本
  
- 5971 ERROR\_CGI\_PARAM\_MISSING\_FILESIZE 缺少参数:filesize

## OSS 错误码

- 6999 ERROR\_OSS\_UNKNOWN 未知错误

-6998 ERROR\_OSS\_INVALID\_PROTOCOL 协议错误

-6997 ERROR\_OSS\_INVALID\_PARAM 参数错误

-6996 ERROR\_OSS\_FILESIZE 文件过大

-6995 ERROR\_OSS\_MKDIR 创建目录失败

-6994 ERROR\_OSS\_GZIP gzip压缩失败

-6993 ERROR\_OSS\_WRITE\_FILE 写文件失败

-6992 ERROR\_OSS\_FILE\_NOTEXIST 文件不存在

-6991 ERROR\_OSS\_READ\_FILE 读文件失败

## 目录接口

### 创建目录

#### 描述

使用 API 在 Bucket 中创建一个名为 dir\_name 的新目录。

前置条件：已经在控制台创建 Bucket。

#### 请求

##### 请求语法

POST /files/v2/<appid>/<bucket\_name>/<dir\_name>/ HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: application/json

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填“create”
biz_attr	否	String	自定义属性

#### 返回

##### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容

参数名称	必选	类型	描述
data	否	数据集合	服务端返回的应答数据

data 数据集合

参数名称	必选	类型	描述
ctime	是	String	创建时间，10 位 Unix 时间戳

## 示例

请求

POST /files/v2/10055004/accesslog/testfolder/ HTTP/1.1

Host: sh.file.myqcloud.com

Authorization: 9s9dacKhIAAMwBOFpGFpPifhc91hPTEwMDU1MDA0Jms9QUtJRHpUOHd3S3VYanhpeF  
 FBa1JcQzJEUIhCdFBkN0NybEpRjMU9MTQ3MjYzOTI1MSZ0PTE0NzI2MzkwNzEmcj0xMDcyMTgzMDA  
 zJmY9JmI9YWNjZXNzbG9n

Content-Type: application/json

Content-Length: 29

```
{
  "op": "create",
  "biz_attr": ""
}
```

返回

HTTP/1.1 200 OK

Content-Length: 58

Date: Wed Aug 31 18:24:31 2016

Server: tencent-cos

```
{
```

```
"code": 0,  
"message": "SUCCESS",  
"data": {  
  "ctime": 1472639071  
}  
}
```



## 列出目录

### 描述

使用 API 列出指定目录下的所有文件目录，可带前缀查询，只支持向后翻页。

前置条件：目录已存在。

说明：在返回内容中不包含 biz\_attr 字段，如需查询，请调用查询目录属性接口。

### 请求

#### 请求语法

GET <Request\_URL>?op=list&num=<Int64> HTTP/1.1

Host: <Region>.file.myqcloud.com

Authorization: <multi\_effect\_signature>

请求地址 Request\_URL 示例

直接查询 bucket 目录下所有文件和目录：

```
sh.file.myqcloud.com/files/v2/<appid>/<bucket_name>/
```

查询指定 bucket 中文件夹 dir\_name 下的文件和目录：

```
sh.file.myqcloud.com/files/v2/<appid>/<bucket_name>/<dir_name>/
```

带前缀的查询：

```
sh.file.myqcloud.com/files/v2/<appid>/<bucket_name>/<dir_name>/[prefix]
```

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

对于Request\_URL，带前缀查询时，[prefix] 末尾无 ‘/’ 。

## 请求参数

参数名称	必选	类型	描述
op	是	String	操作类型，填 “ list”
num	是	Int	本次查询所要列出的目录总数,最大支持1000
context	否	String	透传字段，从响应的返回内容中得到。若查看第一页，则将空字符串作为 context 传入。若需要翻页，需要将前一页返回内容中的 context 透传到参数中。

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

其中的 data 数据集合：

参数名称	必选	类型	描述
context	是	String	透传字段，用于翻页，业务端不需理解，需要往后翻页则透传给腾讯云
listover	是	Bool	是否有内容可以继续往后翻

参数名称	必选	类型	描述
			页
infos	是	数据集合	文件和文件夹列表，若当前目录下不存在文件或文件夹，则该返回值可能为空

其中的 infos 数据集合：

参数名称	必选	类型	描述
name	是	String	文件名
filesize	否	Int	文件大小，当类型为文件时返回
filelen	否	Int	文件已传输大小，当类型为文件时返回
sha	否	String	文件 sha，当类型为文件时返回
ctime	是	String	创建时间，10位 Unix 时间戳
mtime	是	String	修改时间，10位 Unix 时间戳
access_url	否	String	生成的资源可访问的cdn url，当类型为文件时返回
authority	否	String	如果没有对文件单独设置该属性，则可能不会返回该字段。返回值：eInvalid（表示继承 bucket 的读写权限）；eWRPrivate（私有读写）；eWPrivateRPublic（公有读私有写）。说明：文件可以和 bucket 拥有不同的权限类型，已经设置过权限的文件如果想要撤销，将会直接被赋值为 eInvalid，即继承 bucket

参数名称	必选	类型	描述
			的权限
source_url	否	String	生成的资源可访问的源站 url，当类型为文件时返回

## 示例

### 请求

```
GET /files/v2/10055004/accesslog/aaa/bbb/?op=list&order=0&num=5 HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: F/mImHxxGII5A7ILYnGa8aBtK6thPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhpeFF  
Ba1JCQzJEUihCdFBkN0NybEpRJmU9MTQ3MjY0MDIwNiZ0PTE0NzI2NDAwMjYmcj03OTUxNzkyMCZ  
mPSZiPWFjY2Vzc2xvZw==
```

