Cloud Object Storage

API Documentation

Product Documentation
Contents

API Documentation
  Overview
  Common Request Headers
  Common Response Headers
  Error Codes
  Request Signature
  Operation list
Service APIs
  GET Service
Bucket APIs
  Basic Operations
    PUT Bucket
    GET Bucket (List Object)
    HEAD Bucket
    DELETE Bucket
    GET Bucket Object Versions
  Access Control List (acl)
    PUT Bucket acl
    GET Bucket acl
  Cross-Origin Resource Sharing (cors)
    PUT Bucket cors
    GET Bucket cors
    DELETE Bucket cors
  Lifecycle (lifecycle)
    PUT Bucket lifecycle
    GET Bucket lifecycle
    DELETE Bucket lifecycle
  Bucket policy(policy)
    PUT Bucket policy
    GET Bucket policy
    DELETE Bucket policy
  Hotlink protection (referer)
    PUT Bucket referer
    GET Bucket Referer
  Tag (tagging)
    PUT Bucket tagging
    GET Bucket tagging
    DELETE Bucket tagging
  Static website(website)
    PUT Bucket website
    GET Bucket website
    DELETE Bucket website
  Bucket inventory(inventory)
    PUT Bucket inventory
    GET Bucket inventory
    List Bucket Inventory Configurations
    DELETE Bucket inventory
  Versioning
    PUT Bucket versioning
    GET Bucket versioning
  Cross-region Replication (replication)
    PUT Bucket replication
    GET Bucket replication
    DELETE Bucket replication
  Log Management(logging)
PUT Bucket logging
GET Bucket logging
Bucket Encryption (encryption)
PUT Bucket encryption
GET Bucket encryption
DELETE Bucket encryption
Global Acceleration (Accelerate)
PUT Bucket Accelerate
GET Bucket Accelerate

Object APIs
Basic Operations
PUT Object
PUT Object - Copy
POST Object
GET Object
HEAD Object
DELETE Object
DELETE Multiple Objects
OPTIONS Object
POST Object restore
SELECT Object Content
Access Control List (acl)
PUT Object acl
GET Object acl
Multipart Upload
Initiate Multipart Upload
Upload Part
Upload Part - Copy
Complete Multipart Upload
Abort Multipart Upload
List Multipart Uploads
List Parts
Batch Operation APIs
CreateJob
DescribeJob
ListJobs
UpdateJobPriority
UpdateJobStatus
CommonElements
ErrorResponse
COS uses XML APIs, which are lightweight, connectionless, and stateless. By calling XML APIs, you can send requests and accept responses directly through HTTP/HTTPS to interact with the COS backend.

Since COS APIs and TencentCloud APIs use different data transfer frameworks, APIs and SDKs provided by COS are independent of TencentCloud APIs and SDKs. You can find more information in API Operation List or download SDKs in SDK Overview. The guides for TencentCloud APIs and corresponding SDKs do not cover the operational features of COS.

- By using Tencent Cloud COS APIs, you acknowledge that you have read and agree to the Tencent Cloud Terms of Service and Tencent Cloud Object Storage Service Level Agreement.
- For more information on the regions where COS is available, see Regions and Access Domain Names.
- Before using the APIs or SDKs to initiate a request, you are recommended to read the Creating Request Overview document to learn more about access domain name, authentication, and how to determine whether an access is via private or public network.

## Glossary

Below are some main concepts and terms that may come up in the documentation:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPID</td>
<td>A unique resource ID at the user level owned by a developer accessing COS services, which is used to identify resources</td>
</tr>
<tr>
<td>SecretId</td>
<td>Developer-owned project ID for identity verification</td>
</tr>
<tr>
<td>SecretKey</td>
<td>Developer-owned project key</td>
</tr>
<tr>
<td>Bucket</td>
<td>A container used to store data in COS</td>
</tr>
<tr>
<td>Object</td>
<td>A specific file stored in COS, which is the basic storage entity</td>
</tr>
<tr>
<td>Region</td>
<td>Region in domain names such as ap-beijing, ap-hongkong, and eu-frankfurt. For enumerated values, see Regions and Access Domain Names</td>
</tr>
<tr>
<td>ACL</td>
<td>Access Control List, a list of access control information for a specified bucket or object</td>
</tr>
<tr>
<td>CORS</td>
<td>Cross-origin resource sharing (CORS). This refers to HTTP requests where the origin of the resource that initiates the request is different from the origin of the destination resource</td>
</tr>
<tr>
<td>Multipart Uploads</td>
<td>Multipart upload mode provided by Tencent Cloud COS service for uploading files in parts</td>
</tr>
</tbody>
</table>

## Getting Started

Take the following steps to start using COS APIs:

1. Activate the COS service in the COS Console.
2. Create a bucket in the COS Console.
3. Obtain the APPID, SecretId, and SecretKey on the Cloud API Key page in the CAM Console.
4. Write a request signature algorithm program or use a server-side SDK. For more information, see Request Signature.
5. Calculate the signature and call an API to perform an operation.
Common Request Headers

Description

This document describes the common request headers that may be used in API requests. The headers mentioned below will not be detailed again in specific API documents.

### Request Headers

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>Carries authentication information, i.e., the signing information used to verify the validity of a request. This header is optional for public-read objects or if the authentication information is passed in through request parameters. For more information, see Request Signature.</td>
<td>string</td>
<td>Yes.</td>
</tr>
<tr>
<td>Content-Length</td>
<td>HTTP request length in bytes as defined in RFC 2616.</td>
<td>integer</td>
<td>No.</td>
</tr>
<tr>
<td>Content-Type</td>
<td>HTTP request content type (MIME) as defined in RFC 2616. For example, application/xml, image/jpeg</td>
<td>string</td>
<td>No.</td>
</tr>
<tr>
<td>Content-MD5</td>
<td>Base64-encoded MD5 hash value of the request body content as defined in RFC 1864, such as ZzD3iDJdrWAbBB0gLLeigg=. This header is used for integrity check to verify whether the request body has changed during the transfer. For PUT and POST requests with a request body (except POST Object requests), it is highly recommended to carry this header.</td>
<td>string</td>
<td>No.</td>
</tr>
<tr>
<td>Date</td>
<td>Current time in GMT format as defined in RFC 1123, such as Wed, 20 May 2019 04:10:12 GMT.</td>
<td>string</td>
<td>No.</td>
</tr>
<tr>
<td>Host</td>
<td>Requested host in the format of &lt;BucketName-APPID&gt;.cos.&lt;Region&gt;.myqcloud.com</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-security-token</td>
<td>The security token field that needs to be passed in when temporary security credentials are used. For more information, see Temporary Security Credentials.</td>
<td>string</td>
<td>No.</td>
</tr>
</tbody>
</table>

### Server-side Encryption Headers

For APIs that support server-side encryption (SSE), the following request headers are applicable based on different encryption methods. Please consult the specific API documents to determine whether SSE is applicable. Whether the following headers are required is only for scenarios where SSE is used. If you request an API that does not support or use SSE, the following headers do not need to be carried. For more information, see Server-side Encryption Overview.

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE-COS</td>
<td></td>
</tr>
</tbody>
</table>

©2013-2019 Tencent Cloud. All rights reserved.
<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption</td>
<td>Server-side encryption algorithm; currently only AES256 is supported</td>
<td>string</td>
<td>Required when you upload or copy objects (including simple upload/copy and multipart upload/copy). This header cannot be specified when you download objects</td>
</tr>
</tbody>
</table>

**SSE-C**

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption-customer-algorithm</td>
<td>Server-side encryption algorithm; currently only AES256 is supported</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-server-side-encryption-customer-key</td>
<td>Base64-encoded server-side encryption key. For example, MDEyMzQ1Njc4OUFCQ0RFRjAxNjMJMTY3ODI1ODVHERY=</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-server-side-encryption-customer-key-MD5</td>
<td>Base64-encoded MD5 hash of the server-side encryption key. For example, USL6I7jcm0fN7r7xg8g==</td>
<td>string</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Common Response Headers

This document describes the common response headers that may be used in API requests. The headers mentioned below will not be detailed again in specific API documents.

Response Headers

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Length</td>
<td>HTTP response length in bytes as defined in RFC 2616</td>
<td>string</td>
</tr>
<tr>
<td>Content-Type</td>
<td>HTTP response content type (MIME) as defined in RFC 2616.</td>
<td>string</td>
</tr>
<tr>
<td>Connection</td>
<td>Indicates whether the network connection will be closed after completion of the response as defined in RFC 2616. Enumerated values: <code>keep-alive</code>, <code>close</code>.</td>
<td>Enum</td>
</tr>
<tr>
<td>Date</td>
<td>Server response time in GMT format as defined in RFC 1123, such as Wed, 29 May 2019 04:10:12 GMT.</td>
<td>string</td>
</tr>
<tr>
<td>Etag</td>
<td>An entity tag (ETag) is an information tag that identifies the content of an object when it is created, such as &quot;8e0b617ca296a564c33316a286ca69df&quot;. It can be used to check whether the content of the object has changed. This header does not necessarily return the MD5 checksum value of the object; instead, it may vary depending on how the object is uploaded and encrypted.</td>
<td>string</td>
</tr>
<tr>
<td>Server</td>
<td>Name of the server that accepts the request and returns the response. Default value: tencent-cos.</td>
<td>string</td>
</tr>
<tr>
<td>x-cos-request-id</td>
<td>An ID automatically generated by the server for each request when the request is sent.</td>
<td>string</td>
</tr>
<tr>
<td>x-cos-trace-id</td>
<td>An ID automatically generated by the server for each error when requests fail. This header will be included in the response only if a request fails.</td>
<td>string</td>
</tr>
</tbody>
</table>

Server-side Encryption Headers

For APIs that support server-side encryption (SSE) and use SSE in the request, the following response headers will be returned according to the specific encryption method. For more information, see Server-side Encryption Overview.

**SSE-COS**

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption</td>
<td>If an object is uploaded with SSE-COS or an SSE-COS-encrypted object is downloaded, the response of the request will return this header, indicating the server-side encryption algorithm used when the object is uploaded.</td>
<td>string</td>
</tr>
</tbody>
</table>

**SSE-C**

<table>
<thead>
<tr>
<th>Header Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption-customer-algorithm</td>
<td>If an object is uploaded with SSE-C or an SSE-C-encrypted object is downloaded, the response of the request will return this header, indicating the server-side encryption algorithm used when the object is uploaded.</td>
<td>string</td>
</tr>
<tr>
<td>Header Name</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>x-cos-server-side-encryption-customer-key-MD5</td>
<td>Base64-encoded MD5 hash of the server-side encryption key used during object upload. For example, U5L61r7jcw8Nt7f8eU58g==</td>
<td>string</td>
</tr>
</tbody>
</table>

**Error Codes**

Last updated: 2019-12-16 14:42:37
Overview

This document describes the error codes and corresponding error messages returned when a request fails.

Error Response

Content-Type: application/xml

Corresponding HTTP status code: 3XX, 4XX, or 5XX. Please note that for the PUT Object - Copy API, even if the HTTP status code is 200, the response body may still include an error.

Response Body

```xml
<?xml version='1.0' encoding='utf-8' ?>
<Error>
  <Code>string</Code>
  <Message>string</Message>
  <Resource>string</Resource>
  <RequestId>string</RequestId>
  <TraceId>string</TraceId>
</Error>
```

The nodes are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>None</td>
<td>Contains all error information</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node **Error**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Error</td>
<td>Error codes used to locate unique error conditions and determine error scenarios. See below for specific error codes</td>
<td>string</td>
</tr>
<tr>
<td>Message</td>
<td>Error</td>
<td>Specific error message</td>
<td>string</td>
</tr>
<tr>
<td>Resource</td>
<td>Error</td>
<td>Requested resource: bucket address or object address</td>
<td>string</td>
</tr>
<tr>
<td>RequestId</td>
<td>Error</td>
<td>An ID automatically generated by the server for each request; COS can use the ID to quickly locate problems</td>
<td>string</td>
</tr>
<tr>
<td>TraceId</td>
<td>Error</td>
<td>An ID automatically generated by the server for each error; COS can use the ID to quickly locate problems</td>
<td>string</td>
</tr>
</tbody>
</table>

Error Code List

**3XX errors**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>PermanentRedirect</td>
<td>This resource has been permanently relocated. Please use the HTTP Location response header to get redirected to the correct location</td>
<td>301 Moved Permanently</td>
</tr>
<tr>
<td>TemporaryRedirect</td>
<td>This resource has been temporarily relocated. Please use the HTTP Location response header to get redirected to the correct location</td>
<td>302 Moved Temporarily</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Redirect</td>
<td>Temporarily redirected</td>
<td>307 Moved Temporarily</td>
</tr>
<tr>
<td>TemporaryRedirect</td>
<td>You will be temporarily redirected during the DNS update</td>
<td>307 Moved Temporarily</td>
</tr>
</tbody>
</table>

### 4XX errors

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppendPositionErr</td>
<td>The object length doesn't match the position during the Append operation</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>AttachmentFull</td>
<td>The number of ACLs and policies has reached the upper limit</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>BadDigest</td>
<td>The provided Content-MD5 value is different from the MD5 hash of the request body received by the server</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>EntityTooLarge</td>
<td>The size of the uploaded object exceeds the specified maximum value</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>EntityTooSmall</td>
<td>The size of the uploaded object is below the specified minimum value. This error often occurs in multipart uploads</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>IncorrectNumberOfFilesInPostRequest</td>
<td>Only one object can be uploaded in one POST request</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Invalid request parameter</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidBucketName</td>
<td>Invalid bucket name</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidCopySource</td>
<td>Invalid source for object copy</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidDigest</td>
<td>The given Content-MD5 value is invalid</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidPart</td>
<td>Missing part</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidPartOrder</td>
<td>Part numbers are not continuous</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRegionName</td>
<td>Invalid region name</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Invalid request</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidSHA1Digest</td>
<td>Invalid SHA1 checksum of the request content</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidURI</td>
<td>Invalid URI</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>KeyTooLong</td>
<td>The object key is too long</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>LifecycleIdNotUnique</td>
<td>The lifecycle ID is not unique</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>LifecycleRuleConflicted</td>
<td>There is a conflict in the lifecycle settings</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MalformedPOSTRequest</td>
<td>The request body content of this POST request is invalid</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>The XML format of the request body does not conform to the XML syntax</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingAppid</td>
<td>APPID is missing in the request headers</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingContentMD5</td>
<td>Content-MD5 is missing in the request headers</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingHost</td>
<td>Host is missing in the request headers</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingRequestBodyError</td>
<td>Missing request body</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MultiBucketNotSupport</td>
<td>Only one destination bucket can be set for cross-region replication</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>NotSupportedStorageClass</td>
<td>The specified storage class is not supported</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>ObjectNotAppendable</td>
<td>The specified object cannot be appended</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>PolicyFull</td>
<td>The number of ACLs and policies has reached the upper limit</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>RequestTimeOut</td>
<td>Data read timed out. Check your network speed or reduce the number of concurrent uploads</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>TooManyBuckets</td>
<td>The number of buckets has reached the upper limit (200)</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>UnexpectedContent</td>
<td>The request does not support the content</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>VerifyAlgorithmNotSupported</td>
<td>The verification algorithm is not supported</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>WebsiteURLException</td>
<td>Invalid custom domain name URL</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>XMLSizeLimit</td>
<td>The length of XML has exceeded the limit</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>AccessDenied</td>
<td>Access denied due to wrong signature or permission</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>ExpiredToken</td>
<td>The signature string has expired</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>InvalidAccessKeyId</td>
<td>The SecretID does not exist</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>InvalidObjectState</td>
<td>The request content conflicts with the object attributes</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>InvalidObjectStorage</td>
<td>Invalid storage class</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>RequestTimeTooSkewed</td>
<td>The gap between the local time and the server time is over 15 minutes</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>SignatureDoesNotMatch</td>
<td>The signature calculated by the client does not match that calculated by the COS server</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>The specified bucket does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchCopySource</td>
<td>The object copy source does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchCORSConfiguration</td>
<td>The specified CORS configuration does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchKey</td>
<td>The specified object key does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchLifecycleConfiguration</td>
<td>The specified lifecycle configuration does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchTagSet</td>
<td>The specified tag set does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchUpload</td>
<td>The UploadId specified for the multipart upload does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchWebsiteConfiguration</td>
<td>The static website configuration does not exist</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>MethodNotAllowed</td>
<td>This resource does not support this HTTP method</td>
<td>405 Method Not Allowed</td>
</tr>
<tr>
<td>RestoreNonArchiveObject</td>
<td>It is not allowed to restore a non-archived object</td>
<td>405 Method Not Allowed</td>
</tr>
<tr>
<td>BucketAlreadyExists</td>
<td>The specified bucket already exists.</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>BucketAlreadyOwnedByYou</td>
<td>The specified bucket already exists and was created by the current account.</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>BucketNotEmpty</td>
<td>The bucket is not empty</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>InvalidBucketState</td>
<td>The bucket state conflicts with the operational request; for example, versioning configuration conflicts with cross-region replication</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>PathConflict</td>
<td>Millisecond-level concurrency conflict occurred for objects with the same name</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>RestoreAlreadyInProgress</td>
<td>This object is being restored</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>MissingContentLength</td>
<td>The Content-Length request header is missing</td>
<td>411 Length Required</td>
</tr>
<tr>
<td>PreconditionFailed</td>
<td>Precondition match failed</td>
<td>412 Precondition</td>
</tr>
<tr>
<td>InvalidRange</td>
<td>The requested object range is invalid</td>
<td>416 Requested Range Not Satisfiable</td>
</tr>
<tr>
<td>UnavailableForLegalReasons</td>
<td>Unavailable for legal reasons</td>
<td>451 Unavailable For Legal Reasons</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>InternalError</td>
<td>Internal server error</td>
<td>500 Internal Server</td>
</tr>
<tr>
<td>NotImplemented</td>
<td>The request has not been implemented yet</td>
<td>501 Not Implemented</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>The service is temporarily unavailable. Please try again</td>
<td>503 Service Unavailable</td>
</tr>
<tr>
<td>SlowDown</td>
<td>Please reduce the access frequency</td>
<td>503 Slow Down</td>
</tr>
</tbody>
</table>
Request Signature

Last updated: 2019-08-06 19:24:35

1. This document is only for the COS XML edition.
2. It is not applicable for HTTP requests for posting objects.

RESTful APIs support anonymous and signed HTTP requests, which help you use COS resources.

- Anonymous request: A request that does not contain any authentication information.
- Signed request: A request that is signed with authentication information. The COS server authenticates the requester of the signed request and only executes the authenticated ones. Otherwise, it returns an error message and denies the request.

The COS server performs HMAC (Hash Message Authentication Code) authentication schema.

Signature Use Cases

When using COS to store the data allowed public access, you can set the objects to public read and private write, i.e., they can be viewed by everyone and only be written by the accounts specified by ACL policy. To control access permissions and validity periods of the operations, you can also add the signature to the ACL policy to authenticate access requests.

The API request signature described in this document is already included in the SDK. You need to follow the steps below only if you want to redevelop based on the initial APIs.

In this case, multiple layers of security protection can be applied to API requests:

1. **Requester authentication**. Authenticate requesters with unique credentials (ID and key).
2. **Data tampering prevention during transmission**. Data is signed and verified to ensure the integrity.
3. **Signature theft prevention**. Assign validity period to the signature to protect it from being disclosed and reused by third parties.

Preparations

1. **APPID, SecretId, and SecretKey**: They can be obtained on the API Key Management page in the CAM Console.
2. **Programming language**: Supported languages include without limitation Java, PHP, C#, C++, Node.js, and Python. You should determine the corresponding HMAC-SHA1, SHA1, and `UrlEncode` functions according to the actual language.

The HMAC-SHA1 and SHA1 functions take UTF-8 encoded strings as input and hexadecimal lowercase strings as output. `UrlEncode` is based on UTF-8 encoding. For printable characters in the ASCII range, the following special symbols should also be encoded:

<table>
<thead>
<tr>
<th>Character</th>
<th>Decimal</th>
<th>Hex</th>
<th>Character</th>
<th>Decimal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>(space)</td>
<td>32</td>
<td>20</td>
<td>;</td>
<td>59</td>
<td>3B</td>
</tr>
<tr>
<td>!</td>
<td>33</td>
<td>21</td>
<td>&lt;</td>
<td>60</td>
<td>3C</td>
</tr>
<tr>
<td>*</td>
<td>34</td>
<td>22</td>
<td>=</td>
<td>61</td>
<td>3D</td>
</tr>
<tr>
<td>#</td>
<td>35</td>
<td>23</td>
<td>&gt;</td>
<td>62</td>
<td>3E</td>
</tr>
<tr>
<td>$</td>
<td>36</td>
<td>24</td>
<td>?</td>
<td>63</td>
<td>3F</td>
</tr>
<tr>
<td>%</td>
<td>37</td>
<td>25</td>
<td>@</td>
<td>64</td>
<td>40</td>
</tr>
<tr>
<td>&amp;</td>
<td>38</td>
<td>26</td>
<td></td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>`</td>
<td>39</td>
<td>27</td>
<td>\</td>
<td>92</td>
<td>5C</td>
</tr>
</tbody>
</table>
### Signing Process

#### Generating KeyTime
1. Get the Unix timestamp "StartTimestamp" corresponding to the current time, which is the total number of seconds starting from January 1, 1970, 00:00:00 UTC (January 1, 1970, 08:00:00 Beijing time).
2. Calculate the Unix timestamp "EndTimestamp" corresponding to the signature expiry time based on the timestamp above and the expected validity period of the signature.
3. Splice the signature validity period in the format of StartTimestamp;EndTimestamp, which is the KeyTime.