### 返回

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 2502
```

```
Date: Wed Aug 31 18:40:26 2016
```

```
Server: tencent-cos
```

```
{  
  "code": 0,  
  "message": "SUCCESS",  
  "data": {  
    "context": "/accesslog/aaa/bbb/14709104920.89088300.txt",  
    "infos": [  
      {  
        "access_url": "http://accesslog-10055004.file.myqcloud.com/aaa/bbb/14709104920.63503400.txt",  
        "authority": "eInvalid",  
        "ctime": 1470910492,  
        "filelen": 10264,  
      }  
    ]  
  }  
}
```

```
"filesize": 10264,
"mtime": 1470910492,
"name": "14709104920.63503400.txt",
"preview_url":
"http://accesslog-10055004.preview.myqcloud.com/aaa/bbb/14709104920.63503400.txt?cmd=txt_pr
eview",
"sha": "cad7d66231fa548b2e5b5f431fcf8f1bf028d0b0",
"source_url": "http://accesslog-10055004.cosh.myqcloud.com/aaa/bbb/14709104920.63503400.txt"
},
{
"access_url": "http://accesslog-10055004.file.myqcloud.com/aaa/bbb/14709104920.71280700.txt",
"authority": "eInvalid",
"ctime": 1470910492,
"filelen": 10264,
"filesize": 10264,
"mtime": 1470910492,
"name": "14709104920.71280700.txt",
"preview_url":
"http://accesslog-10055004.preview.myqcloud.com/aaa/bbb/14709104920.71280700.txt?cmd=txt_pr
eview",
"sha": "4ba80f5c809954e93b97de3a3bb0950f6374bdf7",
"source_url": "http://accesslog-10055004.cosh.myqcloud.com/aaa/bbb/14709104920.71280700.txt"
},
{
"access_url": "http://accesslog-10055004.file.myqcloud.com/aaa/bbb/14709104920.77835100.txt",
"authority": "eInvalid",
"ctime": 1470910492,
"filelen": 10264,
"filesize": 10264,
"mtime": 1470910492,
"name": "14709104920.77835100.txt",
"preview_url":
"http://accesslog-10055004.preview.myqcloud.com/aaa/bbb/14709104920.77835100.txt?cmd=txt_pr
eview",
```

```
"sha": "2130bd5f70e0da1a50e9ad70907f5d694ad88bb4",
"source_url": "http://accesslog-10055004.cossh.myqcloud.com/aaa/bbb/14709104920.77835100.txt"
},
{
"access_url": "http://accesslog-10055004.file.myqcloud.com/aaa/bbb/14709104920.83866400.txt",
"authority": "eInvalid",
"ctime": 1470910492,
"filelen": 10264,
"filesize": 10264,
"mtime": 1470910492,
"name": "14709104920.83866400.txt",
"preview_url":
"http://accesslog-10055004.preview.myqcloud.com/aaa/bbb/14709104920.83866400.txt?cmd=txt_pr
eview",
"sha": "1f47ec790b23c950869449309797737e39f1cb92",
"source_url": "http://accesslog-10055004.cossh.myqcloud.com/aaa/bbb/14709104920.83866400.txt"
},
{
"access_url": "http://accesslog-10055004.file.myqcloud.com/aaa/bbb/14709104920.89088300.txt",
"authority": "eInvalid",
"ctime": 1470910492,
"filelen": 10264,
"filesize": 10264,
"mtime": 1470910493,
"name": "14709104920.89088300.txt",
"preview_url":
"http://accesslog-10055004.preview.myqcloud.com/aaa/bbb/14709104920.89088300.txt?cmd=txt_pr
eview",
"sha": "afe17bb32c0835e8b9ce69450f95e3e6f2af4606",
"source_url": "http://accesslog-10055004.cossh.myqcloud.com/aaa/bbb/14709104920.89088300.txt"
}
],
"listover": false
}
```

```
}
```

## 查询目录属性

### 描述

使用 API 查询目录的属性信息，包括：目录名、目录属性、创建时间、修改时间。

前置条件：该目录已存在。

### 请求

#### 请求语法

GET /files/v2/<appid>/<bucket\_name>/<dir\_name>/?op=stat HTTP/1.1

Host: <Region>.file.myqcloud.com

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

### 返回

#### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

#### data 数据集合

参数名称	必选	类型	描述
biz_attr	是	String	目录属性，业务端维护
ctime	是	String	创建时间，10 位 Unix 时间戳



参数名称	必选	类型	描述
mtime	是	String	修改时间，10 位 Unix 时间戳

## 示例

### 请求

```
GET /files/v2/10055004/accesslog/testfolder/?op=stat HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: Lb6JYSxu9GXfRGQybWu2ofs1kOlhPTEwMDU1MDA0Jms9QUtJRHpUOHd3S3VYanhpeF  
FBa1JCQzJEUlhCdFBkN0NybEpRjMmU9MTQ3MjYzOTI1MSZ0PTE0NzI2MzkwNzEmcj0xMzc2NzQ0MzA  
mZj0mYj1hY2Nlc3Nsb2c=
```

### 返回

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 91
```

```
Date: Wed Aug 31 18:24:31 2016
```

```
Server: tencent-cos
```

```
{  
  "code": 0,  
  "message": "SUCCESS",  
  "data": {  
    "biz_attr": "",  
    "ctime": 1472639071,  
    "mtime": 1472639071  
  }  
}
```

## 删除目录

### 描述

使用 API 删除指定目录。

前置条件：必须已清空该目录下的所有文件或文件夹。

限制条件：目标目录不可填写『/』，无法删除根目录，即无法删除Bucket。

### 请求

#### 请求语法

POST /files/v2/<appid>/<bucket\_name>/<dir\_name>/ HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: application/json

Authorization: <once\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填 " delete"

### 返回

#### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容

## 示例

### 请求

```
POST /files/v2/10055004/accesslog/testfolder/ HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: JYv1bYoo9SdiJSBMxZYveGsB001hPTEwMDU1MDA0Jms9QUtJRHpUOHd3S3VYanhpeF
```

```
FBa1JCQzJEUihCdFBkN0NybEpRjMU9MCZ0PTE0NzI2MzkwNzEmcj03ODgwMTAxNTQmZj0vMTAwNT
```

```
UwMDQvYWNjZXNzbG9nL3Rlc3Rmb2xkZXIvJmI9YWNjZXNzbG9n
```

```
Content-Type: application/json
```

```
Content-Length: 15
```

```
{  
  "op": "delete"  
}
```

### 返回

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 30
```

```
Date: Wed Aug 31 18:24:31 2016
```

```
Server: tencent-cos
```

```
{  
  "code": 0,  
  "message": "SUCCESS"  
}
```

## 文件接口

### 简单上传文件

#### 描述

使用 API 上传  $\leq 20\text{MB}$  的文件（超过 20M 的文件，请使用“分片上传文件” API）。

请求地址中的参数即指明，将名称为 file\_name 的文件存储于路径 bucket\_name/[dir\_name] 下，其中，bucket\_name 必填，dir\_name 可缺省。

前置条件：指定的文件目录已存在。

#### 请求

##### 请求语法

POST /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: multipart/form-data

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

##### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填 “upload”
filecontent	是	Binary	文件内容
sha	否	String	文件的 SHA-1 校验码
biz_attr	否	String	文件属性，业务端维护
insertOnly	否	Int	同名文件覆盖选项，有效值： 0 覆盖（删除已有的重名文

参数名称	必选	类型	描述
			件，存储新上传的文件) 1 不覆盖（若已存在重名文件，则不做覆盖，返回“上传失败”）。 默认为 1 不覆盖。

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

data 数据集合：

参数名称	必选	类型	描述
access_url	是	String	通过 CDN 访问该文件的资源链接（访问速度更快）
resource_path	是	String	该文件在 COS 中的相对路径名，可作为其 ID 标识。格式 /appid/bucket/filename。推荐业务端存储 resource_path，然后根据业务需求灵活拼接资源 url（通过 CDN 访问 COS 资源的 url 和直接访问 COS 资源的 url 不同）。
source_url	是	String	（不通过 CDN）直接访问 COS 的资源链接
url	是	String	操作文件的 url

参数名称	必选	类型	描述
			。业务端可以将该 url 作为请求地址来进一步操作文件，对应 API：文件属性、更新文件、删除文件、移动文件中的请求地址。

说明：腾讯云 COS 会默认为每个资源生成经 CDN 的访问链接 access\_url，当业务端尚未开通 CDN 时，仍然可以获得该链接，但是无法访问。

## 示例

请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: VKQg5H9I/QMj/tzkk2M3JOlJFr9hPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhpeFF
Ba1JCQzJEUlhCdFBkN0NybEpRJmU9MTQ3MjY0NjAzOCZ0PTE0NzI2NDU4NTgmcj0xMjUyMDk1MjM5
JmY9JmI9YWNjZXNzbG9n
```

```
Content-Length: 577
```

```
Content-Type: multipart/form-data; boundary=-----aa502a40917c
```

```
-----aa502a40917c
```

```
Content-Disposition: form-data; name="op"
```

```
upload
```

```
-----aa502a40917c
```

```
Content-Disposition: form-data; name="sha"
```

```
1fe86826617c3a32da4f06bcc71a2a718b274c6
```

```
-----aa502a40917c
```

```
Content-Disposition: form-data; name="biz_attr"
```

```
-----aa502a40917c
```

Content-Disposition: form-data; name="filecontent"; filename="/data/liudg/cos-php-sdk/111.txt"

Content-Type: application/octet-stream

hello, I am 111.txt.