**Example:** 1557902800;1557910000

#### Generating SignKey
Calculate the message digest (hash value) using HMAC-SHA1 with SecretKey as the key and KeyTime as the message, which is SignKey.

**Example:** 36bcd76db8c9f066472fec403df8a34cab34c77

#### Generating UrlParamList and HttpParameters
1. Traverse the HTTP request parameters, generate the key-to-value mapping "Map" and the key list "KeyList", where key is converted to lowercase and value is UrlEncoded.
   - In parameters without value, the value is considered to be an empty string. For example, if the request path is /?acl, it is considered to be /?acl=
2. Sort the KeyList in lexicographical order.
3. UrlEncode the keys in the Map and KeyList and convert them to lowercase too.
4. Splice all key-value pairs in the Map in the order of the KeyList in the format of key1=value1&key2=value2&key3=value3, which is the HttpParameters.
5. Splice all items in the KeyList in the order of the KeyList in the format of key1;key2;key3, which is the UrlParamList.

**Example:**

- **Example 1:**
  
  Request path: /?prefix=example-folder%2F&delimiter=%2F&max-keys=10
  
  UrlParamList: delimiter;max-keys;prefix
  
  HttpParameters: delimiter=%2F&max-keys=10&prefix=example-folder%2F

  The request parameters in the request path are UrlEncoded too when the request is actually sent; therefore, be careful not to repeat the UrlEncoding.

- **Example 2:**
  
  Request path: /exampleobject?acl
  
  UrlParamList: acl
  
  HttpParameters: acl

#### Generating HeaderList and HttpHeaders
Generate the HeaderList and HttpHeaders in the same way as described in Generating UrlParamList and HttpParameters, where the HTTP request parameters used for generation need to be replaced with HTTP request headers. The generated HttpParameters is the HttpHeaders, while the
generated UrlParamList is the HeaderList.

**Example:**
Request header:

```
Date: Thu, 16 May 2019 03:15:06 GMT
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
x-cos-acl: private
x-cos-grant-read: uin="100000000011"
```

Calculate the HeaderList as `date;host;x-cos-acl;x-cos-grant-read` and HttpHeaders as `date=Thu%2C%2016%20May%202019%2003%3A15%3A06%20GMT&host=examplebucket-1250000000.cos.ap-shanghai.myqcloud.com&x-cos-acl=private&x-cos-grant-read=uin%3D%22100000000011%22`.

**Generating HttpRequest**

Generate the HttpRequest based on the HTTP method, HTTP request path, HttpParameters, and HttpHeaders in the format of

```
HttpRequest
(UriPathname)
(HttpParameters)
(HttpHeaders)
```

Here, HttpMethod is converted to lowercase, such as `get` or `put`; UriPathname is the request path, such as `/` or `/exampleobject`; and `n` is a line break (if a string is empty, the line breaks before and after it need to be retained, such as `get
/exampleobject

`).

**Generating StringToSign**

Generate the StringToSign based on the `KeyTime` and HttpRequest in the format of `sha1
KeyTime
SHA1(HttpRequest)`.

Here, SHA1(HttpRequest) is the message digest generated by calculating the HttpRequest using SHA1.

**Generating Signature**

Calculate the message digest using HMAC-SHA1 with `SignKey` as the key and [StringToSign] as the message, which is the Signature.

**Generating the Actual Signature**

Generate the actual signature based on `SecretId`, `KeyTime`, `HeaderList`, `UrlParamList`, and `Signature` in the following format:

```
q-sign-algorithm=sha1
&q-ak=SecretId
&q-sign-time=KeyTime
&q-key-time=KeyTime
&q-header-list=HeaderList
&q-url-param-list=UrlParamList
&q-signature=Signature
```

Line breaks in the example above are for easy understanding only and are not included in a real signature.

**Signature Use**

Signed HTTP requests initiated to COS via RESTful APIs can pass the signature in the following ways:

1. Pass through a standard HTTP Authorization header, such as `Authorization: q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753;1557996953&...&q-signature=...`

2. Pass as an HTTP request parameter (be sure to UrlEncode), such as `/exampleobject?q-sign-algorithm=sha1&q-ak=...&q-sign-time=1557989753;1557996953&...&q-signature=...`

In the example above, `...` is used to substitute the specific signature content.

**Sample Code**

**Pseudocode**
Sample Message Digest Calculation

The samples below illustrate how to call HMAC-SHA1 in different languages:

**PHP**

```php
$sha1HttpString = sha1('ExampleHttpString');

$signKey = hash_hmac('sha1', 'ExampleKeyTime', 'YourSecretKey');
```

**Java**

```java
import org.apache.commons.codec.digest.DigestUtils;
import org.apache.commons.codec.digest.HmacUtils;

String sha1HttpString = DigestUtils.sha1Hex("ExampleHttpString");

String signKey = HmacUtils.hmacSha1Hex("YourSecretKey", "ExampleKeyTime");
```

**Python**

```python
import hmac
import hashlib

sha1 = hashlib.sha1()
sha1_http_string = sha1.update('ExampleHttpString'.encode('utf-8')).hexdigest()

sign_key = hmac.new('YourSecretKey'.encode('utf-8'), 'ExampleKeyTime'.encode('utf-8'), hashlib.sha1).hexdigest()
```

**Node.js**

```javascript
var crypto = require('crypto');

var sha1HttpString = crypto.createHash('sha1').update('ExampleHttpString').digest('hex');
var signKey = crypto.createHmac('sha1', 'YourSecretKey').update('ExampleKeyTime').digest('hex');
```

**Go**

```go
import (
    "crypto/hmac"
    "crypto/sha1"
)

h := sha1.New()
h.Write([]byte("ExampleHttpString"))
sha1HttpString := h.Sum(nil)

var hashFunc = sha1.New
h = hashFunc.New(hashFunc, []byte("YourSecretKey"))
h.Write([]byte("ExampleKeyTime"))
signKey := h.Sum(nil)
```

**Use Case**

**Preparations**

Obtain your APPID, SecretId, and SecretKey on the API Key Management page in the CAM Console. Below is an example:
### Uploading an Object

#### Original Request

PUT /exampleobject(Example object) HTTP/1.1  
Date: Thu, 16 May 2019 06:45:51 GMT  
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com  
Content-Type: text/plain  
Content-Length: 13  
x-cos-acl: private  
x-cos-grant-read: uin="1000000000111

ObjectContent

#### Intermediate Variables

- KeyTime = 1557989151;1557996351
- SignKey = eb2519b49b8b2ac213cb1f3d1a3d27a3b3c9bc5f
- UrlParamList = (empty string)
- HttpHeaders = content-length;content-md5;content-type;date;host;x-cos-acl;x-cos-grant-read
- HttpString = put
/exampleobject(tencentcloud)

content-length=13&content-md5=mQ%2FfVh815F3k6TAUm8m0eg==
content-type=text/plain&date=Thu%2C%2016%20May%202019%2006%3A45%3A51%20GMT&host=examplebucket-1250000000.cos.ap-beijing.myqcloud.com&x-cos-acl=private&x-cos-grant-read=uin%3D%22100000000011%22

StringToSign = sha1
1557989151;1557996351
8b2751e77f43a0995d6e9eb947774885c3a4172

Signature = 3b8851a1a569213c17ba8fa7dcf2abec6935172

Here, (empty string) represents an empty string with a length of 0 and a line break.

#### Signed Request

PUT /exampleobject(Example object) HTTP/1.1  
Date: Thu, 16 May 2019 06:45:51 GMT  
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com  
Content-Type: text/plain  
Content-Length: 13  
x-cos-acl: private  
x-cos-grant-read: uin="1000000000111

ObjectContent

### Downloading an Object

#### Original Request

GET /exampleobject(Example object) HTTP/1.1  
Date: Thu, 16 May 2019 06:55:53 GMT  
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com

#### Intermediate Variables

- KeyTime = 1557989953;1557996953
- SignKey = 937914bf490e9e8c18983642052e4feeb35eaf

©2013-2019 Tencent Cloud. All rights reserved.
**Signed Request**

```
GET /exampleobject(tencentcloud)?response-content-type=application%2Foctet-stream&response-cache-control=max-age%3D600 HTTP/1.1
Date: Thu, 16 May 2019 06:55:53 GMT
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDQjz3ltompVjBni5LitkWHFlFpwkn9U5q&q-sign-time=1557989753;1557996953&q-key-time=1557989753;1557996953&q-header-list=date;host&q-url-param-list=response-cache-control;response-content-type&q-signature=01681b8c9d798a678e43b685a9f1bbaf6c0e812
```

Here, `\n` represents a line break.
## Operation list

Last updated : 2020-02-28 18:14:50

Tencent Cloud Cloud Object Storage service (COS)-related APIs and instructions are as follows:

### Service interface

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET Service</td>
<td>Query Bucket list</td>
<td>Query the list of all Bucket under the specified account</td>
</tr>
</tbody>
</table>

### Bucket interface

#### Basic operation interface

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket</td>
<td>Create Bucket</td>
<td>Create a Bucket under the specified account</td>
</tr>
<tr>
<td>GET Bucket (List Object)</td>
<td>Query object list</td>
<td>Query some or all of the objects under Bucket</td>
</tr>
<tr>
<td>HEAD Bucket</td>
<td>Search Bucket and his Permission</td>
<td>Confirm whether Bucket exists and whether there is Permission and Access</td>
</tr>
<tr>
<td>DELETE Bucket</td>
<td>Deleting Buckets</td>
<td>Delete the empty Bucket under the specified account</td>
</tr>
<tr>
<td>GET Bucket Object Versions</td>
<td>Query object version</td>
<td>Query some or all of the objects under Bucket and their historical version information</td>
</tr>
</tbody>
</table>

#### Access controls (acl) interface

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket acl</td>
<td>Set Bucket ACL</td>
<td>Set up the specified Bucket Access Permission control list</td>
</tr>
<tr>
<td>GET Bucket acl</td>
<td>Query Bucket ACL</td>
<td>Query Bucket's Access control list</td>
</tr>
</tbody>
</table>

#### Cross-domain resource sharing (cors) interface

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket cors</td>
<td>Set cross-domain configuration</td>
<td>Set up Bucket's cross-domain Access and Permission</td>
</tr>
<tr>
<td>GET Bucket cors</td>
<td>Query cross-domain configuration</td>
<td>Query Bucket's cross-domain Access configuration information</td>
</tr>
<tr>
<td>DELETE Bucket cors</td>
<td>Delete cross-domain configuration</td>
<td>Delete Bucket's cross-domain Access configuration information</td>
</tr>
</tbody>
</table>

#### Lifecycle (lifecycle) interface

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket lifecycle</td>
<td>Setting Lifecycle</td>
<td>Set up the configuration of Bucket's life cycle management</td>
</tr>
<tr>
<td>API</td>
<td>Operation name</td>
<td>pedagogical operation</td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>GET Bucket lifecycle</td>
<td>Query life cycle</td>
<td>Query the configuration of Bucket's life cycle management</td>
</tr>
<tr>
<td>DELETE Bucket lifecycle</td>
<td>Delete Lifecycle</td>
<td>Delete Bucket's life cycle management configuration</td>
</tr>
</tbody>
</table>

**Bucket Strategy (policy) Interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket policy</td>
<td>Set Bucket strategy</td>
<td>Set the Permission strategy for the specified Bucket</td>
</tr>
<tr>
<td>GET Bucket policy</td>
<td>Query Bucket's strategy</td>
<td>Query the Permission strategy of specified Bucket</td>
</tr>
<tr>
<td>DELETE Bucket policy</td>
<td>Delete Bucket policy</td>
<td>Delete the Permission strategy of the specified Bucket</td>
</tr>
</tbody>
</table>

**Hotlink protection Prevent hotlinking (referer) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket referer</td>
<td>Set Bucket referer</td>
<td>Set Bucket Referer whitelist or blacklist</td>
</tr>
<tr>
<td>GET Bucket Referer</td>
<td>Query Bucket referer</td>
<td>Query Bucket Referer whitelist or blacklist</td>
</tr>
</tbody>
</table>

**Bucket tag (tagging) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket tagging</td>
<td>Set Bucket tag</td>
<td>Set the label for the existing Bucket</td>
</tr>
<tr>
<td>GET Bucket tagging</td>
<td>Query Bucket tag</td>
<td>Query the existing Bucket tags under the specified Bucket</td>
</tr>
<tr>
<td>DELETE Bucket tagging</td>
<td>Delete Bucket tag</td>
<td>Delete the specified Bucket tag</td>
</tr>
</tbody>
</table>

**Static website (website) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket website</td>
<td>Setting up a Static Website</td>
<td>Configure a static website for Bucket</td>
</tr>
<tr>
<td>GET Bucket website</td>
<td>Query static site configuration</td>
<td>Query the static website configuration information with Bucket and Associate</td>
</tr>
<tr>
<td>DELETE Bucket website</td>
<td>Delete static site configuration</td>
<td>Delete the static website configuration information of the specified Bucket</td>
</tr>
</tbody>
</table>

**Inventory (inventory) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket inventory</td>
<td>Set up inventory job</td>
<td>Create inventory and job in Bucket</td>
</tr>
<tr>
<td>GET Bucket inventory</td>
<td>Inquire about inventory job</td>
<td>Query Bucket's specified inventory configuration information</td>
</tr>
<tr>
<td>List Bucket Inventory Configurations</td>
<td>Inquire about all inventory</td>
<td>Inquire about all inventory and job of Bucket</td>
</tr>
<tr>
<td>DELETE Bucket inventory</td>
<td>Delete inventory job</td>
<td>Delete inventory and job specified in Bucket</td>
</tr>
</tbody>
</table>

**Version control (versioning) interface**
<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket versioning</td>
<td>Setting Versioning</td>
<td>Enable or suspend Bucket's version control function</td>
</tr>
<tr>
<td>GET Bucket versioning</td>
<td>Query version control</td>
<td>Query Bucket's version control information</td>
</tr>
</tbody>
</table>

**Copy (replication) interfaces across regions**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket replication</td>
<td>Setting Cross-region Replication</td>
<td>Configure cross-region replication rules for Bucket with version control enabled</td>
</tr>
<tr>
<td>GET Bucket replication</td>
<td>Query cross-region replication</td>
<td>Query Bucket's cross-region replication configuration information</td>
</tr>
<tr>
<td>DELETE Bucket replication</td>
<td>Delete cross-region replication</td>
<td>Delete Bucket's cross-region replication configuration information</td>
</tr>
</tbody>
</table>

**Log management (logging) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket logging</td>
<td>Set up log management</td>
<td>Enable logging for source Bucket</td>
</tr>
<tr>
<td>GET Bucket logging</td>
<td>Query log management</td>
<td>Query the log configuration information of the source Bucket</td>
</tr>
</tbody>
</table>

**Bucket encrypts (encryption) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Bucket encryption</td>
<td>Set Bucket encryption</td>
<td>Set the default encryption configuration under the specified Bucket</td>
</tr>
<tr>
<td>GET Bucket encryption</td>
<td>Query Bucket encryption</td>
<td>Query the default encryption configuration under the specified Bucket</td>
</tr>
<tr>
<td>DELETE Bucket encryption</td>
<td>Delete Bucket encryption</td>
<td>Delete the default encryption configuration under the specified Bucket</td>
</tr>
</tbody>
</table>

**Object interface**

**Basic operation interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Object</td>
<td>Simple upload object</td>
<td>Upload an object to Bucket</td>
</tr>
<tr>
<td>PUT Object-Copy</td>
<td>Set up object replication</td>
<td>Copy files to the destination path</td>
</tr>
<tr>
<td>POST Object</td>
<td>Form upload object</td>
<td>Use the form to request uploading objects</td>
</tr>
<tr>
<td>GET Object</td>
<td>Download Object</td>
<td>Download an object locally</td>
</tr>
<tr>
<td>HEAD Object</td>
<td>Query object metadata</td>
<td>Query the metadata information of an object</td>
</tr>
<tr>
<td>DELETE Object</td>
<td>Deleting a Single Object</td>
<td>Delete the specified object in Bucket</td>
</tr>
<tr>
<td>DELETE Multiple Objects</td>
<td>Deleting Multiple Objects</td>
<td>Delete objects in batch in Bucket</td>
</tr>
<tr>
<td>OPTIONS Object</td>
<td>Pre-request cross-domain configuration</td>
<td>Use pre-requests to confirm whether a true cross-domain request can be sent</td>
</tr>
<tr>
<td>API</td>
<td>Operation name</td>
<td>pedagogical operation</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>POST Object restore</td>
<td>Resume archived object</td>
<td>Retrieve objects of archive type back to Access</td>
</tr>
<tr>
<td>SELECT Object Content</td>
<td>Retrieve object content</td>
<td>Retrieve the contents of the specified object</td>
</tr>
</tbody>
</table>

**Access control interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUT Object acl</td>
<td>Set object ACL</td>
<td>Set the Access control list for an object in Bucket</td>
</tr>
<tr>
<td>GET Object acl</td>
<td>Query object ACL</td>
<td>Access control list for querying objects</td>
</tr>
</tbody>
</table>

**Multipart upload interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate Multipart Upload</td>
<td>Initialize multipart upload</td>
<td>Initialize multipart upload job</td>
</tr>
<tr>
<td>Upload Part</td>
<td>Upload part</td>
<td>Multipart upload document</td>
</tr>
<tr>
<td>Upload Part-Copy</td>
<td>Copy part</td>
<td>Copy other objects as a part</td>
</tr>
<tr>
<td>Complete Multipart Upload</td>
<td>Complete multipart upload</td>
<td>Multipart upload who completed the whole document</td>
</tr>
<tr>
<td>Abort Multipart Upload</td>
<td>Terminate multipart upload</td>
<td>Terminate a multipart upload operation and delete the uploaded block</td>
</tr>
<tr>
<td>List Multipart Uploads</td>
<td>Inquire about multipart upload</td>
<td>Query the information of multipart upload in progress</td>
</tr>
<tr>
<td>List Parts</td>
<td>Query uploaded blocks</td>
<td>Query uploaded blocks in a specific multipart upload operation</td>
</tr>
</tbody>
</table>

**Batch processing (batch) interface**

<table>
<thead>
<tr>
<th>API</th>
<th>Operation name</th>
<th>pedagogical operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateJob</td>
<td>Create task</td>
<td>Used to create batch processing job in Bucket</td>
</tr>
<tr>
<td>DescribeJob</td>
<td>Describe job</td>
<td>Used to obtain the parameters of the created batch processing job and the execution status of job</td>
</tr>
<tr>
<td>ListJobs</td>
<td>Inquire about job</td>
<td>Used to list the batch processing job that has been created</td>
</tr>
<tr>
<td>UpdateJobPriority</td>
<td>Update job priority</td>
<td>Used to update the priority of the created job</td>
</tr>
<tr>
<td>UpdateJobStatus</td>
<td>Update job status</td>
<td>Used to update the status of the created job</td>
</tr>
</tbody>
</table>
Service APIs
GET Service

Description
This API is used to query the list of all buckets under a requester's account or in a specific region.

Request

Sample Request

Sample 1

GET / HTTP/1.1
Host: service.cos.myqcloud.com
Date: GMT Date
Authorization: Auth String

Sample 2

GET / HTTP/1.1
Host: cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

- Authorization: Auth String (see Request Signature for more information).
- Host: specify this as service.cos.myqcloud.com to query the list of all buckets; specify this as cos.<Region>.myqcloud.com to query the list of buckets in a specific region.

Request Parameters
This API does not use any request parameter.

Request Headers
This API only uses common request headers. For more information on common request headers, see Common Request Headers.

Request Body
This API does not have a request body.

Response

Response Headers
This API only returns common response headers. For more information, see Common Response Headers.

Response Body
A successful query returns application/xml data which include the list of all buckets or the list of buckets in a specific region.