-----aa502a40917c--

返回

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Content-Length: 434

Date: Wed Aug 31 20:17:38 2016

Server: tencent-cos

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "access_url": "http://accesslog-10055004.file.myqcloud.com/testfolder/111.txt",
    "preview_url":
"http://accesslog-10055004.preview.myqcloud.com/testfolder/111.txt?cmd=txt_preview",
    "resource_path": "/10055004/accesslog/testfolder/111.txt",
    "source_url": "http://accesslog-10055004.cosh.myqcloud.com/testfolder/111.txt",
    "url": "http://sh.file.myqcloud.com/files/v2/10055004/accesslog/testfolder/111.txt"
  }
}
```

## 初始化分片上传

### 描述

使用 API 对大于 20 MB 的文件进行分片上传之前，需要先使用本 API 创建「分片上传会话」，即得到此次上传的 session。

前置条件：指定目录已存在。

### 分片上传使用流程说明

1. 发送「初始化分片上传」请求。
2. 腾讯云会判断文件的上传状态，若未完成上传，会返回-4019错误，用户需调用查询上传分片接口查询已经上传完的分片，并进行断点续传的操作；若文件从未传输过，则返回用户上传的session。
3. 设置 session 和 offset 参数，构造「逐个分片上传」请求，逐片上传后续文件。
4. 循环执行第 3 步直到文件的分片数据上传完成。
5. 调用finish接口结束分片上传。

### 请求

#### 请求语法

POST /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: multipart/form-data

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填"upload_slice_init"



参数名称	必选	类型	描述
filesize	是	Int 64	文件总大小，单位为 Byte
slice_size	是	Int 64	分片大小，单位为 Byte。□ 有效值：524288 (512 KB), 1048576 (1 MB), 2097152 (2 MB), 3145728 (3 MB)
biz_attr	否	String	文件属性，业务端维护
insertOnly	否	Int	同名文件是否覆盖。 有效值： 0 覆盖（删除已有的重名文件，存储新上传的文件） 1 不覆盖（若已存在重名文件，则不做覆盖，返回上传失败）。 不填此参数，默认不覆盖。

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

data 数据集合：

参数名称	必选	类型	描述
session	否	String	唯一标识此文件传输过程的 id，命中秒传则不携带
slice_size	否	Int	分片大小，单位为 Byte。□ 有效值：524288 (512 KB), 1048576 (1 MB),

参数名称	必选	类型	描述
			2097152 (2 MB), 3145728 (3 MB)

## 示例

### 初始化请求

此示例呈现了一个正常的初始化分片上传的请求和返回，对象存储系统中没有相同的文件，即没有命中秒传。

#### 请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1
Host: sh.file.myqcloud.com
Authorization: JAVEi4cbYHm5cqGpUp96HKP+JXNhPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhp
eFFBa1JCQzJEUIhCdFBkN0NybEpRjMUmU9MTQ3MjY0Nzc4MCZ0PTE0NzI2NDc2MDAmcj00NjQ3OTQxO
DkmZj0mYj1hY2Nlc3Nsb2c=
Method:POST
Content-Length: 361
Content-Type: multipart/form-data; boundary=-----f43b7cf5bf1b

-----f43b7cf5bf1b
Content-Disposition: form-data; name="op"

upload_slice_init
-----f43b7cf5bf1b
Content-Disposition: form-data; name="filesize"

36864
-----f43b7cf5bf1b
Content-Disposition: form-data; name="slice_size"

524288
-----f43b7cf5bf1b--
```

返回

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Content-Length: 168

Date: Wed Aug 31 20:46:40 2016

Server: tencent-cos

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "session":
"58ccd1b3120e85c4ebb7b786154408df4fe22230cb8f328d41599b237661855d6242793ba2481c1ac0b
1277051003cab",
    "slice_size": 524288
  }
}
```

至此，请求者获取到 session可继续进行「逐个分片上传」。

## 逐个上传分片

### 描述

使用 API 对大于 20 MB 的文件进行逐片上传之前，需要先使用「初始化分片上传」API，从腾讯云获取到分片上传所需的参数（session 和 slice\_size）。

前置条件：指定目录已存在，且已获得分片上传所需参数（session 和 slice\_size）。

### 分片上传使用流程说明

1. 发送「初始化分片上传」请求。
2. 腾讯云会判断文件的上传状态，若未完成上传，会返回-4019错误，用户需调用查询上传分片接口查询已经上传完的分片，并进行断点续传的操作；若文件从未传输过，则返回用户上传的session。
3. 设置 session 和 offset 参数，构造「逐个分片上传」请求，逐片上传后续文件。
4. 循环执行第 3 步直到文件的分片数据上传完成。
5. 调用finish接口结束分片上传。

## 请求

### 请求语法

POST /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: multipart/form-data

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填 "upload_slice_data"

参数名称	必选	类型	描述
filecontent	是	Binary	文件内容
session	是	String	唯一标识此文件传输过程的id, 由后台下发, 调用方透传
offset	是	Int 64	本次分片位移

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

data 数据集合：

参数名称	必选	类型	描述
session	否	String	唯一标识此文件传输过程的id
offset	否	Int 64	请求包体里的传输的位移, 调用方如果用多线程等方式传输, 可以用来唯一确定本次分片结果

## 示例

### 逐个分片上传

请求

POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1

Host: sh.file.myqcloud.com

Authorization: uusZIBVdpHf800YuKImvK5zvw75hPTEwMDU1MDA0Jms9QUtjRHpuOHd3S3VYanhpeF

FBa1JCQzJEUIhCdFBkN0NybEpRjMjU9MTQ3MjY0Nzc4MCZ0PTE0NzI2NDc2MDAmcj0xMDYwMzQ1OT  
M4JmY9JmI9YWNjZXNzbG9n

Content-Length: 37406

Content-Type: multipart/form-data; boundary=-----adb78d5a54eb

-----adb78d5a54eb

Content-Disposition: form-data; name="op"

upload\_slice\_data

-----adb78d5a54eb

Content-Disposition: form-data; name="session"

58ccd1b3120e85c4ebb7b786154408df4fe22230cb8f328d41599b237661855d6242793ba2481c1ac0b1  
277051003cab

-----adb78d5a54eb

Content-Disposition: form-data; name="offset"

0

-----adb78d5a54eb

Content-Disposition: form-data; name="filecontent"

.....G.E2...Q..I..E.i.U....w.o.v.....U2

(此处忽略文件内容)

-----adb78d5a54eb--

返回

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Content-Length: 175

Date: Wed Aug 31 20:46:40 2016

Server: tencent-cos

{

```
"code": 0,  
"message": "SUCCESS",  
"data": {  
  "datalen": 36864,  
  "offset": 0,  
  "session":  
"58ccd1b3120e85c4ebb7b786154408df4fe22230cb8f328d41599b237661855d6242793ba2481c1ac0b  
1277051003cab"  
}  
}
```

## 结束上传分片

### 描述

使用 API 对大于 20 MB 的文件进行逐片上传之前，需要先使用「初始化分片上传」API，从腾讯云获取到分片上传所需的参数（session 和 slice\_size）。

前置条件：指定目录已存在，且已获得分片上传所需参数（session 和 slice\_size）。

当分片数据上传完成后，需调用此接口结束文件上传流程，腾讯云会返回文件的信息。

### 分片上传使用流程说明

1. 发送「初始化分片上传」请求。
2. 腾讯云会判断文件的上传状态，若未完成上传，会返回-4019错误，用户需调用查询上传分片接口查询已经上传完的分片，并进行断点续传的操作；若文件从未传输过，则返回用户上传的session。
3. 设置 session 和 offset 参数，构造「逐个分片上传」请求，逐片上传后续文件。
4. 循环执行第 3 步直到文件的分片数据上传完成。
5. 调用finish接口结束分片上传。

## 请求

### 请求语法