<ListAllMyBucketsResult>
<Owner>
<ID>string</ID>
<DisplayName>string</DisplayName>
</Owner>
<Buckets>
<Bucket>
<Name>string</Name>
<Location>Enum</Location>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>
The nodes are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListAllMyBucketsResult</td>
<td>None</td>
<td>Stores the result of the GET Service request</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node ListAllMyBucketsResult:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>ListAllMyBucketsResult</td>
<td>Bucket owner information</td>
<td>Container</td>
</tr>
<tr>
<td>Buckets</td>
<td>ListAllMyBucketsResult</td>
<td>Bucket list</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node Owner:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ListAllMyBucketsResult.Owner</td>
<td>Complete ID of the bucket owner in the format of qcs::cam::uin[/OwnerUin]:uin[/OwnerUin], such as qcs::cam::uin/100000000001:uin/100000000001</td>
<td>string</td>
</tr>
<tr>
<td>DisplayName</td>
<td>ListAllMyBucketsResult.Owner</td>
<td>Bucket owner name</td>
<td>string</td>
</tr>
</tbody>
</table>

**Content of the Container node Buckets:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>ListAllMyBucketsResult.Buckets</td>
<td>Bucket information</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node Buckets, Bucket:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ListAllMyBucketsResult.Buckets.Bucket</td>
<td>Bucket name in the format of &lt;BucketName-APPID&gt;, such as examplebucket-1250000000</td>
<td>string</td>
</tr>
<tr>
<td>Location</td>
<td>ListAllMyBucketsResult.Buckets.Bucket</td>
<td>Bucket region such as ap-beijing, ap-hongkong, and eu-frankfurt. For the enumerated values, see Regions and Access Domain Names</td>
<td>Enum</td>
</tr>
<tr>
<td>CreationDate</td>
<td>ListAllMyBucketsResult.Buckets.Bucket</td>
<td>Bucket creation time in ISO8601 format, such as 2019-05-24T10:56:40Z</td>
<td>date</td>
</tr>
</tbody>
</table>

**Error Codes**

There is no special error message for this API. For all error messages, see Error Codes.
Examples

Example 1. Querying the list of all buckets

Request

GET / HTTP/1.1
Host: service.cos.myqcloud.com
Date: Fri, 24 May 2019 11:59:50 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNn82oY1glJQGF0c3J0****&q-sign-time=1558699198;1558706300&q-key-time=1558699198;1558706300&q-header-list=date;host&q-url-param-list=&q-signature=89fa1f6a56c34e46f3dbd65f928a8f34a****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 805
Connection: close
Date: Fri, 24 May 2019 11:59:51 GMT
Server: tencent-cos
x-cos-request-id: NWNlN2RjYjdfOGFiMjM1MGFfNTVjMl8zMmI1****

<ListAllMyBucketsResult>
<Owner>
<ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
<DisplayName>100000000001</DisplayName>
</Owner>
<Buckets>
<Bucket>
<Name>examplebucket1-1250000000</Name>
<Location>ap-beijing</Location>
<CreationDate>2019-05-24T11:49:50Z</CreationDate>
</Bucket>
<Bucket>
<Name>examplebucket2-1250000000</Name>
<Location>ap-beijing</Location>
<CreationDate>2019-05-24T11:51:50Z</CreationDate>
</Bucket>
<Bucket>
<Name>examplebucket3-1250000000</Name>
<Location>eu-frankfurt</Location>
<CreationDate>2019-05-24T11:53:50Z</CreationDate>
</Bucket>
<Bucket>
<Name>examplebucket4-1250000000</Name>
<Location>eu-frankfurt</Location>
<CreationDate>2019-05-24T11:55:50Z</CreationDate>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>

Example 2. Querying the list of buckets in a specific region

Request

GET / HTTP/1.1
Host: cos.ap-beijing.myqcloud.com
Date: Fri, 24 May 2019 11:59:51 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNn82oY1glJQGF0c3J0****&q-sign-time=1558699199;1558706300&q-key-time=1558699199;1558706300&q-header-list=date;host&q-url-param-list=&q-signature=89fa1f6a56c34e46f3dbd65f928a8f34a****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 495
<ListAllMyBucketsResult>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
    <DisplayName>100000000001</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>examplebucket1-1250000000</Name>
      <Location>ap-beijing</Location>
      <CreationDate>2019-05-24T11:49:50Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>examplebucket2-1250000000</Name>
      <Location>ap-beijing</Location>
      <CreationDate>2019-05-24T11:51:50Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
Bucket APIs
Basic Operations
PUT Bucket

Description
This API is used to create a bucket under a specified account. This API does not support anonymous requests; only a request carrying the Authorization signature can create a new bucket. The bucket creator is the bucket owner by default.

If no access permission is specified for a bucket when it is created, the private Read/Write (private) access will be used by default.

Request

Sample Request

PUT / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: 0
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).

Request Parameters

This API does not use any request parameter.

Request Headers

In addition to common request headers, this API also supports the following request headers. For more information on common request headers, see Common Request Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-acl</td>
<td>Defines the access control list (ACL) attribute of the bucket. For the enumerated values such as private and public-read, see the &quot;Preset ACL for buckets&quot; section in ACL Overview. Default value: private</td>
<td>Enum</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read</td>
<td>Allows grantee to read the bucket; format: id=[OwnerUin], such as id=&quot;100000000001&quot;. You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-write</td>
<td>Allows grantee to write to the bucket; format: id=[OwnerUin], such as id=&quot;100000000001&quot;. You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read-acp</td>
<td>Allows grantee to read the ACL and the policy of the bucket; format: id=[OwnerUin], such as id=&quot;100000000001&quot;. You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-write-acp</td>
<td>Allows grantee to write to the ACL and the policy of the bucket; format: id=[OwnerUin], such as id=&quot;100000000001&quot;. You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-full-control</td>
<td>Grants a user full permission to perform operations on the bucket; format: id=[OwnerUin], such as id=&quot;100000000001&quot;. You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>
Request Body
This API does not have a request body.

Response

Response Headers
This API only returns common response headers. For more information, see Common Response Headers.

Response Body
The response body of this API is empty.

Error Codes
The implementation of this operation returns the following special error messages. For all error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>BucketAlreadyExists</td>
<td>The specified bucket already exists.</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>BucketAlreadyOwnedByYou</td>
<td>The specified bucket already exists and was created by the current account.</td>
<td>409 Conflict</td>
</tr>
</tbody>
</table>

Examples

Example 1. Basic example

Request

PUT / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Sun, 26 May 2019 14:51:38 GMT
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1558882298;1558889498&q-key-time=1558882298;1558889498&q-header-list=content-length;date;host&q-url-param-list=&q-signature=c25fd640274a6da2318935ceebfbcba4598****
Connection: close

Response

HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Sun, 26 May 2019 14:51:37 GMT
Server: tencent-cos
x-cos-request-id: NWNlYWE3ZjlfZDQyNzVkJRNgNzg1N18yNzFh****

Example 2. Specifying Public-Read and granting a specific user permission to read the permissions and write to the bucket

Request

PUT / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 14 Jun 2019 13:48:59 GMT
x-cos-acl: public-read
x-cos-grant-write: id="100000000002"
x-cos-grant-read-acp: id="100000000002"
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1560520139;1560527339&q-key-time=1560520139;1560527339&q-header-list=content-length;date;host&x-cos-acl;x-cos-grant-read-acp;x-cos-grant-write&q-url-param-list=&q-signature=df03e7917270e0bf2b679bc5f99793bd0c63****
Connection: close

Response
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 14 Jun 2019 13:49:00 GMT
Server: tencent-cos
x-cos-request-id: NMqwM2E1Y2NhZjBxOD8iMTI1OTYyNzNpZDRi****

GET Bucket (List Object)

Last updated: 2019-12-18 10:19:12
Description

The GET Bucket request is equivalent to the List Object request and can be used to list some or all of the objects in a bucket. To make this request, you need to have the permission to read the bucket.

If you upload an object to the bucket and immediately call the GET Bucket API, due to the eventual consistency characteristic of this API, the response may not include the just uploaded object.

Request

Sample Request

GET / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String (see Request Signature for details).

Request Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Matching prefix for object keys, which sets that the response will only contain object keys with the specified prefix</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>delimiter</td>
<td>A character delimiter used to group object keys. The identical paths between prefix or, if no prefix is specified, the beginning and the first delimiter are grouped and defined as a Prefix node under CommonPrefixes. The grouped object keys will no longer appear in the subsequent object list. For specific scenarios and usage, see the examples below</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>encoding-type</td>
<td>Specifies the encoding type of the returned value. Valid value: url, which means that the returned object keys are URL-encoded (percent-encoded) values. For example, &quot;Tencent Cloud&quot; will be encoded as Tencent%20Cloud</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>marker</td>
<td>Marker for the starting object key. Object key entries will be returned in UTF-8 lexicographical order starting from the first object key after the marker</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>max-keys</td>
<td>Maximum number of entries returned at a time. Default value: 1,000; maximum value: 1,000</td>
<td>integer</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information on common request headers, see Common Request Headers.

Request Body

This API does not have a request body.

Response

Response Headers

In addition to common response headers, this API returns the following response headers. For more information, see Common Response Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
</table>
Response Body

A successful query returns application/xml data which contain information on the objects in the bucket. For response bodies in different scenarios, see the examples below.

```xml
<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
  <Name>string</Name>
  <Prefix>string</Prefix>
  <Marker/>
  <MaxKeys>integer</MaxKeys>
  <Delimiter>string</Delimiter>
  <IsTruncated>boolean</IsTruncated>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>string</Prefix>
  </CommonPrefixes>
  <Contents>
    <Key>string</Key>
    <LastModified>date</LastModified>
    <ETag>string</ETag>
    <Size>integer</Size>
    <Owner>
      <ID>string</ID>
      <DisplayName>string</DisplayName>
    </Owner>
    <StorageClass>Enum</StorageClass>
  </Contents>
  <Contents>
    <Key>string</Key>
    <LastModified>date</LastModified>
    <ETag>string</ETag>
    <Size>integer</Size>
    <Owner>
      <ID>string</ID>
      <DisplayName>string</DisplayName>
    </Owner>
    <StorageClass>Enum</StorageClass>
  </Contents>
</ListBucketResult>
```

The nodes are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListBucketResult</td>
<td>None</td>
<td>Stores the result of the GET Bucket request</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node ListBucketResult:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ListBucketResult</td>
<td>Bucket name in the format of <code>&lt;BucketName-APPID&gt;</code>, such as <code>examplebucket-1250000000</code></td>
<td>string</td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>EncodingType</td>
<td>ListBucketResult</td>
<td>Encoding format, which corresponds to the encoding-type parameter in the request and will be returned only if the encoding-type parameter is specified in the request</td>
<td>string</td>
</tr>
<tr>
<td>Prefix</td>
<td>ListBucketResult</td>
<td>Matching prefix for object keys, which corresponds to the prefix parameter in the request</td>
<td>string</td>
</tr>
<tr>
<td>Marker</td>
<td>ListBucketResult</td>
<td>Marker for the starting object key. Object key entries will be returned in UTF-8 lexicographical order starting from the first object key after the marker. This parameter corresponds to the marker parameter in the request</td>
<td>string</td>
</tr>
<tr>
<td>MaxKeys</td>
<td>ListBucketResult</td>
<td>Maximum number of entries returned in a single response, which corresponds to the max-keys parameter in the request</td>
<td>integer</td>
</tr>
<tr>
<td>Delimiter</td>
<td>ListBucketResult</td>
<td>Delimiter, which corresponds to the delimiter parameter in the request and will be returned only if the delimiter parameter is specified in the request</td>
<td>string</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>ListBucketResult</td>
<td>Whether the returned list is truncated, which is a boolean value. Valid values: true, false</td>
<td>boolean</td>
</tr>
<tr>
<td>NextMarker</td>
<td>ListBucketResult</td>
<td>This node will be returned only if the returned list is truncated (i.e., the value of IsTruncated is true). The value of this parameter is the last object key in the current response and will be passed in as the marker parameter in the next request if you need to request subsequent entries</td>
<td>string</td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>ListBucketResult</td>
<td>The identical paths between prefix or, if no prefix is specified, the beginning and the first delimiter are grouped and defined as a common prefix. This node will be returned only if the delimiter parameter is specified in the request</td>
<td>Container</td>
</tr>
<tr>
<td>Contents</td>
<td>ListBucketResult</td>
<td>Object entries</td>
<td>Container</td>
</tr>
</tbody>
</table>
### Content of the Container node **CommonPrefixes**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>ListBucketResult.CommonPrefixes</td>
<td>A single common prefix</td>
<td>string</td>
</tr>
</tbody>
</table>

### Content of the Container node **Contents**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>ListBucketResult.Contents</td>
<td>Object key</td>
<td>string</td>
</tr>
<tr>
<td>LastModified</td>
<td>ListBucketResult.Contents</td>
<td>Last modified time of an object in ISO8601 format, such as <code>2019-05-24T10:56:40Z</code></td>
<td>date</td>
</tr>
<tr>
<td>ETag</td>
<td>ListBucketResult.Contents</td>
<td>Hash value calculated based on the file content</td>
<td>string</td>
</tr>
<tr>
<td>Size</td>
<td>ListBucketResult.Contents</td>
<td>Object size in bytes</td>
<td>integer</td>
</tr>
<tr>
<td>Owner</td>
<td>ListBucketResult.Contents</td>
<td>Bucket owner information</td>
<td>Container</td>
</tr>
<tr>
<td>StorageClass</td>
<td>ListBucketResult.Contents</td>
<td>Object storage class, such as <code>STANDARD_IA</code> and <code>ARCHIVE</code>. For enumerated values, see Storage Type</td>
<td>Enum</td>
</tr>
</tbody>
</table>

### Content of the Container node **Contents.Owner**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ListBucketResult.Contents.Owner</td>
<td>Bucket APPID</td>
<td>string</td>
</tr>
<tr>
<td>DisplayName</td>
<td>ListBucketResult.Contents.Owner</td>
<td>Object owner name</td>
<td>string</td>
</tr>
</tbody>
</table>

### Error Codes

There is no special error message for this API. For all error messages, see Error Codes.

### Examples

#### Example 1. Basic example

**Request**

```plaintext
GET / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 27 May 2019 11:26:14 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1558956374;1558963574&q-key-time=1558956374;1558963574&q-header-list=date;host&q-url-param-list=&q-signature=9a2596f2a4dc0f5eea901095a8275af4d****
Connection: close
```

**Response**

```plaintext
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1517
Connection: close
Date: Mon, 27 May 2019 11:26:15 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWNlYmM5NTdfZjI4NWQ2NF81ZmMwX2Q5N2E1****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
```
Example 2. Specifying the `delimiter` parameter (listing objects and subdirectories in the root directory)

Request

GET /?delimiter=%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 27 May 2019 11:26:15 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1558956375;1558963575&q-key-time=1558956375;1558963575&q-header-list=date;host&q-url-param-list=delimiter&q-signature=d409a7025e1633c16213ab34fbf7e87ef50****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1008
Connection: close
Date: Mon, 27 May 2019 11:26:15 GMT
Server: tencent-cos
Example 3. Specifying the `prefix` and `delimiter` parameters (listing objects and subdirectories in the specified subdirectory)

**Request**

```http
GET /?prefix=example-folder-1%2F&delimiter=%2F HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 27 May 2019 11:26:15 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1558956375;1558963575&q-key-time=1558956375;1558963575&q-header-list=date;host&q-url-param-list=delimiter;prefix&q-signature=f4a0854aa3a3b0b1699a19890675297867f3****
Connection: close
```

**Response**

```xml
<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
  <Name>examplebucket-1250000000</Name>
  <Prefix>example-folder-1/</Prefix>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <CommonPrefixes>
    <Prefix>example-folder-1/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>example-folder-2/</Prefix>
  </CommonPrefixes>
  <Contents>
    <Key>example-object-1.jpg</Key>
    <LastModified>2019-05-27T11:26:14.000Z</LastModified>
    <ETag>"0f0cd12c4b979d0b7f3f95255a36cb618quot;</ETag>
    <Size>20</Size>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>example-object-2.jpg</Key>
    <LastModified>2019-05-27T11:26:14.000Z</LastModified>
    <ETag>"51370fc64b79d0d3c7c609635be6418quot;</ETag>
    <Size>20</Size>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
```
Example 4. Getting the first list of entries when the total number of matching entries exceeds the maximum number of entries allowed for a single response

Request

GET / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 27 May 2019 11:07:30 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1558955250;1558962450&q-key-time=1558955250;1558962450&q-header-list=date;host&q-url-param-list=&q-signature=4d515ac9853b6e341cf5432c0d57faebe165****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 321247
Connection: close
Date: Mon, 27 May 2019 11:07:30 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWNlYmM0ZjJfZDcyNzVkNjRfNjQ5OV9lNzdk****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
  <Name>examplebucket-1250000000</Name>
  <Prefix/>
  <Marker/>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>true</IsTruncated>
  <NextMarker>example-object-1000.jpg</NextMarker>
  <Contents>
    <Key>example-object-0001.jpg</Key>
    <LastModified>2019-05-27T11:07:29.000Z</LastModified>
    <ETag>&quot;f49ffbb2dd542ef6843135d66ec97855&quot;</ETag>
    <Size>23</Size>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <Contents>
    <Key>example-object-0002.jpg</Key>
    <LastModified>2019-05-27T11:07:30.000Z</LastModified>
    <ETag>&quot;c9d2869897od6fe6fa1d0c459a7d358quot;&quot;</ETag>
    <Size>23</Size>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
Example 5. Getting the second list of entries when the total number of matching entries exceeds the maximum number of entries allowed for a single response

Request

GET /Marker=example-object-1000.jpg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 27 May 2019 11:08:36 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1558955316;1558962516&q-key-time=1558955316;1558962516&q-header-list=date;host&q-url-param-list=marker&q-signature=f9ed03c3944926d06281e6076331467890e****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1834
Connection: close
Date: Mon, 27 May 2019 11:08:36 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWNlYmM1MzRfZmVhODBiMDlfMmViNjRfZDQw****

<?xml version='1.0' encoding='utf-8' ?>
<ListBucketResult>
  <Name>examplebucket-1250000000</Name>
  <Prefix/>
  <Marker>example-object-1000.jpg</Marker>
  <MaxKeys>1000</MaxKeys>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>example-object-1001.jpg</Key>
    <LastModified>2019-05-27T11:07:29.000Z</LastModified>
    <ETag>&quot;b76c104f431e06bbfbbe7a97c87aecd8&quot;</ETag>
    <Size>23</Size>
    <Owner>
      <ID>1250000000</ID>
      <DisplayName>1250000000</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
</ListBucketResult>
HEAD Bucket

Description

This API is used to check whether a bucket exists and whether you have the permission to access it. Possible scenarios include:

- If the bucket exists and you have the permission to read it, HTTP status code 200 will be returned.
- If you do not have the permission to read the bucket, HTTP status code 403 will be returned.
- If the bucket does not exist, HTTP status code 404 will be returned.