```
POST /files/v2/<appid>/<bucket_name>[/dir_name]/<file_name> HTTP/1.1
```

```
Host: <Region>.file.myqcloud.com
```

```
Content-Type: multipart/form-data
```

```
Authorization: <multi_effect_signature>
```

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

### 请求内容

参数名称	必选	类型	描述
------	----	----	----



参数名称	必选	类型	描述
op	是	String	操作类型，填 “upload_slice_finish”
session	是	String	唯一标识此文件传输过程的 id，由后台下发， 调用方透传
filesize	是	Int 64	文件大小

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

data 数据集合：

参数名称	必选	类型	描述
access_url	否	String	通过 CDN 访问该文件的资源链接（访问速度更快）
resource_path	否	String	该文件在COS中的相对路径名，可作为其 ID 标识。 格式 /appid/bucket/filename。推荐业务端存储 resource_path，然后根据业务需求灵活拼接资源 url（通过 CDN 访问 COS 资源的 url 和直接访问 COS 资源的 url 不同）。
source_url	否	String	（不通过 CDN）直接访问 COS 的资源链接
url	否	String	操作文件的

参数名称	必选	类型	描述
			url。业务端可以将该 url 作为请求地址来进一步操作文件，对应 API：查询文件属性、更新文件、删除文件、移动文件中的请求地址。

说明：腾讯云对象存储会默认为每个资源生成经 CDN 的访问链接 access\_url，当业务端尚未开通 CDN 时，仍然可以获得该链接，但是无法访问。

## 示例

### 逐个分片上传

请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: uusZIBVdpHf800YuKImvK5zvw75hPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhpeF
FBa1JCQzJEUIhCdFBkN0NybEpRJmU9MTQ3MjY0Nzc4MCZ0PTE0NzI2NDc2MDAmcj0xMDYwMzQ1OT
M4JmY9JmI9YWNjZXNzbG9n
```

```
Content-Length: 450
```

```
Content-Type: multipart/form-data; boundary=-----2335c30f02ab
```

```
-----2335c30f02ab
```

```
Content-Disposition: form-data; name="op"
```

```
upload_slice_finish
```

```
-----2335c30f02ab
```

```
Content-Disposition: form-data; name="session"
```

```
58ccd1b3120e85c4ebb7b786154408df4fe22230cb8f328d41599b237661855d6242793ba2481c1ac0b1
277051003cab
```

```
-----2335c30f02ab
```

```
Content-Disposition: form-data; name="filesize"
```

36864

-----2335c30f02ab--

返回

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Content-Length: 434

Date: Wed Aug 31 20:46:41 2016

Server: tencent-cos

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "access_url": "http://accesslog-10055004.file.myqcloud.com/testfolder/111.txt",
    "preview_url":
"http://accesslog-10055004.preview.myqcloud.com/testfolder/111.txt?cmd=txt_preview",
    "resource_path": "/10055004/accesslog/testfolder/111.txt",
    "source_url": "http://accesslog-10055004.cossh.myqcloud.com/testfolder/111.txt",
    "url": "http://sh.file.myqcloud.com/files/v2/10055004/accesslog/testfolder/111.txt"
  }
}
```

## 移动文件

### 描述

使用 API 进行文件的移动或重命名操作。“请求地址”中的参数为源文件的位置，“请求内容”中的参数 `dest_fileid` 指明移动至的目标路径（以及新的文件名）。

说明：当移动文件 A 至目标目录时，若目标目录下存在与 A 同名的文件，则请求中的 `to_over_write` 将决定是否覆盖该同名文件。若选择覆盖，则移动操作成功，若选择不覆盖，则移动操作失败。

前置条件：该文件已存在，且目标目录也已存在。

### 请求

#### 请求语法

POST /files/v2/<appid>/<bucket\_name>/[dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: multipart/form-data

Authorization: <once\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填“move”
dest_fileid	是	String	目标路径（不带路径则为当前路径下，带路径则会移动到携带指定的路径下）
to_over_write	否	Int	覆盖写入目标文件选项，有效值： 0 不覆盖（若已存在同名文件，则不做覆盖，返回“移

参数名称	必选	类型	描述
			动失败” ) 1 覆盖 默认值为 0 不覆盖。

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容

## 示例

### 请求

POST /files/v2/251668577/rabbitliutest4x/rabbit.txt HTTP/1.1

Authorization:ksYXsEbCG1G9ELdP15bp3LH9WQthPTEyNTE2Njg1Nzcmaz1BS0IEV3RUQ0JZak01T3dM  
QjIDQXdBMVFiMIRoVFNVamZHRk8mZT0xNDc5Mzg4Njg3JnQ9MTQ3Njc5NjY4NyZyPTk1OTM4ODQ  
wNSZmPSZiPXJhYmJpdGxpdXRlc3Q0eA==

Host: gz.file.myqcloud.com

Content-Length: 255

Content-Type: multipart/form-data; boundary=-----e492f6322ce5

HTTP/1.1 100 Continue

-----e492f6322ce5

Content-Disposition: form-data; name="op"

move

-----e492f6322ce5

Content-Disposition: form-data; name="dest\_fileid"

rabbit.txt.bak

-----e492f6322ce5--

返回

HTTP/1.1 200 OK

Server: tencent-cos

Date: Thu, 15 Jun 2016 06:46:48 GMT

Content-Type: \*/\*

Content-Length: 41