Request

Sample Request

```
HEAD / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

Request Parameters

This API does not use any request parameter.

Request Headers

This API only uses common request headers. For more information on common request headers, see Common Request Headers.

Request Body

This API does not have a request body.

Response

Response Headers

In addition to common response headers, this API returns the following response headers. For more information, see Common Response Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-bucket-region</td>
<td>Bucket region such as ap-beijing, ap-hongkong, and eu-frankfurt. For the enumerated values, see Regions and Access Domain Names</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Response Body

The response body of this API is empty.

Error Codes

There is no special error message for this API. For all error messages, see Error Codes.

Example

Request

```
HEAD / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 28 May 2019 03:16:12 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0cSj0****&q-sign-time=1559013372;1559020572&q-key-time=1559013372;1559020572&q-HEAD-Bucket-LastUpdated=2019-12-16T20:04:28Z
```

©2013-2019 Tencent Cloud. All rights reserved.
Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: close
Date: Tue, 28 May 2019 03:16:12 GMT
Server: tencent-cos
x-cos-bucket-region: ap-beijing
x-cos-request-id: NWNlY2E3ZmNfZjhjMDBiMDlfMTBjOWRfZDcz****
DELETE Bucket

Description

This API is used to delete a specified bucket. To make this request, you need to have the permission to write to the bucket.

Before deleting a bucket, please make sure that all the data and incomplete multipart uploads in the bucket have been deleted; otherwise, the bucket cannot be deleted.

Request

Sample Request

DELETE / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).

Request Parameters

This API does not use any request parameter.

Request Headers

This API only uses common request headers. For more information on common request headers, see Common Request Headers.

Request Body

This API does not have a request body.

Response

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body

The response body of this API is empty.

Error Codes

The implementation of this operation returns the following special error messages. For all error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>BucketNotEmpty</td>
<td>The bucket is not empty</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>The specified bucket does not exist</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>

Example

Request

DELETE / HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
<table>
<thead>
<tr>
<th>Date</th>
<th>Tue, 28 May 2019 03:19:13 GMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>q-sign-algorithm=sha1&amp;q-ak=AKID8A0fBVtYFrNmB2oY1g1JQGF0c3J0****&amp;q-sign-time=1559013553;1559020753&amp;q-key-time=1559013553;1559020753&amp;q-header-list=date;host&amp;q-url-param-list=&amp;q-signature=478b1db6182db32c8ed450d6a723a0f500b2****</td>
</tr>
<tr>
<td>Connection</td>
<td>close</td>
</tr>
</tbody>
</table>

**Response**

```
HTTP/1.1 204 No Content
Content-Length: 0
Connection: close
Date: Tue, 28 May 2019 03:19:14 GMT
Server: tencent-cos
x-cos-request-id: NWNlY2E4YjFfNjljMDBiMDlfMmNiZTlfZGE0****
```
GET Bucket Object Versions

Description

The GET Bucket Object Versions API is used to retrieve all objects in a bucket and their historical version information. You can also filter certain objects and their version information by specifying relevant parameters. The requester needs to have read permissions to the storage bucket.

Request

Sample Request

GET /?versions HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String

Request Parameter

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix</td>
<td>Prefix match used to specify the prefix address of the file returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>delimiter</td>
<td>The delimiter is a symbol. If there is a prefix, the identical paths between</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>prefix and delimiter will be grouped into one class and defined as a</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>common prefix, and then all common prefixes will be listed. If there is no</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>prefix, the listing process will start from the beginning of the path</td>
<td></td>
<td></td>
</tr>
<tr>
<td>key-marker</td>
<td>By default, entries are listed in UTF-8 binary order starting from the</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>marker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>encoding-type</td>
<td>Specifies the encoding type of the return value. Valid value: url</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>max-keys</td>
<td>Maximum number of entries returned at a time; default and maximum value: 1,000</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>version-id-marker</td>
<td>Specifies that you need a list of all historical versions of an object</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>starting from the version ID &quot;version-id-marker&quot;. Valid values: Valid version ID, Default. If you do not specify any version ID, the latest version of the object will be listed by default</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information, see Common Request Headers.

Request Body

This API does not have a request body.

Response

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body
Below are the details:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListVersionsResult</td>
<td>None</td>
<td>Stores all the information on the results of the GetBucket request</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node ListVersionsResult:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>ListVersionsResult</td>
<td>Bucket name</td>
<td>string</td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>ListVersionsResult</td>
<td>Encoding type</td>
<td>string</td>
</tr>
<tr>
<td>Prefix</td>
<td>ListVersionsResult</td>
<td>Prefix match used to specify the prefix address of the file returned by the request response</td>
<td>string</td>
</tr>
<tr>
<td>KeyMarker</td>
<td>ListVersionsResult</td>
<td>By default, entries are listed in UTF-8 binary order starting from the marker</td>
<td>string</td>
</tr>
<tr>
<td>MaxKeys</td>
<td>ListVersionsResult</td>
<td>Maximum number of results returned in one response</td>
<td>string</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>ListVersionsResult</td>
<td>Whether response entries are truncated, which is a boolean value of true or false</td>
<td>boolean</td>
</tr>
<tr>
<td>NextMarker</td>
<td>ListVersionsResult</td>
<td>If the returned entries are truncated, then return NextMarker which marks the start of the next entry</td>
<td>string</td>
</tr>
</tbody>
</table>
## Cloud Object Storage

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteMarker</td>
<td>ListVersionsResult</td>
<td>Any object that has ever been deleted has a delete marker</td>
<td>Container</td>
</tr>
<tr>
<td>Version</td>
<td>ListVersionsResult</td>
<td>If an object is not deleted and exists in the bucket, this container records the object's metadata</td>
<td>Container</td>
</tr>
</tbody>
</table>

### Content of the Container node DeleteMarker:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>ListVersionsResult.DeleteMarker</td>
<td>Index of the deleted objects</td>
<td>string</td>
</tr>
<tr>
<td>VersionId</td>
<td>ListVersionsResult.DeleteMarker</td>
<td>Object version ID</td>
<td>string</td>
</tr>
<tr>
<td>IsLatest</td>
<td>ListVersionsResult.DeleteMarker</td>
<td>Indicates whether the deleted object is the latest version</td>
<td>string</td>
</tr>
<tr>
<td>LastModified</td>
<td>ListVersionsResult.DeleteMarker</td>
<td>Last modified time of the object</td>
<td>string</td>
</tr>
<tr>
<td>Owner</td>
<td>ListVersionsResult.DeleteMarker</td>
<td>Bucket owner information</td>
<td>Container</td>
</tr>
</tbody>
</table>

### Content of the Container node Version:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>ListVersionsResult.Version</td>
<td>Index of the deleted objects</td>
<td>string</td>
</tr>
<tr>
<td>VersionId</td>
<td>ListVersionsResult.Version</td>
<td>Object version ID</td>
<td>string</td>
</tr>
<tr>
<td>IsLatest</td>
<td>ListVersionsResult.Version</td>
<td>Indicates whether the deleted object is the latest version</td>
<td>string</td>
</tr>
<tr>
<td>LastModified</td>
<td>ListVersionsResult.Version</td>
<td>Last modified time of the object</td>
<td>string</td>
</tr>
<tr>
<td>ETag</td>
<td>ListVersionsResult.Version</td>
<td>An Entity Tag (ETag) is a hash value generated based on the object content rather than the metadata. Different objects have different ETags, so Etag can be used as an indicator of whether the specified object has been modified</td>
<td>string</td>
</tr>
<tr>
<td>Size</td>
<td>ListVersionsResult.Version</td>
<td>The size of the object in bytes</td>
<td>string</td>
</tr>
<tr>
<td>StorageClass</td>
<td>ListVersionsResult.Version</td>
<td>Storage class of the object; enumerators: STANDARD, STANDARD_IA, ARCHIVE</td>
<td>string</td>
</tr>
<tr>
<td>Owner</td>
<td>ListVersionsResult.Version</td>
<td>Bucket owner information</td>
<td>Container</td>
</tr>
</tbody>
</table>

### Content of the Container node Owner:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
</table>
Node Name (Keyword) | Parent Node | Description | Type
--- | --- | --- | ---
UID | ListVersionsResult.Contents.Owner | Bucket owner's APPID | string

**Samples**

**Request**

```plaintext
GET /?versions HTTP/1.1
Host: exampleBucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9Sm****&q-sign-time=1480932292;1981012292&q-key-time=1480932292;1981012292&q-url-param-list=versions&q-header-list=host&q-signature=5118a936b49f644482bb61369235cf4abf****
```

**Response**

```xml
<ListVersionsResult>
<Name>exampleBucket-1250000000</Name>
<Prefix/>
<KeyMarker/>
<VersionIdMarker/>
<MaxKeys>1000</MaxKeys>
<IsTruncated>false</IsTruncated>
<DeleteMarker>
<Key>100K.txt</Key>
<VersionId>MTg0NDUxODM2NDYxNTg1MTgxODk</VersionId>
<IsLatest>true</IsLatest>
<LastModified>2019-06-13T13:09:23.000Z</LastModified>
<Owner>
<UID>1250000000</UID>
</Owner>
</DeleteMarker>
<Version>
<Key>100K.txt</Key>
<VersionId>MTg0NDUxODM2NDYxNTg1MTgxODk</VersionId>
<IsLatest>false</IsLatest>
<LastModified>2019-06-13T12:05:51.000Z</LastModified>
<ETag>&quot;fffc7956ba9a7b58a63c01b6ce1ddc458quot;&quot;</ETag>
<Size>102401</Size>
<StorageClass>STANDARD</StorageClass>
<Owner>
<UID>1250000000</UID>
</Owner>
</Version>
<Version>
<Key>100M.txt</Key>
<VersionId>MTg0NDUxODM2NDYxNTg1MTgxODk</VersionId>
<IsLatest>false</IsLatest>
<LastModified>2019-06-13T10:00:09.000Z</LastModified>
<ETag>&quot;fffc7956ba9a7b58a63c01b6ce1ddc458quot;&quot;</ETag>
<Size>102401</Size>
<StorageClass>STANDARD</StorageClass>
<Owner>
<UID>1250000000</UID>
</Owner>
</Version>
<Version>
<Key>100K.txt</Key>
<VersionId>null</VersionId>
<IsLatest>false</IsLatest>
<LastModified>2019-06-13T10:00:09.000Z</LastModified>
<ETag>&quot;fffc7956ba9a7b58a63c01b6ce1ddc458quot;&quot;</ETag>
<Size>102401</Size>
<StorageClass>STANDARD</StorageClass>
<Owner>
<UID>1250000000</UID>
</Owner>
</Version>
```
Access Control List (acl)
PUT Bucket acl

Description

This API is used to write an access control list (ACL) to a bucket. You can pass in the ACL information through the request headers `x-cos-acl` and `x-cos-grant-` or through the request body in XML format.

- You can set the ACL information either through request headers or through the request body.
- PUT Bucket acl is an overwriting operation. The new ACL passed in will overwrite the old one.
- To make this request, you need to have the permission to write ACL to the bucket.

Request

Sample Request

Sample 1

```
PUT /?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: 0
Authorization: Auth String
```

Sample 2

```
PUT /?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: application/xml
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Request Body]
```

Authorization: Auth String (see Request Signature for details).

Request Parameters

This API does not use any request parameter.

Request Headers

In addition to common request headers, this API also supports the following request headers. For more information on common request headers, see Common Request Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-acl</td>
<td>Defines the access control list (ACL) attribute of the bucket. For the enumerated values such as <code>private</code> and <code>public-read</code>, see the “Preset ACL for buckets” section in ACL Overview. Default value: <code>private</code></td>
<td>Enum</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read</td>
<td>Allows grantee to read the bucket; format: `id=[OwnerUin]</td>
<td>id=[OwnerUin]<code>, such as </code>id=100000000001</td>
<td>id=100000000002<code>. You can use comma (,) to separate multiple users, such as </code>id=100000000001,id=100000000002`</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>x-cos-grant-write</td>
<td>Allows grantee to write to the bucket; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;.</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read-acp</td>
<td>Allows grantee to read the ACL and the policy of the bucket; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;.</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-write-acp</td>
<td>Allows grantee to write to the ACL and the policy of the bucket; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;.</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-full-control</td>
<td>Grants a user full permission to perform operations on the bucket; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;.</td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

**Request Body**

This request body submits the application/xml request data which include the bucket owner information and full authorization information.

```xml
<AccessControlPolicy>
  <Owner>
    <ID>string</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>string</URI>
      </Grantee>
      <Permission>Enum</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>string</ID>
      </Grantee>
      <Permission>Enum</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

The nodes are described in details below:

### Content of the Container node AccessControlPolicy:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>AccessControlPolicy</td>
<td>Bucket owner information</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>AccessControlList</td>
<td>AccessControlPolicy</td>
<td>Information on the grantees and permissions</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Content of the Container node Owner:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>ID</td>
<td>AccessControlPolicy.Owner</td>
<td>Complete ID of the bucket owner in the format of qcs::cam::uin/[OwnerUin]:uin/[OwnerUin], such as qcs::cam::uin/100000000001:uin/100000000001.</td>
<td>string</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Content of the Container node AccessControlList:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>AccessControlPolicy.AccessControlList</td>
<td>A single permission entry. One AccessControlList can have up to 100 Grant entries.</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Content of the Container node AccessControlList.Grant:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grantee</td>
<td>AccessControlPolicy.AccessControlList.Grant</td>
<td>Grantee information. xsi:type can be specified as Group or CanonicalUser. If it is specified as Group, the child node includes and can only include URI. If it is specified as CanonicalUser, the child node includes and can only include ID.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Permission</td>
<td>AccessControlPolicy.AccessControlList.Grant</td>
<td>Permissions. For the enumerated values such as WRITE and FULL_CONTROL, see the &quot;Actions on buckets&quot; section in ACL Overview.</td>
<td>Enum</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Content of the Container node AccessControlList.Grant.Grantee:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
</table>
Cloud Object Storage

Node Name (Keyword) | Parent Node | Description | Type | Required
--- | --- | --- | --- | ---
URI | AccessControlPolicy.AccessControlList.Grant.Grantee | Preset user group such as http://cam.qcloud.com/groups/global/AllUsers or http://cam.qcloud.com/groups/global/AuthenticatedUsers. For more information, see the “Preset user group” section in ACL Overview | string | Yes if xsi:type Grantee specified. Group
ID | AccessControlPolicy.AccessControlList.Grant.Grantee | Complete ID of the grantee in the format of qcs::cam::uin/[OwnerUin]:uin/[OwnerUin], such as qcs::cam::uin/100000000001:uin/100000000001 | string | Yes if xsi:type Grantee specified. Canonical

Response

Response Headers
This API only returns common response headers. For more information, see Common Response Headers.

Response Body
The response body of this API is empty.

Error Codes
The implementation of this operation returns the following special error messages. For all error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidDigest</td>
<td>The given Content-MD5 checksum is invalid</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>The XML format of the request body does not conform to the XML syntax</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>

Examples

Example 1. Configuring ACL through request headers

Request

```
PUT /?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:30:12 GMT
x-cos-acl: public-read
x-cos-grant-write: id="100000000002"
x-cos-grant-read-acp: id="100000000002"
Content-Length: 0
Authorization: q-sign-algorithm=sha1&q-ak=AKIDBA0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1560760212;1560767412&q-key-time=1560760212;1560767412&q-header-list=content-length;date;host;x-cos-acl;x-cos-grant-read-acp;x-cos-grant-write&q-url-param-list=acl&q-signature=5b10c6ea4e6c9630c885e1f85476c768b5&****
Connection: close
```

Response

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Mon, 17 Jun 2019 08:30:13 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzRmOTRfODhjMjJhMDlfMWRlYl81Mzc0****
```

Example 2. Configuring ACL through request body

Request
PUT /?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:30:13 GMT
Content-Type: application/xml
Content-Length: 812
Content-MD5: 1qS+8SqnivarcO6Z11R0nw==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JJQF0c3J0****&q-sign-time=1560760213;1560767413&q-key-time=1560760213;1560767413&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=acl&q-signature=70f96b91823f3715905df125d96fe447554****
Connection: close

<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
      </Grantee>
      <Permission>READ_ACP</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>

Response

HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Mon, 17 Jun 2019 08:30:13 GMT
Server: tencent-cos
x-cos-request-id: NMQ0wZxRnOTV1zBjMDJhMDfOTW3MF8yNzdj****
GET Bucket acl

Description
This API is used to get the access control list (ACL) of a bucket. To make this request, you need to have the permission to read the ACL of the bucket.

Request

Sample Request

GET /acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for details).

Request Parameters
This API does not use any request parameter.

Request Headers
This API only uses common request headers. For more information on common request headers, see Common Request Headers.

Request Body
This API does not have a request body.

Response

Response Headers
This API only returns common response headers. For more information, see Common Response Headers.

Response Body
A successful query will return application/xml data which include the bucket owner information and full authorization information.

```
<AccessControlPolicy>
  <Owner>
    <ID>string</ID>
    <DisplayName>string</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>string</URI>
      </Grantee>
      <Permission>Enum</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>string</ID>
        <DisplayName>string</DisplayName>
      </Grantee>
      <Permission>Enum</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

The nodes are described in details below:
<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlPolicy</td>
<td>None</td>
<td>Stores the result of the GET Bucket acl operation</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node** `AccessControlPolicy`:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>AccessControlPolicy</td>
<td>Bucket owner information</td>
<td>Container</td>
</tr>
<tr>
<td>AccessControlList</td>
<td>AccessControlPolicy</td>
<td>Information on the grantee and permissions</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node** `Owner`:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>AccessControlPolicy.Owner</td>
<td>Complete ID of the bucket owner in the format of <code>qcs::cam::uin/[OwnerUin]:uin/[OwnerUin]</code>, such as <code>qcs::cam::uin/100000000001:uin/100000000001</code></td>
<td>string</td>
</tr>
<tr>
<td>DisplayName</td>
<td>AccessControlPolicy.Owner</td>
<td>Bucket owner name</td>
<td>string</td>
</tr>
</tbody>
</table>

**Content of the Container node** `AccessControlList`:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>AccessControlPolicy.AccessControlList</td>
<td>A single permission entry</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node** `AccessControlList.Grant`:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grantee</td>
<td>AccessControlPolicy.AccessControlList.Grant</td>
<td>Grantee information. If <code>xsi:type</code> is specified as <code>Group</code>, the child node includes and only includes URI. If it is specified as <code>CanonicalUser</code>, the child nodes include and only include ID and <code>DisplayName</code></td>
<td>Container</td>
</tr>
<tr>
<td>Permission</td>
<td>AccessControlPolicy.AccessControlList.Grant</td>
<td>Permissions. For the enumerated values such as <code>WRITE</code> and <code>FULL_CONTROL</code>, see the “Actions on buckets” section in ACL Overview</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**Content of the Container node** `AccessControlList.Grant.Grantee`:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Preset user group such as <code>http://cam.qcloud.com/groups/global/ALUsers</code> or <code>http://cam.qcloud.com/groups/global/AuthenticatedUsers</code>. For more information, see the “Preset user group” section in ACL Overview</td>
<td>string</td>
</tr>
</tbody>
</table>
### Node Name (Keyword)

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Complete ID of the grantee in the format qcs::cam::uin/[OwnerUin]:uin/[OwnerUin], such as qcs::cam::uin/100000000001:uin/100000000001</td>
<td>string</td>
</tr>
<tr>
<td>DisplayName</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Grantee name</td>
<td>string</td>
</tr>
</tbody>
</table>

### Error Codes

There is no special error message for this API. For all error messages, see Error Codes.

### Example

#### Request

GET /acl HTTP/1.1  
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com  
Date: Mon, 17 Jun 2019 08:37:35 GMT  
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1560760655;1560767855&q-key-time=1560760655;1560767855&q-header-list=date;host&q-url-param-list=acl&q-signature=24b9d377eac80917a33c8c29842ce5b1a5****  
Connection: close

#### Response

HTTP/1.1 200 OK  
Content-Type: application/xml  
Content-Length: 1035  
Connection: close  
Date: Mon, 17 Jun 2019 08:37:36 GMT  
Server: tencent-cos  
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****

```xml
<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
    <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
        <Permission>READ</Permission>
      </Grant>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
          <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
          <DisplayName>qcs::cam::uin/100000000002:uin/100000000002</DisplayName>
          <Permission>WRITE</Permission>
        </Grantee>
      </Grant>
      <Grant>
        <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
          <ID>qcs::cam::uin/100000000002:uin/100000000002</ID>
          <DisplayName>qcs::cam::uin/100000000002:uin/100000000002</DisplayName>
          <Permission>READ_ACP</Permission>
        </Grantee>
      </Grant>
    </AccessControlList>
</AccessControlPolicy>
```
Cross-Origin Resource Sharing (cors)
PUT Bucket cors

Description

API PUT Bucket cors is used to request the cross-domain resource sharing of Bucket, Permission. You can implement the configuration by passing a configuration file in XML format. The file size is limited to 64KB. By default, the Bucket holder directly has Permission to use the API, and the Bucket holder can also grant Permission to other users.

Request

Request Sample

```
PUT /?cors HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: length
Content-Type: application/xml
Content-MD5: MD5
Authorization: Auth String
```

Note:

For more information on Authorization: Auth String, please see Request Signature Documentation.

Request header

In addition to using common request headers, this API also supports the following required request headers. For more information on common request headers, please see Common Request Headers Document.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>The Base64-encoded MD5 hash value of the request body content defined in RFC 1864, which is used for integrity checking to verify whether the request body has changed during transmission.</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request body

The request body of the request is a cross-domain rule.

```
<?xml version="1.0" encoding="UTF-8" ?>
<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>
```

Below are the details:

<table>
<thead>
<tr>
<th>Node name (keyword)</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node name (keyword)</td>
<td>Parent node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>CORSConfiguration</td>
<td>None</td>
<td>Describes all the information about cross-domain resource sharing configuration, which can contain up to 100 CORSRule</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of Container node CORSConfiguration:

<table>
<thead>
<tr>
<th>Node name (keyword)</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORSRule</td>
<td>CORSConfiguration</td>
<td>Describes all the information about cross-domain resource sharing configuration, which can contain up to 100 CORSRule</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of Container node CORSRule:

<table>
<thead>
<tr>
<th>Node name (keyword)</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>CORSConfiguration.CORSRule</td>
<td>Configure rule ID, but Optional</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>AllowedOrigin</td>
<td>CORSConfiguration.CORSRule</td>
<td>Allowed origin in the format of protocol://domain name[:port number], such as <a href="http://www.qq.com">http://www.qq.com</a>. Wildcard * is supported</td>
<td>Strings</td>
<td>Yes</td>
</tr>
<tr>
<td>AllowedMethod</td>
<td>CORSConfiguration.CORSRule</td>
<td>Allowed HTTP operations. Enumerated values: GET,PUT,HEAD,POST,DELETE</td>
<td>Strings</td>
<td>Yes</td>
</tr>
<tr>
<td>AllowedHeader</td>
<td>CORSConfiguration.CORSRule</td>
<td>When sending a OPTIONS request, tell the server which custom HTTP request headers can be used for subsequent requests, and wildcards are supported. *</td>
<td>Strings</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxAgeSeconds</td>
<td>CORSConfiguration.CORSRule</td>
<td>Set the validity period of the result obtained by the OPTIONS request</td>
<td>Integer</td>
<td>Yes</td>
</tr>
<tr>
<td>ExposeHeader</td>
<td>CORSConfiguration.CORSRule</td>
<td>Set the custom header information that the browser can receive from the server</td>
<td>Strings</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Response

**Response header**
This API only returns the public response header. For more information, please see Common Response Header Document.

Response body
The response body is empty.