```
{  
  "code": 0,  
  "message": "SUCCESS"  
}
```

## 查询文件属性

### 描述

使用 API 查询文件的属性信息，包括文件大小、文件 SHA-1 校验码、文件创建时间、修改时间、访问权限、自定义头部信息等。

前置条件：该文件已存在。

### 请求

#### 请求语法

GET /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name>?op=stat HTTP/1.1

Host: <Region>.file.myqcloud.com

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

### 返回

#### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	否	数据集合	服务器返回的应答数据

data 数据集合：

参数名称	必选	类型	描述
name	是	String	文件名
biz_attr	是	Int	文件属性，业务端维护

参数名称	必选	类型	描述
filesize	是	Int	文件大小
sha	是	String	文件 SHA-1 校验码
ctime	是	String	创建时间，10 位 Unix 时间戳
mtime	是	String	修改时间，10 位 Unix 时间戳
access_url	是	String	通过 CDN 访问该文件的资源链接
source_url	是	String	( 不通过 CDN ) 直接访问 COS 的资源链接
authority	否	String	Object 的权限，默认与 Bucket 权限一致，此时不会返回该字段。 如果设置了独立权限，则会返回该字段。 有效值： eInvalid 空权限，此时系统会默认调取 Bucket 权限 eWRPrivate 私有读写 eWPrivateRPublic 公有读私有写
custom_headers	否	数据集合	用户自定义头部

custom\_headers 数据集合：

参数名称	必选	类型	描述
Cache-Control	否	String	文件的缓存机制
Content-Type	否	String	文件的 MIME 信息
Content-Disposition	否	String	MIME 协议的扩展
Content-Language	否	String	文件的语言
Content-Encoding	否	String	文件的加密格式
x-cos-meta-自定义内容	否	String	自定义内容



## 示例

### 请求

```
GET /files/v2/10055004/accesslog/testfolder/111.txt?op=stat HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: QVZWK/lxpiMoz+ArZH9Hny7IvDZhPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhpe
```

```
FFBa1JCQzJEUIhCdFBkN0NybEpRJmU9MTQ3MjY0NjAzOCZ0PTE0NzI2NDU4NTgmcj0yODU2NDU2N
```

```
DcmZj0mYj1hY2Nlc3Nsb2c=
```

### 返回

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 576
```

```
Date: Wed Aug 31 20:17:38 2016
```

```
Server: tencent-cos
```

```
{  
  "code": 0,  
  "message": "SUCCESS",  
  "data": {  
    "access_url": "http://accesslog-10055004.file.myqcloud.com/testfolder/111.txt",  
    "authority": "eWPrivateRPublic",  
    "biz_attr": "",  
    "ctime": 1472645858,  
    "custom_headers": {  
      "Cache-Control": "no",  
      "Content-Language": "ch",  
      "Content-Type": "application/pdf"  
    },  
    "filelen": 21,  
    "filesize": 21,  
    "forbid": 0,
```

```
"mtime": 1472645858,  
"preview_url":  
"http://accesslog-10055004.preview.myqcloud.com/testfolder/111.txt?cmd=txt_preview",  
"sha": "1fe86826617c3a32da4f06bccc71a2a718b274c6",  
"slicesize": 21,  
"source_url": "http://accesslog-10055004.cossh.myqcloud.com/testfolder/111.txt"  
}  
}
```

## 查询上传分片

### 描述

使用 API 对大于 20 MB 的文件进行逐片上传之前，需要先使用「初始化分片上传」API，从腾讯云获取到分片上传所需的参数（session 和 slice\_size）。

前置条件：指定目录已存在，且已获得分片上传所需参数（session 和 slice\_size）。

当初始化分片上传返回-4019错误的时候，用户需调用此接口查询分片的断点和session，进行续传操作。

### 分片上传使用流程说明

1. 发送「初始化分片上传」请求。
2. 腾讯云会判断文件的上传状态，若未完成上传，会返回-4019错误，用户需调用查询上传分片接口查询已经上传完的分片，并进行断点续传的操作；若文件从未传输过，则返回用户上传的session。
3. 设置 session 和 offset 参数，构造「逐个分片上传」请求，逐片上传后续文件。
4. 循环执行第 3 步直到文件的分片数据上传完成。
5. 调用finish接口结束分片上传。

## 请求

### 请求语法

POST /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: multipart/form-data

Authorization: <multi\_effect\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

### 请求内容

参数名称	必选	类型	描述
------	----	----	----

参数名称	必选	类型	描述
op	是	String	操作类型，填 "upload_slice_list"

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容
data	是	数据集合	服务器返回的应答数据

data 数据集合：

参数名称	必选	类型	描述
session	否	String	唯一标识此文件传输过程的id，命中秒传则不携带
slice_size	否	Int	分片大小，单位为 Byte。□ 有效值：524288 (512 KB), 1048576 (1 MB), 2097152 (2 MB), 3145728 (3 MB)
listparts	否	String	完成上传的文件的分片信息
access_url	否	String	文件已上传完成返回，通过CDN 访问该文件的资源链接（访问速度更快）
resource_path	否	String	文件已上传完成返回，该文件在COS中的相对路径名，可作为其 ID 标识。格式 /appid/bucket/filename。 推荐业务端存储 resource_path，然后根据业务需求灵活拼接资源