Error Codes

<table>
<thead>
<tr>
<th>Error Codes</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SignatureDoesNotMatch</td>
<td>The signature provided does not conform to the rules. The error code is returned.</td>
<td>four hundred and three Forbidden</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>If the Bucket of the rule you are trying to add does not exist, return the error code</td>
<td>four hundred and four Not Found</td>
</tr>
</tbody>
</table>

Actual case

Request

```
PUT /?cors HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Content-MD5: q+xJ56ypmuO9k0hj0Zi0==
Content-Type: application/xml
Authorization: q-sign-algorithm=sha1&q-ak=AKIDVMyLTL4B8rVt52LTozzPZBYffPs9****&q-sign-time=1578385303;1578392503&q-key-time=1578385303;1578392503&q-header-list=content-md5;content-type;host&q-url-param-list=cors&q-signature=730a82c7afed2a6e051878d54895193235e8****
Content-Length: 385

<?xml version="1.0" encoding="UTF-8" ?>
<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>
```

Response

```
HTTP/1.1 200 OK
content-length: 0
connection: close
date: Tue, 07 Jan 2020 08:21:44 GMT
server: tencent-cos
x-cos-request-id: NWUxNDNmOThfNWFiMjU4NjRfMWIxYl9lYWY1****
```
GET Bucket cors

Last updated: 2019-12-16 16:17:43

Description

This API allows the bucket owner to obtain the cross-origin resource sharing (CORS, a W3C standard) configuration of a bucket. By default, the bucket owner has the permission to use this API and can grant such permission to other users.

Request

Sample Request

```plaintext
GET /?cors HTTP/1.1
Host: <Bucketname-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for details).
```

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response contains common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This response does not use any special response header.

Response Body

CORS configuration information is obtained successfully.
The data are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORSConfiguration</td>
<td>None</td>
<td>describes all the information on the CORS configuration, which can contain up to 100 CORSRules</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node CORSConfiguration:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORSRule</td>
<td>CORSConfiguration</td>
<td>describes all the information on the CORS configuration, which can contain up to 100 CORSRules</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node CORSRule:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>CORSConfiguration.CORSRule</td>
<td>ID of the configuration rule; optional</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>AllowedOrigin</td>
<td>CORSConfiguration.CORSRule</td>
<td>Allowed origin in the format of protocol://domain name[:port number], such as <a href="http://www.qq.com">http://www.qq.com</a>. Wildcard * is supported</td>
<td>Strings</td>
<td>Yes</td>
</tr>
<tr>
<td>AllowedMethod</td>
<td>CORSConfiguration.CORSRule</td>
<td>Allowed HTTP operations. Enumerated values: GET, PUT, HEAD, POST, DELETE</td>
<td>Strings</td>
<td>Yes</td>
</tr>
<tr>
<td>AllowedHeader</td>
<td>CORSConfiguration.CORSRule</td>
<td>Tells the server side when sending the OPTIONS request what user-defined HTTP request headers can be used for subsequent requests. Wildcard * is supported</td>
<td>Strings</td>
<td>Yes</td>
</tr>
<tr>
<td>MaxAgeSeconds</td>
<td>CORSConfiguration.CORSRule</td>
<td>Sets the validity period of the result of the OPTIONS request</td>
<td>Integer</td>
<td>Yes</td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>ExposeHeader</td>
<td>CORSConfiguration.CORSRule</td>
<td>Sets the user-defined header information from the server side that can be received by the browser</td>
<td>strings</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Error Codes**

The implementation of this operation does not return special error messages. For common error messages, see Error Codes.

**Example**

**Request**

```
GET /?cors HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 28 Oct 2016 21:32:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDMtcYfJf50wLB9Aw1062ThfSUyfGF0b&q-sign-time=1484815944;325577119448&q-key-time=1484815944;325577119448&q-header-list=host&q-url-param-list=cors&q-signature=a2d28e1b9023d09f9277982775a4b3b705d0e23e
```

**Response**

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 345
Connection: keep-alive
Date: Wed, 28 Oct 2016 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg4MDdlNGZfNDYyMDRlXzM0YWFfZTBh

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

Description

This API is used to delete cross-origin access configuration information.

Request

Sample Request

```
DELETE /?cors HTTP/1.1
Host: <Bucketname-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This response does not use any special response header.

Response Body

This response body is empty.

Error Codes

The implementation of this operation returns the following error messages. For common error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Deleted successfully. The response body is empty.</td>
<td>204 No Content</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>This error code will be returned if the bucket you are trying to access does not exist</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>

Example
Request

DELETE /?cors HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 12:59:09 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOa07cPwF9cXFrJD0a11CvR0BJM&q-sign-time=1502859472;1502939472&q-key-time=1502859472;1502939472&q-header-list=host&q-url-param-list=lifecycle&q-signature=49c1414c700643f11139210384332a3ec4e5485b

Response

HTTP/1.1 204 No Content
Content-Type: application/xml
Date: Wed, 16 Aug 2017 12:59:09 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDQxOWNMj04OGY3Zz0NGRfMjE=
Lifecycle (lifecycle)
PUT Bucket lifecycle

Description

COS allows you to manage the lifecycle of objects in buckets through lifecycle configuration that contains one or more rule sets that will be applied to a set of objects. Each rule defines an operation for COS. There are two types of operations:

- **Transition**: Defines the time when an object is transitioned to another storage class. For example, 30 days after the creation of an object, you can choose to transition it to the `STANDARD_IA` storage class which is suitable for objects that are not accessed frequently. You can also choose to transition it to the `ARCHIVE` storage class for even lower costs which is available in regions within Mainland China. For specific parameters, see `Transition` in the sample request description.

- **Expiration**: Specifies the expiration time of an object. COS will automatically delete expired objects.

Notes

This API is used to create a new lifecycle configuration for a bucket. If a lifecycle configuration has already been made for the bucket, the new configuration created with this API will overwrite the existing one.

Request

Sample Request

```
PUT /?lifecycle HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Content-Length: length
Date: GMT Date
Authorization: Auth String
Content-MD5: MD5
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers
The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers
Required headers
The implementation of this request operation uses the following required request headers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>Base64-encoded 128-bit MD5 checksum as defined in RFC 1864. This header is used to verify whether the file content has changed</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Body

The node content of this request body is:

```
<LifecycleConfiguration>
  <Rule>
    <ID></ID>
  <Filter>
```
The content is described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>LifecycleConfiguration</td>
<td>None</td>
<td>Lifecycle configuration</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Rule</td>
<td>LifecycleConfiguration</td>
<td>Rule description</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Filter</td>
<td>LifecycleConfiguration.Rule</td>
<td>Describes the set of objects subject to a rule</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Status</td>
<td>LifecycleConfiguration.Rule</td>
<td>Specifies whether a rule is enabled. Enumerated values: Enabled, Disabled</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>LifecycleConfiguration.Rule</td>
<td>Uniquely identifies a rule; up to 255 characters in length</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>And</td>
<td>LifecycleConfiguration.Rule.Filter</td>
<td>Used to connect Prefix</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Prefix</td>
<td>LifecycleConfiguration.Rule.Filter</td>
<td>Specifies the prefix to which a rule applies. Objects that match the prefix are subject to the rule. You can specify at most one Prefix</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Expiration</td>
<td>LifecycleConfiguration.Rule</td>
<td>Expiration attribute of a rule</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Transition</td>
<td>LifecycleConfiguration.Rule</td>
<td>Transition attribute of a rule, specifying when an object is transitioned to Standard_IA or ARCHIVE storage class</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Days</td>
<td>LifecycleConfiguration.Rule.Transition or Expiration</td>
<td>Specifies the number of days between the last modified date of an object and the date when the operation corresponding to a rule is performed. If the operation is Transition, a valid value of this field should be a non-negative integer; if it is Expiration, a valid value of this field should be a positive integer. Maximum: 3,650 days</td>
<td>Integer</td>
<td>No</td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Date</td>
<td>LifecycleConfiguration.Rule.Transition or Expiration</td>
<td>Specifies when the operation corresponding to a rule is performed</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ExpiredObjectDeleteMarker</td>
<td>LifecycleConfiguration.Rule.Expiration</td>
<td>Whether to delete the delete marker of an expired object. Enumerated values: true, false</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>AbortIncompleteMultipartUpload</td>
<td>LifecycleConfiguration.Rule</td>
<td>Sets the maximum amount of time allowed for a multipart upload to keep running</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>DaysAfterInitiation</td>
<td>LifecycleConfiguration.Rule .AbortIncompleteMultipartUpload</td>
<td>Specifies in how many days a multipart upload must be completed once started</td>
<td>Integer</td>
<td>Yes</td>
</tr>
<tr>
<td>NoncurrentVersionExpiration</td>
<td>LifecycleConfiguration.Rule</td>
<td>Specifies when non-current versions of an object expire</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>NoncurrentVersionTransition</td>
<td>LifecycleConfiguration.Rule</td>
<td>Specifies when non-current versions of an object are transitioned to STANDARD_IA or ARCHIVE storage class</td>
<td>Container</td>
<td>No</td>
</tr>
</tbody>
</table>
### Node Name (Keyword) | Parent Node | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NoncurrentDays</td>
<td>LifecycleConfiguration.Rule, .NoncurrentVersionExpiration, or NoncurrentVersionTransition</td>
<td>Specifies the number of days between the date when an object becomes non-current and the date when the operation corresponding to a rule is performed. If the operation is Transition, a valid value of this field should be a non-negative integer; if it is Expiration, a valid value of this field should be a positive integer. Maximum: 3,650 days</td>
</tr>
</tbody>
</table>

| StorageClass         | LifecycleConfiguration.Rule.Transition or NoncurrentVersionTransition | Specifies to which storage class an object is transitioned. Enumerated values: STANDARD_IA, ARCHIVE |

### Response

#### Response Headers

**Common Response Headers**

This response uses common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**

This response does not use any special response header.

#### Response Body

The response body of this response is empty.

#### Error Codes

The following describes some frequent special errors that may occur when you make this request. For specific reasons of errors, see the returned error messages. For more COS error codes or a complete list of errors, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchBucket</td>
<td>404 Not Found</td>
<td>The bucket you are trying to access does not exist</td>
</tr>
<tr>
<td>Error Code</td>
<td>HTTP Status Code</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>400 Bad Request</td>
<td>Invalid XML format. Please check against the RESTful API documentation</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>400 Bad Request</td>
<td>Invalid request. If the error message shows “Conflict lifecycle rule”, it means that multiple rules in the XML data conflict with each other.</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>400 Bad Request</td>
<td>Invalid request parameter. If the error message shows “Rule ID must be unique. Found same ID for more than one rule”, it means that multiple rules have the same ID field.</td>
</tr>
</tbody>
</table>

**Example**

**Request**

```plaintext
PUT /lifecycle HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 11:59:33 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0Ao7cllgPv9cXFJD8a11CvR98JMN&q-sign-time=1502855771;1502935771&q-key-time=1502855771;1502935771&q-header-list=content-md5;host&q-url-param-list=lifecycle&q-signature=766fca6fc056d03680685d5e56973ccfc12b4654
Content-MD5: LcNUuow8OSZMrEDnvndw1Q==
Content-Length: 348
Content-Type: application/x-www-form-urlencoded

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>100</Days>
      <StorageClass>ARCHIVE</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>10</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

**Response**

```plaintext
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Wed, 16 Aug 2017 11:59:33 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDMzYTRfMjQ4OGY3Xzc3NGRfMY=
```
GET Bucket lifecycle

Last updated: 2019-12-16 16:31:43

Description

This API is used to query the lifecycle configuration of a bucket. If the bucket does not have a lifecycle rule configured, `NoSuchLifecycleConfiguration` will be returned.

Request

Sample Request

```
GET /?lifecycle HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response contains common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This response does not use any special response header.

Response Body

The content and meaning of elements in this response body are the same as those in the request body for PUT Bucket lifecycle. For more information, see the request body node description section in PUT Bucket lifecycle.

Error Codes

The implementation of this operation does not return special error messages. For common error messages, see Error Codes.

Example

Request
GET /?lifecycle HTTP/1.1
Host: examplebucket-125000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 12:23:54 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7c1gPv9cXFrJ0a1ICvR98JMt&q-sign-time=1502857357;1502937357&q-key-time=1502857357;1502937357&q-header-list=host&q-url-param-list=lifecycle&q-signature=da155cda3461bee7422ee0536facc03f947e92

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 312
Date: Wed, 16 Aug 2017 12:23:54 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDM5NWFfMjQ4OGY3Xzc3NGRfMjA=

<LifecycleConfiguration>
  <Rule>
    <ID>id1</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>100</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>id2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>10</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
DELETE Bucket lifecycle

Description

This API is used to delete the lifecycle configuration of a bucket. If the bucket does not have a lifecycle rule configured, `NoSuchLifecycleConfiguration` will be returned.

Request

Sample Request

```
DELETE /?lifecycle HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

```
Authorization: Auth String (see Request Signature for more information).
```

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response contains common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This response does not use any special response header.

Response Body

This response body is empty.

Error Codes

The implementation of this operation returns the following error messages. For common error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Deleted successfully. The response body is empty.</td>
<td>204 No Content</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>This error code will be returned if the bucket you are trying to access does not exist</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>
## Example

### Request

```
DELETE /?lifecycle HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 16 Aug 2017 12:59:09 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrdB0a1ICvR9B&u-sign-time=1502859472;1502939472&q-key-time=1502859472;1502939472&u-sregunta-signature=49c1414c700643f11139219384332a3ec4e9485b
```

### Response

```
HTTP /1.1 204 No Content
Content-Type: application/xml
Date: Wed, 16 Aug 2017 12:59:09 GMT
Server: tencent-cos
x-cos-request-id: NTk5NDQxOWNfMjQ4OGY3Xzc3NGRfMjE=
```
Bucket policy(policy)
PUT Bucket policy

Description

This API (PUT Bucket policy) is used to write a permission policy for a Bucket. The policy passed in will overwrite the existing permission policy.

Request

Request example

```plaintext
PUT /?policy HTTP/1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: date
Content-Type:application/json
Content-MD5:MD5
Authorization: Auth String

Authorization: Auth String (For more information, see Request Signature).
```

Request headers

**Common headers**

Common request headers are used for the implementation of this request operation. For more information, see Common Request Headers.

Request parameters

No special request parameter is needed.

Request body

```json
{
  "Statement": [
    {
      "Principal": {
        "qcs": [
          "qcs::cam::uin/${owner_uin}:uin/${sub_uin}"
        ],
        "Effect": "$\{effect\}",
        "Action": [
          "name/cos:${action}"
        ],
        "Resource": [
          "qcs::cos:${region}:uid/${appid}:${bucket}/*"
        ]
      }
    ]
  },
  "version": "2.0"
}
```

Response

Response headers

**Common response headers**

This response uses common response header. For more information, see Common Response Headers.
Special response headers
No special response header is used for this request operation.

Response body
The response body is empty.

Error codes
No special error code is returned. For common error codes, see Error Codes.

Example

Request

```
PUT /?policy HTTP/1.1
Host:examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUJfGF0&q-sign-time=1484813288;32557709288&q-key-time=1484813288;32557709288&q
-host-list=host&q-url-param-list=policy&q-signature=05f7fc093636f910a94a0c815e1f1752f034d47a
Content-Type: application/json
Content-Length: 233

}
```

Response

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu Jan 19 16:19:22 2017
Server: tencent-cos
x-cos-request-id: NTg4MDc2OGFfNDUyMDElXzc3NTIzZTc4
```
GET Bucket policy

Description

This API (GET Bucket policy) is used to read the permission policy of a Bucket.

Request

Request example:

GET /?policy HTTP/1.1
Host:<bucketname-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String

Authorization: Auth String (For more information, see Request Signature).

Request headers

Common headers

Common request headers are used for the implementation of this request operation. For more information, see Common Request Headers.

Special request headers

This request operation does not use any special request header.

Request body

The request body of this request is empty.

Response

Response headers

Common headers

This response uses common response headers. For more information, see Common Response Headers.

Special response headers

This response does not use any special response header.

Response body

The permission policy is returned in the response body.

```
{
  "Statement": [ 
    {
      "Principal": { 
        "qcs": [ 
          "qcs::cam::uin/${owner_uin}:uin/${sub_uin}" 
        ],
        "Effect": "${effect}",
        "Action": [ 
          "name/cos:${action}" 
        ],
        "Resource": [ 
          "qcs::cos:${region}:uid/${appid}:${bucket}/*
```
Error codes

No special error message is returned for this request operation. For common error messages, see Error Codes.

Example

Request

GET /?policy HTTP/1.1
Host:examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb27h3SUj6F0B&q-sign-time=1484814099;32557710099&q-key-time=1484814099;32557710099&q-header-list=host&q-url-param-list=policy&q-signature=0523d7c6305b6676861c447986c48b559e68869

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 237
Connection: keep-alive
Date: Thu Jan 19 16:21:46 2017
Server: tencent-cos
x-cos-request-id: NTg4MDc3MWFfOWIxZjRlXzZmNDVfZTBl

} } "version": "2.0" "version": "2.0"
DELETE Bucket policy

Last updated : 2019-06-14 18:49:49

Description

This API (DELETE Bucket policy) is used to delete the permission policy of a Bucket.

Only the Bucket owner is allowed to initiate this request. You will receive “204 No Content” if the permission policy does not exist.

Request

Request example

DELETE /?policy HTTP/1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date: date
Authorization: Auth String

Authorization: Auth String (For more information, see Request Signature).

Request headers

Common headers

Common request headers are used for the implementation of this request operation. For more information, see Common Request Headers.

Special request headers

This request operation does not use any special request header.

Request body

The request body of this request is empty.

Response

Response headers

Common response headers

This response uses common response headers. For more information, see Common Response Headers.

Special response headers

This response does not use any special response header.

Response body

The response body is empty.

Example

Request

DELETE /?policy HTTP/1.1
Host:examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfgFO&q-sign-time=1484814613;32557710613&q-key-time=1484814613;32557710613&q
-url-param-list=policy&q-signature=57c9a3f67b786d0b2c208641944ec7f9b02f98

©2013-2019 Tencent Cloud. All rights reserved.
### Response

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu Jan 19 16:30:21 2017
Server: tencent-cos
x-cos-request-id: NTg4MDc5MWYyNDQyMDRlKzNiMDRfZTEw
```
Hotlink protection (referer)
PUT Bucket referer

Description

This API (GET Bucket Referer) is used to read the whitelist or the blacklist of Bucket Referer.

Request

Request Sample

GET /?referer HTTP 1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Note:
Authorization:Auth String (see Request Signature Documentation).

Request header

This API only uses common request headers. For more information, please see Common Request Headers Document.

Request body

The request body of this request is empty.

Response

Response header

This API only returns the public response header. For more information, please see Common Response Header Document.

Response body

Application/xml data is returned in the response body. It contains the complete node data, as shown below:

```
<RefererConfiguration>
  <Status>Enabled</Status>
  <RefererType>White-List</RefererType>
  <DomainList>
    <Domain>*.qq.com</Domain>
    <Domain>*.qcloud.com</Domain>
  </DomainList>
  <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
</RefererConfiguration>
```

Please find the details below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefererConfiguration</td>
<td>None</td>
<td>Hotlink protection Prevent hotlinking configuration Information</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Parent node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>Status</td>
<td>RefererConfiguration</td>
<td>Whether to enable Hotlink protection. Prevent hotlinking. Enumerated values: Enabled, Disabled</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>RefererType</td>
<td>RefererConfiguration</td>
<td>Hotlink protection. Prevent hotlinking type. Enumerated value: Black-List, White-List</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>DomainList</td>
<td>RefererConfiguration</td>
<td>List of valid domain names. Multiple domain names with prefixes are supported, domain names with ports are supported, and IP supports wildcards. Wildcard, who is a second-level domain name or a multi-level domain name</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Domain</td>
<td>DomainList</td>
<td>A single effective domain name, such as <a href="http://www.qq.com/example">www.qq.com/example</a>, 192.168.1.2:8080, *.qq.com</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>EmptyReferConfiguration</td>
<td>RefererConfiguration</td>
<td>Whether to allow empty Referer Access. Enumerated values: Allow, Deny, default is Deny.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**Error Codes**

There is no special error message for this request operation. For all error messages, see Error Codes Document.

**Actual case**

**Request**

```
GET /?referer HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 25 Feb 2017 04:10:22 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOaO7cllgPvF9cXFrJ09a11ClvR****&q-sign-time=1547105134;3252698134&q-key-time=1547105134;32620001134&q-header-list=content-md5;content-type;host&q-url-param-list=referer&q-signature=0f7fef5b1d2188deaf6f92fa2ee0cf87ae83****
```

**Response**

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 260
Connection: keep-alive
Date: Fri, 25 Feb 2017 04:10:22 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjFJMsjFWXZjRkXZwNDHf****
```
```xml
<RefererConfiguration>
  <Status>Enabled</Status>
  <RefererType>White-List</RefererType>
  <DomainList>
    <Domain>*.qq.com</Domain>
    <Domain>*.qcloud.com</Domain>
  </DomainList>
  <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
</RefererConfiguration>
```

**GET Bucket Referer**
Description

This API (GET Bucket Referer) is used to read the whitelist or the blacklist of Bucket Referer.

Request

Request example

GET /?referer HTTP 1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (For more information, see Request Signature).