参数名称	必选	类型	描述
			url ( 通过 CDN 访问 COS 资源的 url 和直接访问 COS 资源的 url 不同 )。
source_url	否	String	文件已上传完成返回, ( 不通过 CDN ) 直接访问 COS 的资源链接
url	否	String	文件已上传完成返回, 操作文件的 url。业务端可以将该 url 作为请求地址来进一步操作文件, 对应 API: 查询文件属性、更新文件、删除文件、移动文件中的请求地址。

说明：腾讯云对象存储会默认为每个资源生成经 CDN 的访问链接 access\_url，当业务端尚未开通 CDN 时，仍然可以获得该链接，但是无法访问。

listparts 数据集合：

参数名称	必选	类型	描述
datalen	是	Int	分片大小
offset	是	Int64	分片偏移

## 示例

### 逐个分片上传

请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: E+w/b7r3hVCOX9Dkc0sD0gU6ycFhPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhpeFFBa1JCQzJEUIhCdFBkN0NybEpRJmU9MTQ3MjY0OTYwOCZ0PTE0NzI2NDk0Mjgmcj01NTgzMDA2MzgmZj0mYj1hY2Nlc3Nsb2c=
```

Content-Length: 154

Content-Type: multipart/form-data; boundary=-----2b804029035c

-----2b804029035c

Content-Disposition: form-data; name="op"

upload\_slice\_list

-----2b804029035c--

返回

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Content-Length: 304

Date: Wed Aug 31 21:19:40 2016

Server: tencent-cos

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "filesize": 104857600,
    "listparts": [
      {
        "datalen": 524288,
        "offset": 0
      },
      {
        "datalen": 524288,
        "offset": 524288
      },
      {
        "datalen": 524288,
        "offset": 1048576
      }
    ]
  }
}
```

```
}  
],  
"session":  
"4331155f7e26e5fc7f032e7d1dac56644b68681d6c4260e39db441604773c883ca2b5ed5f44ab5bd44f8  
1f9ee383165f",  
"slice_size": 524288  
}  
}
```

文件已经上传完成的情况

请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1  
Host: sh.file.myqcloud.com  
Authorization: 5/6eYiDGOxpd/YRDTscWG4Y265hhPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhpe  
FFBa1JCQzJEUIhCdFBkN0NybEpRJmU9MTQ3MjY0OTQ4OCZ0PTE0NzI2NDkzMDgmcj0xNDA1ODgwN  
Tk0JmY9JmI9YWNjZXNzbG9n  
Content-Length: 154  
Content-Type: multipart/form-data; boundary=-----72e1396a1d85  
  
-----72e1396a1d85  
Content-Disposition: form-data; name="op"  
  
upload_slice_list  
-----72e1396a1d85--
```

返回

```
HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8  
Content-Length: 434  
Date: Wed Aug 31 21:15:08 2016  
Server: tencent-cos
```

```
{
  "code": 0,
  "message": "SUCCESS",
  "data": {
    "access_url": "http://accesslog-10055004.file.myqcloud.com/testfolder/111.txt",
    "preview_url":
"http://accesslog-10055004.preview.myqcloud.com/testfolder/111.txt?cmd=txt_preview",
    "resource_path": "/10055004/accesslog/testfolder/111.txt",
    "source_url": "http://accesslog-10055004.cosh.myqcloud.com/testfolder/111.txt",
    "url": "http://sh.file.myqcloud.com/files/v2/10055004/accesslog/testfolder/111.txt"
  }
}
```



## 更新文件属性

### 描述

使用 API 对腾讯云对象存储中某个文件的属性进行设置或修改。

前置条件：该文件已存在。

说明：目前只提供了对 authority 和 custom\_headers 的更新操作，在更新时需要用 flag 来决定此次操作是更新 authority 还是 custom\_headers 或者二者都更新。

对于自定义头部 custom\_headers 中的各属性值，均由用户来定义和维护，腾讯云对象存储不负责维护其属性值是否正确有效。

### 请求

#### 请求语法

POST /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: application/json

Authorization: <once\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填“ update”
authority	否	String	Object 的权限，默认与 Bucket 权限一致，此时不会返回该字段。 如果设置了独立权限，则会返回该字段。

参数名称	必选	类型	描述
			有效值： eInvalid 空权限，此时系统会默认调取 Bucket 权限 eWRPrivate 私有读写 eWPrivateRPublic 公有读私有写
custom_headers	否	数据集合	用户自定义 header，在执行更新操作时，只需填写需要增加或修改的项

其中，custom\_headers 数据集合可以包含以下参数：

参数名称	必选	类型	描述
Cache-Control	否	String	文件的缓存机制
Content-Type	否	String	文件的 MIME 信息
Content-Disposition	否	String	MIME 协议的扩展
Content-Language	否	String	文件的语言
Content-Encoding	否	String	文件的编码格式
x-cos-meta-自定义内容	否	String	自定义内容

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容

## 示例

### 请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

Authorization: 63pxuy84NVa8KA2PsMvb4KNpUpthPTEwMDU1MDA0Jms9QUtJRHpuOHd3S3VYanhp  
eFFBa1JCQzJEUIhCdFBkN0NybEpRJmU9MCZ0PTE0NzI2NDU4NTgmcj0xNTQ5NjU4MzYyJmY9LzEwM  
DU1MDA0L2FjY2Vzc2xvZy90ZXN0Zm9sZGVyLzExMS50eHQmYj1hY2Nlc3Nsbn2c=

Content-Type: application/json

Content-Length: 158