Request headers

Common headers

Common request headers are used for the implementation of this request operation. For more information, see Common Request Headers.

Special request headers

This request operation does not use any special request header.

Request body

The request body of this request is empty.

Response

Response headers

Common response headers

This response uses common response headers. For more information, see Common Response Headers.

Special response headers

This response does not use any special response header.

Response body

Application/xml data is returned in the response body. It contains the complete node data, as shown below:

```
<RefererConfiguration>
  <Status></Status>
  <RefererType></RefererType>
  <DomainList>
    <Domain></Domain>
    <Domain></Domain>
  </DomainList>
  <EmptyReferConfiguration></EmptyReferConfiguration>
</RefererConfiguration>
```

Detailed data is shown as below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RefererConfiguration</td>
<td>None</td>
<td>Hotlink protection configuration</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Status</td>
<td>RefererConfiguration</td>
<td>Indicates whether to enable hotlink protection. Enumerated value: Enabled and Disabled</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>RefererType</td>
<td>RefererConfiguration</td>
<td>Hotlink protection type. Enumerated value: Black-List and White-List</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>DomainList</td>
<td>RefererConfiguration</td>
<td>Sets a list of domain names used in the blacklist/whitelist. You can configure multiple domain names by using prefix match. Domain names and IPs with ports are supported. Wildcard &quot;*&quot; is supported for second-level and third-level domain names.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Domain</td>
<td>DomainList</td>
<td>Sets a single domain name used in the blacklist/whitelist, such as <a href="http://www.qq.com/example">www.qq.com/example</a>, 192.168.1.2:8080, and *.qq.com.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>EmptyReferConfiguration</td>
<td>RefererConfiguration</td>
<td>Indicates whether an access request without Referer is allowed. Enumerated value: Allow and Deny. It defaults to Deny.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**Error codes**

No special error message is returned for this request operation. For common error messages, see Error Codes.

**Example**

**Request**

```
GET /?referer HTTP 1.1  
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com  
Date: Fri, 25 Feb 2017 04:10:22 GMT  
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0A0c7c1gPwFoQFt0JdD08a11GvR0BJM&q-sign-time=1547105134;3252689134&q-key-time=1547105134;32620001134&q-header-list=content-md5;content-type;host&q-url-param-list=referer&q-signature=0f7fef5b1d2188deadf6f92fa2ee0cf87ae83f0cd
```

**Response**

```
HTTP/1.1 200 OK  
Content-Type: application/xml  
Content-Length: 260  
Connection: keep-alive
```
<RefererConfiguration>
  <Status>Enabled</Status>
  <RefererType>White-List</RefererType>
  <DomainList>
    <Domain>*.qq.com</Domain>
    <Domain>*.qcloud.com</Domain>
  </DomainList>
  <EmptyReferConfiguration>Allow</EmptyReferConfiguration>
</RefererConfiguration>
Tag (tagging)
PUT Bucket tagging

Last updated : 2019-12-16 17:34:19

Description

COS supports setting tags for existing buckets. This API is used to set key-value pairs for a bucket as tags, helping you manage existing bucket resources and costs.

Currently, up to 30 different tags can be set for one bucket.

Request

Sample Request

PUT /?tagging HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

For this request, you need to configure the following set of tags:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<Tagging>
  <TagSet>
    <Tag>
      <Key>age</Key>
      <Value>18</Value>
    </Tag>
    <Tag>
      <Key>name</Key>
      <Value>xiaoming</Value>
    </Tag>
  </TagSet>
</Tagging>
```

The data are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagging</td>
<td>None</td>
<td>Tag set</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>TagSet</td>
<td>Tagging</td>
<td>Tag set</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>Tagging.TagSet</td>
<td>Tag set, which can contain up to 10 tags</td>
<td>Containers</td>
<td>Yes</td>
</tr>
<tr>
<td>Key</td>
<td>Tagging.TagSet.Tag</td>
<td>Tag key, which can contain up to 128 characters. A tag key can contain English letters, digits, spaces, plus signs, minus signs, underscores, equals signs, dots, colons, and slashes</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Value</td>
<td>Tagging.TagSet.Tag</td>
<td>Tag value, which can contain up to 256 characters. A tag value can contain English letters, digits, spaces, plus signs, minus signs, underscores, equals signs, dots, colons, and slashes</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Response

Response Headers

Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This request operation does not use any special response header.

Response Body

The response body of this request is empty.

Error Codes

The following describes some frequent special errors that may occur when you make this request:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SignatureDoesNotMatch</td>
<td>If the provided signature does not conform to the rule, this error code will be returned</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>If the bucket to which you want to add the rule does not exist, this error code will be returned</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>Invalid XML format. Please check against the RESTful API documentation</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>BadRequest</td>
<td>The maximum number of tags allowed for one bucket is exceeded. Currently, up to 10 tags can be set</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidTag</td>
<td>Tag key or value contains the reserved string &quot;cos:&quot; or &quot;Project&quot;</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>

### Example

#### Request

The following request writes two tags "{age:18}" and "{name:xiaoming}" to the bucket examplebucket-1250000000. COS successfully configured the tags and returns 204 (success).

```plaintext
PUT /Tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDrbAYjEBqqdEconpFi8NPFsOjrnX4LYUE&q-sign-time=1516361923;1517361973&q-key-time=1516361923;1517361973&q-url-param-list=tagging&q-header-list=content-md5;host&q-signature=71251feb4501494edcf740379404
Content-Md5: LIbd5t5HLPhuNWYkP6qHcQ==
Content-Length: 127
Content-Type: application/xml

<Tagging>
  <TagSet>
    <Tag>
      <Key>age</Key>
      <Value>18</Value>
    </Tag>
    <Tag>
      <Key>name</Key>
      <Value>xiaoming</Value>
    </Tag>
  </TagSet>
</Tagging>
```

#### Response

```
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 19 Jan 2018 11:40:22 GMT
Server: tencent-cos
x-cos-request-id: NWE2MWQ5MjZ1NTBhYzEUMGFyNTA5ODV1NTVjNDM=
```
GET Bucket tagging

Last updated: 2019-12-16 17:34:52

Description

COS supports setting tags for existing buckets. This API is used to get existing tags of a specified bucket.

If you call the GET Bucket tagging API using a sub-account, please make sure that you have obtained the permission to use this API from the root account.

Request

Sample Request

```
GET /?tagging HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This request operation does not use any special response header.

Response Body

The elements of this response body are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagging</td>
<td>None</td>
<td>Tag set</td>
<td>Container</td>
</tr>
<tr>
<td>TagSet</td>
<td>Tagging</td>
<td>Tag set</td>
<td>Container</td>
</tr>
<tr>
<td>Tag</td>
<td>Tagging.TagSet</td>
<td>Tag set, which can contain up to 10 tags</td>
<td>Containers</td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>Key</td>
<td>Tagging.TagSet.Tag</td>
<td>Tag key, which can contain up to 128 characters. A tag key can contain English letters, digits, spaces, plus signs, minus signs, underscores, equals signs, dots, colons, and slashes</td>
<td>String</td>
</tr>
<tr>
<td>Value</td>
<td>Tagging.TagSet.Tag</td>
<td>Tag value, which can contain up to 256 characters. A tag value can contain English letters, digits, spaces, plus signs, minus signs, underscores, equals signs, dots, colons, and slashes</td>
<td>String</td>
</tr>
</tbody>
</table>

**Error Codes**

The following describes some frequent special errors that may occur when you make this request:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SignatureDoesNotMatch</td>
<td>If the provided signature does not conform to the rule, this error code will be returned</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>If the bucket to which you want to add the rule does not exist, this error code will be returned</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchTagSetError</td>
<td>No bucket tag has been set for the requested bucket</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>

**Example**

**Request**

The following request is made to get the tags of the bucket `examplebucket-12500000000`. COS parses the request and returns the two existing tags of the bucket, `{age:18}` and `{name:xiaoming}.

```
GET /?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-12500000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDrbAYjEBqqdEconpFi8NPFsOjrnX4LYUE&q-sign-time=1516361923;1517361973&q-key-time=1516361923;1517361973&q-url-param-list=tagging&q-header-list=content-md5;host&q-signature=71251feb4501494edc6b01747f873803759404
Content-Md5: LIbd5t5HLPhuNWYkP6qHcQ==
Content-Length: 127
Content-Type: application/xml
```

**Response**

```
HTTP/1.1 200 OK
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2018 11:40:22 GMT
Server: tencent-cos
<Tagging>
  <TagSet>
    <Tag>
      <Key>age</Key>
      <Value>18</Value>
    </Tag>
    <Tag>
      <Key>name</Key>
      <Value>xiaoming</Value>
    </Tag>
  </TagSet>
</Tagging>
```
DELETE Bucket tagging

Last updated: 2019-12-16 17:35:16

Description
COS supports setting tags for existing buckets. This API is used to delete existing tags of a specified bucket.

If you call the `DELETE Bucket tagging` API using a sub-account, please make sure that you have obtained the permission to use this API from the root account.

**Request**

**Sample Request**

```
DELETE /?tagging HTTP 1.1
Host:<BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).
```

**Request Headers**

**Common Headers**

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

**Special Headers**

This request operation does not use any special request header.

**Request Body**

The request body of this request is empty.

**Response**

**Response Headers**

**Common Response Headers**

This response uses common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**

This request operation does not use any special response header.

**Response Body**

This request does not have special response body information.

**Error Codes**

The following describes some frequent special errors that may occur when you make this request:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SignatureDoesNotMatch</td>
<td>If the provided signature does not conform to the rule, this error code will be returned</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>If the bucket to which you want to add the rule does not exist, this error code will be returned</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>NoSuchTagSetError</td>
<td>No bucket tag has been set for the requested bucket</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>

**Example**
Request
The following request is made to delete the existing tags of the bucket examplebucket-1250000000. COS parses the request and deletes all the tags of the bucket.

DELETE /?tagging HTTP/1.1
User-Agent: curl/7.29.0
Accept: */*
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-ak=AKIDrbAYjEBqqdEconpFi8NPFsOjnrXAXLYUE&q-sign-time=1516361923;1517361973&q-key-time=1516361923;1517361973&q-url-param-list=tagging&q-header-list=content-md5;host&q-signature=71251feb4501494edcfb01747f873803759484
Content-Md5: LIbd5t5HLPhuNWYkP6qHcQ==
Content-Length: 127
Content-Type: application/xml

Response
HTTP/1.1 204 No Content
Content-Type: application/xml
Connection: close
Date: Fri, 19 Jan 2018 11:40:22 GMT
Static website(website)
PUT Bucket website

Description

This API (PUT Bucket website) is used to configure a static website for a Bucket by importing configuration files in XML format. The file size is limited to 64 KB.

Request

Request example

PUT /?website HTTP/1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date: date
Content-Length: length
Content-Type:application/xml
Authorization: Auth String
<XML file>

Authorization: Auth String (For more information, see Request Signature).

Request headers

Common headers

Common request headers are used for the implementation of this request operation. For more information, see Common Request Headers.

Special request headers

This request operation does not use any special request header.

Request body

<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <RedirectAllRequestsTo>
    <Protocol>https</Protocol>
  </RedirectAllRequestsTo>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKeyWith>404.html</ReplaceKeyWith>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>docs/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
<Redirect>
  <RoutingRule>
  <Condition>
  <KeyPrefixEquals>img/</KeyPrefixEquals>
  </Condition>
  <Redirect>
  <Protocol>https</Protocol>
  <ReplaceKeyWith>demo.jpg</ReplaceKeyWith>
  </Redirect>
  </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>

Details are described below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebsiteConfiguration</td>
<td>None</td>
<td>Static website configuration, including index document, error document, protocol conversion and routing rules.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>IndexDocument</td>
<td>WebsiteConfiguration</td>
<td>Index document</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Suffix</td>
<td>IndexDocument</td>
<td>Specifies the index document</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>ErrorDocument</td>
<td>WebsiteConfiguration</td>
<td>Error document</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Key</td>
<td>ErrorDocument</td>
<td>Specifies the common error page returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>RedirectAllRequestsTo</td>
<td>WebsiteConfiguration</td>
<td>Redirects all requests</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Protocol</td>
<td>RedirectAllRequestsTo</td>
<td>Specifies the routing protocol for the entire website. Only HTTP is supported.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>RoutingRules</td>
<td>WebsiteConfiguration</td>
<td>Sets routing rules in batch. A maximum of 100 routing rules can be set.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>RoutingRule</td>
<td>RoutingRules</td>
<td>Sets a single routing rule. You can set a routing rule for the requests with a specified path prefix or for the requests for which a specified error code is returned.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Condition</td>
<td>RoutingRule</td>
<td>Specifies the condition for the redirection. You cannot specify both prefix match routing condition and error code routing condition.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>HttpStatusCodeReturnedEquals</td>
<td>Condition</td>
<td>Specifies the redirection error code. Only 4XX status codes are supported. It has a higher priority than ErrorDocument.</td>
<td>Integer</td>
<td>No</td>
</tr>
<tr>
<td>KeyPrefixEquals</td>
<td>Condition</td>
<td>Specifies the prefix of the paths to be redirected.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Redirect</td>
<td>RoutingRule</td>
<td>Specifies the replacement rule when the redirection condition is met.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>ReplaceKeyWith</td>
<td>Redirect</td>
<td>Specifies the content which is used to replace the entire Key.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ReplaceKeyPrefixWith</td>
<td>Redirect</td>
<td>Specifies the content which is used to replace the prefix of Key. This is allowed only when Condition is KeyPrefixEquals.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**Response**

**Response headers**

**Common response headers**

This response uses common response headers. For more information, see Common Response Headers.

**Special response headers**

This response does not use any special response header.

**Response body**

The response body is empty.

**Error codes**

No special error message is returned for this request operation. For common error messages, see Error Codes.
Example

Request

```xml
PUT /website HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Date: Thu, 21 Sep 2017 13:05:41 +0000
Content-Type: application/xml
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjGFO&q-sign-time=1484814927;32557710927&q-key-time=1484814927;32557710927&q-header-list=host&q-url-param-list=website&q-signature=8b9f05dabc2e2578f3a79d732386e7cbade9033e3

<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <RedirectAllRequestsTo>
    <Protocol>https</Protocol>
  </RedirectAllRequestsTo>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKeyWith>404.html</ReplaceKeyWith>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>docs/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>img/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKey>demo.jpg</ReplaceKey>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu, 21 Sep 2017 13:05:54 GMT
Server: tencent-cos
x-cos-request-id: NTljM2I5MzJfMjQ4OGY3MGFfNzk4OV83Zg==
```
GET Bucket website

Last updated: 2019-12-14 10:56:31

Description

This API (GET Bucket website) is used to obtain the configuration of the static website associated with the Bucket.

Request

Request example

GET /?website HTTP/1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date:date
Authorization: Auth String

Authorization: Auth String (For more information, see Request Signature).

Request headers

Common headers

Common request headers are used for the implementation of this request operation. For more information, see Common Request Headers.

Special request headers

This request operation does not use any special request header.

Request body

The request body of this request is empty.

Response

Response headers

Common response headers

This response uses common response headers. For more information, see Common Response Headers.

Special response headers

This response does not use any special response header.

Response body

Application/xml data is returned in the response body. It contains the complete node data, as shown below:

```xml
<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <RedirectAllRequestsTo>
    <Protocol>https</Protocol>
  </RedirectAllRequestsTo>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
```
Details are described below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebsiteConfiguration</td>
<td>None</td>
<td>Static website configuration, including index document, error document, protocol conversion and routing rules.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>IndexDocument</td>
<td>WebsiteConfiguration</td>
<td>Index document</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Suffix</td>
<td>IndexDocument</td>
<td>Specifies the index document</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>ErrorDocument</td>
<td>WebsiteConfiguration</td>
<td>Error document</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Key</td>
<td>ErrorDocument</td>
<td>Specifies the common error page returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>RedirectAllRequestsTo</td>
<td>WebsiteConfiguration</td>
<td>Redirects all requests</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Protocol</td>
<td>RedirectAllRequestsTo</td>
<td>Specifies the routing protocol for the entire website. Only HTTP is supported.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>RoutingRules</td>
<td>WebsiteConfiguration</td>
<td>Sets routing rules in batch. A maximum of 100 routing rules can be set.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>RoutingRule</td>
<td>RoutingRules</td>
<td>Sets a single routing rule. You can set a routing rule for the requests with a specified path prefix or for the requests for which a specified error code is returned.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Condition</td>
<td>RoutingRule</td>
<td>Specifies the condition for the redirection. You cannot specify both prefix match routing condition and error code routing condition.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>ErrorCodeReturnedEquals</td>
<td>Condition</td>
<td>Specifies the redirection error code. Only 4XX status codes are supported. It has a higher priority than ErrorDocument.</td>
<td>Integer</td>
<td>No</td>
</tr>
<tr>
<td>KeyPrefixEquals</td>
<td>Condition</td>
<td>Specifies the prefix of the paths to be redirected.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Redirect</td>
<td>RoutingRule</td>
<td>Specifies the replacement rule when the redirection condition is met.</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>ReplaceKeyWith</td>
<td>Redirect</td>
<td>Specifies the content which is used to replace the entire Key.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ReplaceKeyPrefixWith</td>
<td>Redirect</td>
<td>Specifies the content which is used to replace the prefix of Key. This is allowed only when Condition is KeyPrefixEquals.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**Error codes**

No special error message is returned for this request operation. For common error messages, see Error Codes.

**Example**
Request

GET /?website HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Date:Thu, 21 Sep 2017 13:09:53 +0000
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=website&q-signature=a2d28e1b9023d09f9277982775a4b3705d8e23e

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 751
Connection: keep-alive
Date: Thu, 21 Sep 2017 13:15:39 GMT
Server: tencent-cos
x-cos-request-id: NTljM2JiN2FfMjQ4OGY3MGFfNzk4OV84Mg==

<WebsiteConfiguration>
  <IndexDocument>
    <Suffix>index.html</Suffix>
  </IndexDocument>
  <RedirectAllRequestsTo>
    <Protocol>https</Protocol>
  </RedirectAllRequestsTo>
  <ErrorDocument>
    <Key>Error.html</Key>
  </ErrorDocument>
  <RoutingRules>
    <RoutingRule>
      <Condition>
        <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKeyWith>404.html</ReplaceKeyWith>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>docs/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
      </Redirect>
    </RoutingRule>
    <RoutingRule>
      <Condition>
        <KeyPrefixEquals>img/</KeyPrefixEquals>
      </Condition>
      <Redirect>
        <Protocol>https</Protocol>
        <ReplaceKey>demo.jpg</ReplaceKey>
      </Redirect>
    </RoutingRule>
  </RoutingRules>
</WebsiteConfiguration>
DELETE Bucket website

Description
This API (DELETE Bucket website) deletes the configuration of the static website associated with the Bucket.

Request

Request example

DELETE /?website HTTP/1.1
Host:<BucketName-APPID>.<Region>.myqcloud.com
Date:date
Authorization: Auth String

Authorization: Auth String (For more information, see Request Signature).

HTTP headers

Common request headers
Common request headers are used for this action. For more information, see Common Request Headers.

Special request headers
Special headers are not used for this action.

Request body
Empty.

Response

Response headers

Common response headers
Common request headers are used for this action. For more information, see Common Response Headers.

Special response headers
Special headers are not used for this action.

Response body
Empty.

Error codes
No special error message is returned for this action. For common error messages, see Error Codes.

Example

Request

DELETE /?website HTTP/1.1
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Date:Thu, 21 Sep 2017 13:21:09 +0000
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjFf0&q-sign-time=1484816836;32557712836&q-key-time=1484816836;32557712836&q-url-param-list=website&q-signature=50d62be50a

Response
HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu, 21 Sep 2017 13:21:18 GMT
Server: tencent-cos
x-cos-request-id: NTljM2ZljjY2RfMjQ4OGY3MGF1Nz40V84Mw==

Bucket inventory(inventory)
PUT Bucket inventory

Feature Description

PUT Bucket inventory is used to create an inventory task in a bucket. You can use this request to create an inventory task after naming it. For more information, see Inventory Feature Overview.

- Up to 1,000 inventory tasks can be configured in one COS bucket.
- You must write a bucket policy to the destination bucket for COS to put the result file of the inventory task in it.
- When calling this request, make sure that you have sufficient permission to manipulate the bucket's inventory tasks, which is granted to the bucket owner by default. If you do not have it, apply for it to the bucket owner first.
- If you specify a prefix for inventory delivery, the COS backend will automatically add `/` to the prefix you specify. If you specify `Prefix` as a prefix, the COS backend delivery inventory report path is `Prefix/inventory_report`.

Request

Request Example

```http
PUT /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
Content-MD5: MD5
Authorization: Auth String (see Request Signature for details).
```

Request Parameter

Calling PUT Bucket inventory requires the inventory task name parameter. The parameter is in the following format:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Name of the inventory task. Default value: None Valid characters: a-z, A-Z, 0-9, -, _, .</td>
</tr>
</tbody>
</table>

Type | Required
--|---
String | Yes

Request Header

Common Header

The implementation of this request operation uses a public request header. For more information about the public request header, see Common Request Headers.

Non-common Header

This request operation has no special request headers.

Request Body

You can use the XML language to set specific configuration information for the inventory task in the request body, including the objects to be analyzed by the inventory task, frequency of analysis, analysis dimensions, format of analysis result, and storage location.

```xml
<InventoryConfiguration>
  <Id>list1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <COSBucketDestination>
      ...
    </COSBucketDestination>
  </Destination>
</InventoryConfiguration>
```
<Format>CSV</Format>
<AccountId>100000000001</AccountId>
<Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
<Prefix>list1</Prefix>
<Encryption>
  <SSE-COS></SSE-COS>
</Encryption>
</COSBucketDestination>
</Destination>
</Schedule>
<Frequency>Daily</Frequency>
</Schedule>
<Filter>
  <Prefix>myPrefix</Prefix>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
</InventoryConfiguration>

The content is described in details below:

<table>
<thead>
<tr>
<th>Node name</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>InventoryConfiguration</td>
<td>None</td>
<td>This contains configuration parameters of the inventory</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Id</td>
<td>InventoryConfiguration</td>
<td>Inventory name, corresponding to the id in the request parameter</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>IsEnabled</td>
<td>InventoryConfiguration</td>
<td>Flag about whether the inventory is enabled. If this is set to true, the inventory is enabled; if false, no inventories will be generated</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>IncludedObjectVersions</td>
<td>InventoryConfiguration</td>
<td>Whether to include object versions in the inventory If this is set to All, the inventory will include all object versions and add VersionId, IsLatest, and DeleteMarker fields If Current, no object version information will be included in the inventory</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Node name</td>
<td>Parent node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>Filter</td>
<td>InventoryConfiguration</td>
<td>This filters the objects to be analyzed. The inventory feature will analyze the objects that match the prefix set in Filter</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Prefix</td>
<td>Filter</td>
<td>Prefix of the objects to be analyzed</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>OptionalFields</td>
<td>InventoryConfiguration</td>
<td>This sets the analysis items that should be included in the inventory result</td>
<td>Container</td>
<td>No</td>
</tr>
<tr>
<td>Field</td>
<td>OptionalFields</td>
<td>Name of the analysis items that can be optionally included in the inventory result. Optional fields include Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, and ReplicationStatus</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Schedule</td>
<td>InventoryConfiguration</td>
<td>This configures the inventory task cycle</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Frequency</td>
<td>Schedule</td>
<td>Inventory task cycle. Options include daily and weekly</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Destination</td>
<td>InventoryConfiguration</td>
<td>This describes the information of the inventory result storage</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>COSBucketDestination</td>
<td>Destination</td>
<td>Information of the bucket where the inventory result is stored after export</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Bucket</td>
<td>COSBucketDestination</td>
<td>Name of the bucket where the inventory result is stored</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>AccountId</td>
<td>COSBucketDestination</td>
<td>ID of the bucket owner such as 100000000001</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Prefix</td>
<td>COSBucketDestination</td>
<td>Prefix of the inventory result</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Format</td>
<td>COSBucketDestination</td>
<td>File format of the inventory result. CSV is available</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Encryption</td>
<td>COSBucketDestination</td>
<td>Option to provide server-side encryption for the inventory result</td>
<td>Container</td>
<td>No</td>
</tr>
</tbody>
</table>
Response

Response Header

Common Response Header
This response uses a common response header. For more information about the common response header, see Common Response Headers.

Special Response Header
The response to this request has no special response headers.

Response Body
The response body return of this request is empty.

Error Codes
Some frequent special errors that may occur with this request are listed below. For common error codes, see Error Codes.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>Status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>Invalid parameter value</td>
<td>HTTP 400 Bad Request</td>
</tr>
<tr>
<td>TooManyConfigurations</td>
<td>The number of inventories has reached the upper limit of 1,000</td>
<td>HTTP 400 Bad Request</td>
</tr>
<tr>
<td>AccessDenied</td>
<td>Unauthorized access. You probably do not have access to the bucket</td>
<td>HTTP 403 Forbidden</td>
</tr>
</tbody>
</table>

Use Case

Request
This example adds an inventory task named list1 to the bucket examplebucket-1250000000.
This inventory task analyzes the objects prefixed with myPrefix in the bucket and all their versions.
The frequency of analysis is once a day.
Analysis dimensions include Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, and ReplicationStatus.
The analysis result is stored in the bucket examplebucket-1250000000 as a CSV file, which is prefixed with list1 and encrypted with SSE-COS.

PUT /?inventory&id=list1 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0Ao7c11gP9cXFrJ0Ua110cR08JW&q-sign-time=1503888878;1503889238&q-key-time=1503888878;1503889238&q-header-list=host&q-url-param-list=inventory&q-signature=254bf9cd3d6615e89a36a652437f9d45c5f63f9
Content-MD5: AAq9nzrpsz5LJ4UEe1f6Q==
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 1024

<?xml version = "1.0" encoding = "UTF-8">
<InventoryConfiguration xmlns = "http://....">
  <Id>list1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <COSBucketDestination>
      <Format>CSV</Format>
      <AccountId>100000000001</AccountId>
      <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
      <Prefix>list1</Prefix>
      <Encryption>SSE-COS</Encryption>
    </COSBucketDestination>
  </Destination>
</InventoryConfiguration>
Response

After the request above is made, COS returns the following response, indicating that the inventory task list1 has been successfully configured.

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, Aug 28, 2018 02:53:38 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF84Y2M
```
GET Bucket inventory

Feature Description

The GET Bucket inventory API is used to query the inventory task information in a bucket. When initiating this request, you need to provide the name of the inventory task and get the request signature to indicate that the request has been authorized. For more information of the inventory feature, see Inventory Feature Overview.

- When calling this request, make sure that you have sufficient permission to manipulate the bucket's inventory tasks.
- This permission is granted to the bucket owner by default. If you do not have it, apply for it to the bucket owner first.
- If you specify a prefix for inventory delivery, the COS backend will automatically add `/` to the prefix you specify. If you specify `Prefix` as a prefix, the COS backend delivery inventory report path is `Prefix/inventory_report`.

Request

Request Example

GET /?inventory&id=inventory-configuration-ID HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for details).

Request Parameter

Calling Get Bucket inventory requires the inventory task name parameter. The parameter is in the following format:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Name of the inventory task. Default value: None Valid characters: <code>a-z, A-Z, 0-9, -, _</code></td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Header

Common Header

The implementation of this request operation uses a public request header. For more information about the public request header, see Common Request Headers.

Non-common Header

This request operation has no special request headers.

Request Body

The request body of this request is empty.

Response

Response Header

Common Response Header

This response uses a common response header. For more information about the common response header, see Common Response Headers.

Special Response Header

This response has no special response headers.

Response Body
The return of this response body is application/xml data. Below is an example containing all the node data:

```xml
<InventoryConfiguration>
  <Id>list1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <COSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1250000000</AccountId>
      <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
      <Prefix>list1</Prefix>
      <Encryption>
        <SSE-COS></SSE-COS>
      </Encryption>
    </COSBucketDestination>
    </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>myPrefix</Prefix>
  </Filter>
  <IncludedObjectVersions>All</IncludedObjectVersions>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>ReplicationStatus</Field>
  </OptionalFields>
</InventoryConfiguration>
```

The content is described in details below:

<table>
<thead>
<tr>
<th>Node name</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Configuration</td>
<td>None</td>
<td>This contains configuration parameters of the inventory</td>
<td>Container</td>
</tr>
<tr>
<td>Id</td>
<td>Inventory Configuration</td>
<td>Inventory name, corresponding to the ID in the request parameter</td>
<td>Container</td>
</tr>
<tr>
<td>IsEnabled</td>
<td>Inventory Configuration</td>
<td>Flag about whether the inventory is enabled. If this is set to true, the inventory is enabled; if false, no inventories will be generated</td>
<td>String</td>
</tr>
<tr>
<td>IncludedObject Versions</td>
<td>Inventory Configuration</td>
<td>Whether to include object versions in the inventory. If this is set to All, the inventory will include all object versions and add VersionId, IsLatest, and DeleteMarker fields; if Current, no object version information will be included in the inventory</td>
<td>String</td>
</tr>
<tr>
<td>Filter</td>
<td>Inventory Configuration</td>
<td>This filters the objects to be analyzed. The inventory feature will analyze the objects that match the prefix set in Filter</td>
<td>Container</td>
</tr>
<tr>
<td>Node name</td>
<td>Parent node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Prefix</td>
<td>Filter</td>
<td>Prefix of the objects to be analyzed</td>
<td>String</td>
</tr>
<tr>
<td>OptionalFields</td>
<td>Inventory Configuration</td>
<td>This sets the analysis dimensions that should be included in the inventory result</td>
<td>Container</td>
</tr>
<tr>
<td>Field</td>
<td>OptionalFields</td>
<td>Name of the analysis dimension that can be optionally included in the inventory result. Optional fields include Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, and ReplicationStatus</td>
<td>String</td>
</tr>
<tr>
<td>Schedule</td>
<td>Inventory Configuration</td>
<td>This configures the inventory task cycle</td>
<td>Container</td>
</tr>
<tr>
<td>Frequency</td>
<td>Schedule</td>
<td>Inventory task cycle. Options include daily and weekly</td>
<td>String</td>
</tr>
<tr>
<td>Destination</td>
<td>Inventory Configuration</td>
<td>This describes the information of the inventory result storage</td>
<td>Container</td>
</tr>
<tr>
<td>COSBucket Destination</td>
<td>Destination</td>
<td>Information of the bucket where the inventory result is stored after export</td>
<td>Container</td>
</tr>
<tr>
<td>Bucket</td>
<td>COSBucket Destination</td>
<td>Name of the bucket where the inventory result is stored</td>
<td>String</td>
</tr>
<tr>
<td>AccountId</td>
<td>COSBucket Destination</td>
<td>ID of the bucket owner</td>
<td>String</td>
</tr>
<tr>
<td>Prefix</td>
<td>COSBucket Destination</td>
<td>Prefix of the inventory result</td>
<td>String</td>
</tr>
<tr>
<td>Format</td>
<td>COSBucket Destination</td>
<td>File format of the inventory result. CSV and OCR are available</td>
<td>String</td>
</tr>
<tr>
<td>Encryption</td>
<td>COSBucket Destination</td>
<td>Option to provide server-side encryption for the inventory result</td>
<td>Container</td>
</tr>
<tr>
<td>SSE-COS</td>
<td>Encryption</td>
<td>Encryption with COS-managed key</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Error Codes**

This request does not generate special error messages. For common error codes, see [Error Codes](#).

**Use Case**

**Request**

The following request sample shows the configuration information for getting the inventory task list1 from the bucket examplebucket-1250000000.

```
GET /?inventory&id=list1 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
```
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0A07c1lgPvF9cXFrJ0Ba11DvR88JM&q-sign-time=15083895278;15083895638&q-key-time=15083895278;15083895638&q-header-list=host&q-url-param-list=inventory&q-signature=f77900be432f72b16af6d222b4b349aabd837cb9
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com

Response

After the request above is made, COS returns the following response indicating that the inventory task list1 is currently enabled in the bucket. This inventory task analyzes the objects prefixed with myPrefix in the bucket examplebucket-1250000000 and all their versions. The frequency of analysis is once a day.

Analysis dimensions include Size, LastModified, StorageClass, ETag, IsMultipartUploaded, and ReplicationStatus.

The analysis result is stored in the bucket examplebucket-1250000000 as a CSV file, which is prefixed with list1 and encrypted with SSE-COS.

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 331
Date: Mon, August 28, 2018 02:53:39 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF84Y2M
<?xml version = "1.0" encoding = "UTF-8">
<InventoryConfiguration xmlns = "http://....">
  <Id>list1</Id>
  <IsEnabled>true</IsEnabled>
  <Destination>
    <COSBucketDestination>
      <Format>CSV</Format>
      <AccountId>1250000000</AccountId>
      <Bucket>qcs::cos:ap-guangzhou::examplebucket-1250000000</Bucket>
      <Prefix>list1</Prefix>
      <Encryption>
        <SSE-COS></SSE-COS>
      </Encryption>
    </COSBucketDestination>
  </Destination>
  <Schedule>
    <Frequency>Daily</Frequency>
  </Schedule>
  <Filter>
    <Prefix>myPrefix</Prefix>
    <IncludedObjectVersions>All</IncludedObjectVersions>
  </Filter>
  <OptionalFields>
    <Field>Size</Field>
    <Field>LastModifiedDate</Field>
    <Field>ETag</Field>
    <Field>StorageClass</Field>
    <Field>IsMultipartUploaded</Field>
    <Field>ReplicationStatus</Field>
  </OptionalFields>
</InventoryConfiguration>
List Bucket Inventory Configurations

Feature Description

List Bucket Inventory Configurations requests all inventory tasks in a bucket. Up to 1,000 inventory tasks can be configured in one bucket.

The result is listed in pages, i.e., up to 100 inventory tasks can be shown on one page at a time. Please check the value of the IsTruncated node in the request is correct. If IsTruncated is false, all inventory tasks in the bucket have been listed. If IsTruncated is true and the NextContinuationToken node has been assigned with a value, you can pass that value to the continuation-token node to get the results on the next page. For more information of the inventory feature, see Inventory Feature Overview.

Make sure that you have sufficient permission to access the bucket’s inventory tasks you requested. By default, the bucket owners are granted the access permissions. If you do not have the permission, apply for it from the bucket owner first.

Request

Request Example

GET /?inventory HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for details).

Request Parameter

The request parameter is in the following format:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuation-token</td>
<td>If IsTruncated in the COS response body is true and there is a parameter</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>value in the NextContinuationToken node, you can use this parameter as</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the value of continuation-token node to get the list of inventory tasks on</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the next page</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default value: None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Request Header

Common Header

The implementation of this request operation uses a common request header. For more information about the public request header, see Common Request Headers.

Non-common Header

This request operation has no special request headers.

Request Body

The request body of this request is empty.

Response

Response Header

Common Response Header
This response uses a common response header. For more information about the common response header, see Common Response Headers.

Special Response Header
This response has no special response headers.

Response Body
The return of this response body is application/xml data. Below is an example containing all the node data:

```xml
<ListInventoryConfigurationResult>
  <InventoryConfiguration>
    <Id>list1</Id>
    <IsEnabled>True</IsEnabled>
    <Destination>
      <COSBucketDestination>
        <Format>CSV</Format>
        <AccountId>1250000000</AccountId>
        <Bucket>qcs::cos:ap-beijing::examplebucket-1250000000</Bucket>
        <Prefix>list1</Prefix>
        <SSE-COS></SSE-COS>
      </COSBucketDestination>
    </Destination>
    <Schedule>
      <Frequency>Daily</Frequency>
    </Schedule>
    <Filter>
      <Prefix>myPrefix</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
      <Field>Size</Field>
      <Field>LastModifiedDate</Field>
      <Field>ETag</Field>
      <Field>StorageClass</Field>
      <Field>IsMultipartUpload</Field>
      <Field>ReplicationStatus</Field>
    </OptionalFields>
  </InventoryConfiguration>
  <InventoryConfiguration>
    <Id>list2</Id>
    <IsEnabled>True</IsEnabled>
    <Destination>
      <COSBucketDestination>
        <Format>CSV</Format>
        <AccountId>1250000000</AccountId>
        <Bucket>qcs::cos:ap-beijing::examplebucket-1250000000</Bucket>
        <Prefix>list2</Prefix>
        <SSE-COS></SSE-COS>
      </COSBucketDestination>
    </Destination>
    <Schedule>
      <Frequency>Weekly</Frequency>
    </Schedule>
    <Filter>
      <Prefix>myPrefix2</Prefix>
    </Filter>
    <IncludedObjectVersions>All</IncludedObjectVersions>
    <OptionalFields>
      <Field>Size</Field>
      <Field>LastModifiedDate</Field>
      <Field>ETag</Field>
      <Field>StorageClass</Field>
    </OptionalFields>
  </InventoryConfiguration>
  <IsTruncated>false</IsTruncated>
</ListInventoryConfigurationResult>
```
The content is described in details below:

<table>
<thead>
<tr>
<th>Node name</th>
<th>Parent node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Inventory Configuration Results</td>
<td>None</td>
<td>List of all inventory tasks in the bucket</td>
<td>Container</td>
</tr>
<tr>
<td>Inventory Configuration</td>
<td>ListInventory Configuration Results</td>
<td>Details of the inventory tasks. See GET Bucket inventory for the XML structure</td>
<td>Container</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>ListInventory Configuration Results</td>
<td>Flag about whether all inventory tasks have been listed. If yes, it is false; otherwise, it is true</td>
<td>Boolean</td>
</tr>
<tr>
<td>Continuation Token</td>
<td>ListInventory Configuration Results</td>
<td>Flag of the inventory list on the current page, which can be understood as the page number. It corresponds to the continuation-token parameter in the request</td>
<td>String</td>
</tr>
<tr>
<td>NextContinuation Token</td>
<td>ListInventory Configuration Results</td>
<td>Flag of the next page of inventory list. If there is a value in this parameter, the value can be used as the continuation-token parameter to initiate a GET request to get the inventory task information of the next page</td>
<td>String</td>
</tr>
</tbody>
</table>

Error Codes

This request does not generate special error messages. For common error codes, see Error Codes.

Use Case

Request

The following request sample shows the configuration information for getting the inventory task list1 from the bucket examplebucket-1250000000.

```
GET /?inventory HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0Ao7c1lgPv9cxFrJD0a1iCr88JN&q-sign-time=1503895278;1503895638&q-key-time=1503895278;1503895638&q-header-list=host&q-url-param-list=inventory&q-signature=f77900be432072916af82222bca849aabaeb37cb9
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

Response

After the request is made, COS returns the following response indicating that currently there are inventory tasks list1 and list2 in the bucket.

**Inventory task list1**

Analyze the objects prefixed with myPrefix in the bucket examplebucket-1250000000 and all their versions.

The frequency of analysis is once a day.

Analysis dimensions include Size, LastModifiedDate, StorageClass, ETag, IsMultipartUploaded, and ReplicationStatus.

The analysis result is stored in the bucket examplebucket-1250000000 as a CSV file, which is prefixed with list1 and encrypted with SSE-COS.

**Inventory task list2**

Analyze the objects prefixed with myPrefix2 in the bucket examplebucket-1250000000 and all their versions.

The frequency of analysis is once a week. The analysis dimensions include Size, LastModifiedDate, StorageClass, and ETag.
The analysis result is stored in the bucket examplebucket-1250000000 as a CSV file, which is prefixed with list2 and encrypted with SSE-COS.

Suppose there are 100 inventory tasks on this page. If IsTruncated is true, COS will further return NextContinuationToken, whose value can be used as the parameter of continuation-token in the GET request to get the next page of information.

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 331
Date: Mon, 28 Aug 2018 02:53:39 GMT
Server: tencent-cos
x-cos-request-id: NTlhMzg1ZWVfMjQ4OGY3MGFfMWE1NF84Y2M
<xml version = "1.0" encoding = "UTF-8">
<ListInventoryConfigurationResult xmlns = "http://....">
<InventoryConfiguration>
<Id>list1</Id>
<IsEnabled>True</IsEnabled>
<Destination>
<COSBucketDestination>
<Format>CSV</Format>
<AccountId>1250000000</AccountId>
<Bucket>qcs::cos:ap-beijing::examplebucket-1250000000</Bucket>
<Prefix>list1</Prefix>
<SSE-COS></SSE-COS>
</COSBucketDestination>
</Destination>
<Schedule>
<Frequency>Daily</Frequency>
</Schedule>
<Filter>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
</OptionalFields>
</InventoryConfiguration>
<InventoryConfiguration>
<Id>list2</Id>
<IsEnabled>True</IsEnabled>
<Destination>
<COSBucketDestination>
<Format>CSV</Format>
<AccountId>1250000000</AccountId>
<Bucket>qcs::cos:ap-beijing::examplebucket-1250000000</Bucket>
<Prefix>list2</Prefix>
<SSE-COS></SSE-COS>
</COSBucketDestination>
</Destination>
<Schedule>
<Frequency>Weekly</Frequency>
</Schedule>
<Filter>
</Filter>
<IncludedObjectVersions>All</IncludedObjectVersions>
</OptionalFields>
</InventoryConfiguration>
</ListInventoryConfigurationResult>
<IsTruncated>false</IsTruncated>
------If ContinuationToken was provided in the request---
<ContinuationToken>...</ContinuationToken>
<IsTruncated>true</IsTruncated>
DELETE Bucket inventory

Last updated: 2019-07-10 15:08:01

Feature Description
DELETE Bucket inventory is used to delete the specified inventory task in the bucket. You need to provide the name of the inventory task to be deleted. For more information of the inventory feature, see Inventory Feature Overview.

- When calling this request, make sure that you have sufficient permission to manipulate the bucket's inventory tasks.
- This permission is granted to the bucket owner by default. If you do not have it, apply for it to the bucket owner first.