```
{
  "op": "update",
  "biz_attr": "",
  "authority": "eWPrivateRPublic",
  "custom_headers": {
    "Cache-Control": "no",
    "Content-Type": "application/pdf",
    "Content-Language": "ch"
  }
}
```

注：0x40 | 0x80，flag 取值表示同时对 authority 和 custom\_headers 进行更新操作

返回

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Content-Length: 30

Date: Wed Aug 31 20:17:38 2016

Server: tencent-cos

```
{
  "code": 0,
  "message": "SUCCESS"
}
```

## 删除文件

### 描述

使用 API 进行文件的删除操作。

前置条件：该文件已存在。

### 请求

#### 请求语法

POST /files/v2/<appid>/<bucket\_name>[/dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: application/json

Authorization: <once\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填“ delete”

### 返回

#### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容

### 示例

## 请求

```
POST /files/v2/10055004/accesslog/testfolder/111.txt HTTP/1.1
```

```
Host: sh.file.myqcloud.com
```

```
Authorization: NvUbNZcimJ20OGI4tTVMX6GwGD5hPTEwMDU1MDA0Jms9QUtJRHpUOHd3S3VYanh  
peFFBa1JCQzJEUlhCdFBkN0NybEpRjMmU9MCZ0PTE0NzI2NDU4NTgmcj0yMzY2NjE2NDkmZj0vMTAw  
NTUwMDQvYWVWNjZlZG9nL3Rlc3Rmb2xkZXIvMTExLnR4dCZiPWFjY2Vzc2xvZW==
```

```
Content-Type: application/json
```

```
Content-Length: 15
```

```
{  
  "op": "delete"  
}
```

## 返回

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
Content-Length: 30
```

```
Date: Wed Aug 31 20:17:38 2016
```

```
Server: tencent-cos
```

```
{  
  "code": 0,  
  "message": "SUCCESS"  
}
```

## 下载文件

### 描述

使用 API 下载对应文件。公有文件无需拼接签名，私有文件需要拼接多次有限签名。

### 请求

#### 请求语法

```
GET /[dir_name]/<file_name> HTTP/1.1
Host: <BucketName>-<AppID>.<Region>.myqcloud.com
Authorization: <multiple_signature>
```

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

无请求内容

### 返回

#### 返回内容

Object文件

### 示例

#### 请求

```
GET /rabbit.txt HTTP/1.1
Authorization:ksYXsEbCG1G9ELdP15bp3LH9WQthPTEyNTE2Njg1Nzcmaz1BS0IEV3RUQ0JZak01T3dM
QjIDQXdBMVFiMIRoVFNVamZHRk8mZT0xNDc5Mzg4Njg3JnQ9MTQ3Njc5NjY4NyZyPTk1OTM4ODQ
```

wNSZmPSZiPXJhYmJpdGxpdXRlc3Q0eA==

Host: rabbitliutest4x-1251668577.cosgz.myqcloud.com

返回

HTTP/1.1 200 OK

Server: tencent-cos

Date: Thu, 15 Jun 2016 06:46:48 GMT

Content-Length: 41

Object File

## 复制文件

### 描述

使用 API 进行文件的复制操作。“请求地址”中的参数为源文件的位置，“请求内容”中的参数 dest\_fileid 指明复制至的目标路径（以及新的文件名）。

说明：当复制文件 A 至目标目录时，若目标目录下存在与 A 同名的文件，则请求中的 to\_over\_write 将决定是否覆盖该同名文件。若选择覆盖，则复制操作成功，若选择不覆盖，则复制操作失败。

前置条件：该文件已存在，且目标目录也已存在。

### 请求

#### 请求语法

POST /files/v2/<appid>/<bucket\_name>/[dir\_name]/<file\_name> HTTP/1.1

Host: <Region>.file.myqcloud.com

Content-Type: multipart/form-data

Authorization: <once\_signature>

说明：<箭头括号>表示 必须 替换为有效值的变量，[英文方括号]表示可选的命令或参数。

#### 请求内容

参数名称	必选	类型	描述
op	是	String	操作类型，填“copy”
dest_fileid	是	String	目标路径（不带路径则为当前路径下，带路径则会复制到携带指定的路径下）
to_over_write	否	Int	覆盖写入目标文件选项，有效值： 0 不覆盖（若已存在重名文件，则不做覆盖，返回“复



参数名称	必选	类型	描述
			制失败” ) 1 覆盖 默认值为 0 不覆盖。

## 返回

### 返回内容

参数名称	必选	类型	描述
code	是	Int	服务端返回码
message	是	String	服务端提示内容

## 示例

### 请求

POST /files/v2/251668577/rabbitliutest4x/rabbit.txt HTTP/1.1

Authorization:ksYXsEbCG1G9ELdP15bp3LH9WQthPTEyNTE2Njg1Nzcmaz1BS0IEV3RUQ0JZak01T3dM  
QjIDQXdBMVFiMIRoVFNVamZHRk8mZT0xNDc5Mzg4Njg3JnQ9MTQ3Njc5NjY4NyZyPTk1OTM4ODQ  
wNSZmPSZiPXJhYmJpdGxpdXRlc3Q0eA==

Host: gz.file.myqcloud.com

Content-Length: 255

Content-Type: multipart/form-data; boundary=-----e492f6322ce5

HTTP/1.1 100 Continue

-----e492f6322ce5

Content-Disposition: form-data; name="op"

copy

-----e492f6322ce5

Content-Disposition: form-data; name="dest\_fileid"

rabbit.txt.bak

-----e492f6322ce5--

返回

HTTP/1.1 200 OK

Server: tencent-cos

Date: Thu, 15 Jun 2016 06:46:48 GMT

Content-Type: \*/\*

Content-Length: 41

```
{  
  "code": 0,  
  "message": "SUCCESS"  
}
```