**Request**

**Request Example**

```
DELETE /?inventory&id=inventory-configuration-id HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

**Request Parameter**

Calling DELETE Bucket inventory requires the inventory task name parameter. The parameter is in the following format:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Name of the inventory task. Default value: None</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

| Valid characters: a-z, A-Z, 0-9, -,_, . |

**Request Header**

**Common Header**

The implementation of this request operation uses a public request header. For more information about the public request header, see Common Request Headers.

**Non-common Header**

This request operation has no special request headers.

**Request Body**

The request body of this request is empty.

**Response**

**Response Header**

**Common Response Header**

This response uses a common response header. For more information about the common response header, see Common Response Headers.

**Special Response Header**

This response has no special response headers.

**Response Body**

The response body return is empty.

**Use Case**

**Request**
The following request sample deletes the inventory task list1 from the bucket examplebucket-1250000000.

```
DELETE /?inventory&id=list1 HTTP/1.1
Date: Mon, 28 Aug 2018 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOa7cIgPvF9cXFvJR0a11CvR88JM&q-sign-time=1503901499;1503901859&q-key-time=1503901499;1503901859&q-
header-list=host&q-url-param-list=inventory&q-signature=761f3f6449c6a1684464f4bf0c6f2f2f0a4e7e0
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
```

Response

After the request above is made, COS returns a response of 204 No Content, indicating that the inventory task list1 has been successfully deleted from the bucket.

```
HTTP/1.1 204 No Content
Server: tencent-cos
Date: Mon, 28 Aug 2018 02:53:40 GMT
x-cos-id-2:0dfafa/DAPDIFdafsafDFdSFdKJdafaKJ2
x-cos-request-id: NTlhM2I3M2JfMjQ4OGY3MGFfMWE1NF84ZTU=
```
Versioning
PUT Bucket versioning

Last updated: 2019-09-24 17:33:07

Description

The PUT Bucket versioning API is used to enable or suspend versioning for a bucket.

Notes

1. If you have never enabled versioning for the bucket, GET Bucket versioning will not return a versioning state value.
2. Once enabled, versioning can only be suspended but cannot be disabled.
3. Set the versioning state value to Enabled or Suspended to enable or suspend versioning, respectively.
4. To set versioning for a bucket, you need to have the permission to write to the bucket.

Request

Sample Request

```
PUT /?versioning HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this request uses a common request header. For more information on common request headers, see Common Request Headers.

Special Headers

This request does not use any special request header.

Request Body

```
<VersioningConfiguration>
<Status></Status>
</VersioningConfiguration>
```

Please find the details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersioningConfiguration</td>
<td>None</td>
<td>Describes the detailed information on versioning</td>
<td>Container</td>
</tr>
<tr>
<td>Status</td>
<td>VersioningConfiguration</td>
<td>Indicates whether versioning is enabled; enumerators: Suspended, Enabled</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Response

Response Headers

Common Response Headers
This response contains a common response header. For more information on common response headers, see Common Response Headers.

Special Response Headers
This response does not have special response headers.

Response Body
This response body is empty.

Error Codes
Some common special errors that may occur with the request are listed below. For more COS error codes or a complete list of errors, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>400 Bad Request</td>
<td>If the xml body of the request to enable versioning is empty, InvalidArgument will be returned.</td>
</tr>
<tr>
<td>InvalidDigest</td>
<td>400 Bad Request</td>
<td>1. The Content-MD5 carried does not match the request body calculated by the server; 2. Only two values are valid for the versioning state: Enabled or Suspended. If other values are entered, InvalidArgument will be returned.</td>
</tr>
</tbody>
</table>

Samples

PUT /?versioning HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Content-Type: application/xml
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9Sm****&q-sign-time=1480932292;1981012292&q-key-time=1480932292;1981012292&q-url-param-list=versioning&q-header-list=host&q-signature=47ec2b0c7378ece394d3b0ad90e120a32f****
Content-Length: 83

<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Wed, 23 Aug 2017 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0****
GET Bucket versioning

Description

The GET Bucket versioning API is used to get the versioning information of a bucket.

Notes

1. To get the versioning state of a bucket, you need to have the permission to read the bucket.
2. There are three versioning states: not enabled, enabled, or suspended.

   - If you have never enabled or suspended versioning for the bucket, the response is:
     <VersioningConfiguration/>

   - If you have enabled versioning for the bucket, the response is:
     <VersioningConfiguration>
     <Status>Enabled</Status>
     </VersioningConfiguration>

   - If you have suspended versioning for the bucket, the response is:
     <VersioningConfiguration>
     <Status>Suspended</Status>
     </VersioningConfiguration>

Request

Sample Request

GET /?versioning HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String

Request Headers

Common Headers

The implementation of this request uses a common request header. For more information on common request headers, see Common Request Headers.

Special Headers

This request does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers
Common Response Headers
This response contains a common response header. For more information on common response headers, see Common Response Headers.

Special Response Headers
This response does not have special response headers.

Response Body

```
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Please find the details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>VersioningConfiguration</td>
<td>None</td>
<td>Describes the detailed information on versioning</td>
<td>Container</td>
</tr>
<tr>
<td>Status</td>
<td>VersioningConfiguration</td>
<td>Indicates whether versioning is enabled; enumerators: Suspended, Enabled</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Samples

GET /?versioning HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15Isski8QKTZhbAoS0hpc0gVlxs0Se****&q-sign-time=1480932292;1981012292&q-key-time=1480932292;1981012292&q-url-param-list=versioning&q-header-list=host&q-signature=5118a936049f4d44482bb61380235cf4abe****

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 120
Connection: keep-alive
Date: Wed, 23 Aug 2017 08:15:16 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5OTRfZDNhZDM1MGFfMjYyMTFfZmU3****

<?xml version='1.0' encoding='utf-8' ?>
<VersioningConfiguration>
  <Status>Enabled</Status>
</VersioningConfiguration>
```
Cross-region Replication (replication)
PUT Bucket replication


Description

The PUT Bucket replication API is used to configure a cross-region replication rule for a bucket where versioning is enabled. If there is already a rule for the bucket, the existing one will be overwritten.

When using this API, make sure that the bucket has versioning enabled. For more information, see the API documentation PUT Bucket versioning.

Request

Sample Request

PUT /?replication HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-MD5: MD5
Authorization: Auth String
request body

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this request uses a common request header. For more information on common request headers, see Common Request Headers.

Special Headers

This request does not use any special request header.

Request Body

You need to set the configuration information for cross-region replication in the request body, including the status of the cross-region replication rule, content to be replicated, and the name and the region of the destination bucket. Currently, you can only configure one cross-region replication rule for each bucket that has versioning enabled.

<ReplicationConfiguration>
<Role=qcs::cam::uin/<OwnerUin>:uin/<SubUin></Role>
<Rule>
<Status></Status>
<ID></ID>
<Prefix></Prefix>
<Destination>
<Bucket=qcs::cos:<Region>::<BucketName-APPID></Bucket>
</Destination>
</Rule>
</ReplicationConfiguration>

The content is described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationConfiguration</td>
<td>None</td>
<td>Describes all cross-region replication configuration information</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Node Name (Keyword) | Parent Node       | Description                                                                 | Type      | Required |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>ReplicationConfiguration</td>
<td>Initiator ID: qcs:cam:uin/&lt;OwnerUin&gt;:uin/&lt;SubUin&gt;</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Rule</td>
<td>ReplicationConfiguration</td>
<td>Specific configuration information of up to 1,000 rules.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>ReplicationConfiguration.Rule</td>
<td>Name used to identify a specific rule</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Status</td>
<td>ReplicationConfiguration.Rule</td>
<td>Indicates whether a rule is in effect; enumerators: Enabled, Disabled</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Prefix</td>
<td>ReplicationConfiguration.Rule</td>
<td>Prefix matching policy. Policies cannot overlap; otherwise, an error will be returned. The prefix matching root directory is empty</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Destination</td>
<td>ReplicationConfiguration.Rule</td>
<td>Destination bucket information</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Bucket</td>
<td>ReplicationConfiguration.Rule.Destination</td>
<td>Resource ID: qcs:cos:&lt;Region&gt;::&lt;BucketName-APPID&gt;</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>StorageClass</td>
<td>ReplicationConfiguration.Rule.Destination</td>
<td>Storage class; enumerators: STANDARD, STANDARD_IA. It follows the storage class of the source bucket by default. Note: Currently, cross-region replication does not support specifying the storage class of object copies as archive storage. If you need to set this class for the copies, you can configure lifecycle management for the destination bucket. For more information, see PUT Bucket lifecycle</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Response

Response Headers

Common Response Headers
This response contains a common response header. For more information on common response headers, see Common Response Headers.

Special Response Headers
The response to this request does not have special response headers.

Response Body
This response body is empty.

Error Codes
Some common special errors that may occur with this request are listed below. For common error codes, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidBucketState</td>
<td>Versioning is not enabled for the current bucket; therefore, cross-region replication cannot be enabled.</td>
<td>409 Conflict</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Invalid parameter.</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>

Samples

Request
The following PUT Bucket replication request adds a cross-region replication configuration to the bucket `originbucket-1250000000`, specifying that objects prefixed with `testPrefix` are to be replicated to the destination bucket `destinationbucket-1250000000` in Guangzhou.

```xml
PUT /?replication HTTP/1.1
Date: Mon, 28 Aug 2017 02:53:38 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0A7clgPvF9cXFJDR0a1lCvR****&q-sign-time=1503888878;1503889238&q-key-time=1503888878;1503889238&q-header-list=host&q-url-param-list=replication&q-signature=254bf0c3c6615e89a36b52437f9d45c5f****
Content-MD5: AAq9nzrpsz5LJ4UEe16Q==
Host: originbucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 312

<ReplicationConfiguration>
  <Role>qcs::cam::uin/100000000001:uin/100000000001</Role>
  <Rule>
    <Status>Enabled</Status>
    <ID>RuleId_01</ID>
    <Prefix>testPrefix</Prefix>
    <Destination>
      <Bucket>qcs::cos:ap-guangzhou::destinationbucket-1250000000</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Response

After the request above is made, COS returns the following response, indicating that the cross-region replication rule has been successfully configured.

```xml
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 14 Apr 2019 07:06:19 GMT
Server: tencent-cos
x-cos-bucket-region: ap-guangzhou
x-cos-request-id: NWQwMzQ3NmJfMjRiMjU4NjRfOTM4NV82ZDU1****
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVwMDA0bC80NWRkZjkrZDNI1Me1M2E2MTRiYWJhNjI4MTQzZjlkZyZkY2ZlNzQ4MzUy
ZTg1NGRiNVY0NTJiOGUyNzBVY2ZgZTgwZmE0YTU=
```
GET Bucket replication

Description

The GET Bucket replication API is used to query the cross-region replication configuration in a bucket. When initiating this request, you need to get the request signature to show that the request has been authorized.

Request

Sample Request

```
GET /?replication HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this request uses a common request header. For more information on common request headers, see Common Request Headers.

Special Headers

This request does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response contains a common response header. For more information on common response headers, see Common Response Headers.

Special Response Headers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-replication-rule-creation-time</td>
<td>UTC timestamp</td>
<td>Creation time of the cross-region replication rule</td>
</tr>
</tbody>
</table>

Response Body

The return of this response body is application/xml data. Below is an example containing all the node data:

```
<ReplicationConfiguration>
  <Role>qcs::cam::uin/[UIN]:uin/[Subaccount]</Role>
  <Rule>
    <Status></Status>
    <ID></ID>
    <Prefix></Prefix>
    <Destination>
      <Bucket>qcs::cos:[Region]:[BucketName-APPID]</Bucket>
      </Destination>
  </Rule>
</ReplicationConfiguration>
```
The content is described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationConfiguration</td>
<td>None</td>
<td>Describes all cross-region replication configuration information</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Role</td>
<td>ReplicationConfiguration</td>
<td>Initiator ID: qcs::cam::uin/&lt;OwnerUin&gt;:uin/&lt;SubUin&gt;</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Rule</td>
<td>ReplicationConfiguration</td>
<td>Specific configuration information of up to 1,000 rules.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>ReplicationConfiguration.Rule</td>
<td>Name used to identify a specific rule</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Status</td>
<td>ReplicationConfiguration.Rule</td>
<td>Indicates whether a rule is in effect; enumerators: Enabled, Disabled</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Prefix</td>
<td>ReplicationConfiguration.Permission</td>
<td>Prefix matching policy. Policies cannot overlap; otherwise, an error will be returned. The prefix matching root directory is empty</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Destination</td>
<td>ReplicationConfiguration.Permission.Destination</td>
<td>Destination bucket information</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>StorageClass</td>
<td>ReplicationConfiguration.Permission.Destination</td>
<td>Storage class; enumerators: Standard, Standard_IA. It follows the storage class of the source bucket by default.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Error Codes

Some common special errors that may occur with this request are listed below. For common error codes, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReplicationConfigurationNotFound</td>
<td>No cross-region replication rule found.</td>
<td>404 Not Found</td>
</tr>
</tbody>
</table>

Samples

Request

The following request sample gets the cross-region replication configuration from the bucket `originbucket-1250000000`.

```
GET /?replication HTTP/1.1
Date: Fri, 14 Apr 2019 07:17:19 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJDBa11CvR88WM&q-sign-time=1503895278;1503895638&q-key-time=1503895278;1503895638&q-header-list=host&q-url-param-list=replication&q-signature=f77900be432072b16afd822b40349a00bd837cb9
Host: originbucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 0
```

Response

After the request above is made, COS returns the following response, indicating that the current cross-region replication configuration for the bucket is enabled. In the rule, the objects to be replicated are the ones prefixed with `testPrefix` in the bucket `originbucket-1250000000`, and the storage class of the object copies is by default the same as that of the objects in the source bucket.

```
Content-Type: application/xml
Content-Length: 309
Connection: keep-alive
Date: Fri, 14 Apr 2019 07:17:19 GMT
Server: tencent-cos
```
<ReplicationConfiguration>
  <Role>qcs::cam::uin/100000000001:uin/100000000001</Role>
  <Rule>
    <Status>Enabled</Status>
    <ID>RuleId_01</ID>
    <Prefix>testPrefix</Prefix>
    <Destination>
      <Bucket>qcs::cos:ap-guangzhou::destinationbucket-1250000000</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
DELETE Bucket replication

The DELETE Bucket replication API is used to delete the cross-region replication configuration from a bucket. When initiating this request, you need to get the request signature to show that the request has been authorized.

Request

Sample Request

```
DELETE /?replication HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

Request Headers

Common Headers

The implementation of this request uses a common request header. For more information on common request headers, see Common Request Headers.

Special Headers

This request does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response contains a common response header. For more information on common response headers, see Common Response Headers.

Special Response Headers

This response does not have special response headers.

Response Body

This response body is empty.

Samples

Request

The following request sample deletes the cross-region replication configuration from the bucket `originbucket-1250000000`

```
DELETE /?replication HTTP/1.1
Date: Fri, 14 Apr 2019 07:47:35 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfbOAo7cllgPvF9cXFrJDBa1ICV88JM&o-sign-time=1503901499;1503901859&q-key-time=1503901499;1503901859&q-header-list=host&q-url-param-list=replication&q-signature=761f3f6440c6a11684464f4b0b6f292f0a4e7e0
Host: originbucket-1250000000.cos.ap-chengdu.myqcloud.com
```

Response
After the request above is made, COS returns a response of 204 No Content, indicating that the cross-region replication configuration has been successfully deleted from the bucket. After the configuration is deleted, COS will no longer replicate the objects in the source bucket to the destination bucket, and the existing object data in the destination bucket will be retained.
Log Management(logging)
PUT Bucket logging

Description

This API (PUT Bucket Logging) is used to enable logging for the source bucket and store its access logs in the specified destination bucket.

Only the source bucket owner can request this operation.

Request

Sample Request

PUT /?logging HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date:date
Content-Length: length
Content-Type: application/xml
Content-MD5: MD5
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this request operation requires the use of the Content-MD5 header to verify the integrity of the message, as shown below. For more information on common request headers, see Common Request Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-MD5</td>
<td>Base64-encoded 128-bit content MD5 checksum as defined in RFC 1864. This header is used to check whether the file content has changed</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Body

The implementation of this request operation requires a request body. An example of the content of a request body with all nodes is as follows:

```
<BucketLoggingStatus>
<LoggingEnabled>
<TargetBucket>logbucket</TargetBucket>
<TargetPrefix>mylogs</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

The specific data descriptions are as follows:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
</table>

©2013-2019 Tencent Cloud. All rights reserved.
<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>BucketLoggingStatus</td>
<td>None</td>
<td>Indicates the status of the logging configuration. If there is no child node information, logging is disabled</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node BucketLoggingStatus:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoggingEnabled</td>
<td>BucketLoggingStatus</td>
<td>Specific information of the bucket logging setting, which is mainly the destination bucket</td>
<td>Container</td>
<td>No</td>
</tr>
</tbody>
</table>

Content of the Container node LoggingEnabled:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetBucket</td>
<td>LoggingEnabled</td>
<td>Destination bucket for storing logs</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>TargetPrefix</td>
<td>LoggingEnabled</td>
<td>Logs are stored in the specified path in the destination bucket</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Response

**Response Headers**

**Common Response Headers**
This response uses a common response header. For more information on common response headers, see Common Response Headers.

**Special Response Headers**
This response has no special response headers.

**Response Body**
The response body return is empty.

Use Cases

**Request**

```xml
PUT /?logging HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2017 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUfABC&q-sign-time=1484814927;32557718927&q-key-time=1484814927;32557718927&q-header-list=host&q-url-param-list=accelerate&q-signature=8b9f95dabece2578f3a7b0732396e7cbade9833e3
Content-Type: application/xml
Content-Length: 147

<BucketLoggingStatus>
```
Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 10 Mar 2017 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg4MDdiZWRfOWExZjRkZzQ2OWNfZG==
GET Bucket logging

Last updated: 2019-09-16 16:30:11

Description

This API (GET Bucket logging) is used to get the log configuration information of the source bucket.

Only the source bucket owner can request this operation.

Request

Sample Request

GET /?logging HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers
The implementation of this request uses a common request header. For more information on common request headers, see Common Request Headers.

Non-common Header
This request operation has no special request headers.

Request Body
The request body of this request is empty.

Response

Response Headers

Common Response Headers
This response uses a common response header. For more information about the common response header, see Common Response Headers.

Special Response Header
This response has no special response headers.

Response Body
The return of this response body is application/xml data. Below is an example containing complete node data:

```xml
<BucketLoggingStatus>
  <LoggingEnabled>
    <TargetBucket>logs</TargetBucket>
    <TargetPrefix>/logdir</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

The specific data content is as follows:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGGING_ENABLED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET_BUCKET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET_PREFIX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGGING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>BucketLoggingStatus</td>
<td>None</td>
<td>Bucket Log Status Information</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node BucketLoggingStatus:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoggingEnabled</td>
<td>BucketLoggingStatus</td>
<td>Bucket Log Configuration Details</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node LoggingEnabled:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetBucket</td>
<td>LoggingEnabled</td>
<td>The destination bucket for storing logs, which can be the same bucket (not recommended) or a bucket in the same region under the same account</td>
<td>String</td>
</tr>
<tr>
<td>TargetPrefix</td>
<td>LoggingEnabled</td>
<td>Specified path to the destination bucket for storing logs</td>
<td>String</td>
</tr>
</tbody>
</table>

Use Cases

Request

```
GET /?logging HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 28 Oct 2017 21:32:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9&q-sign-time=1484815944;32557711944&q-key-time=1484815944;32557711944&q-header-list=host&q-url-param-list=accelerate&q-signature=a2d28e1b90253d89f9277982775a4b3b705e023e
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 142
Connection: keep-alive
Date: Wed, 28 Oct 2017 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg4MDdlNGZfNDYyMDRfZT==

<BucketLoggingStatus>
  <LoggingEnabled>
    <TargetBucket>logs</TargetBucket>
    <TargetPrefix>logdir</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```
Bucket Encryption (encryption)
PUT Bucket encryption

Description

This API (PUT Bucket encryption) is used to set the default encryption configuration for the specified bucket.

To call this API, you must have the PutBucketEncryption permission. By default, the bucket owner has direct permission to use this API and can grant such permission to other users.

Request

Sample request

PUT /?encryption HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Request parameters

This API does not use any request parameter.

Request headers

This API only uses common request headers. For more information, please see Common Request Headers.

Request body

You can use XML in the request body to set the default encryption configuration information for a bucket. The encryption configuration information mainly contains encryption items.

Below is the request body used to set SSE-COS:

```xml
<ServerSideEncryptionConfiguration>
  <Rule>
    <ApplySideEncryptionConfiguration>
      <SSEAlgorithm>AES256</SSEAlgorithm>
    </ApplySideEncryptionConfiguration>
  </Rule>
</ServerSideEncryptionConfiguration>
```

The specific elements are as follows:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerSideEncryptionConfig</td>
<td>None</td>
<td>This contains the default encryption configuration parameters</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Rules</td>
<td>ServerSideEncryptionConfig</td>
<td>Default server-side encryption configuration rules</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Element Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>ApplyServerSideEncryptionByDefault</td>
<td>Rules</td>
<td>Default configuration information of server-side encryption</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>SSEAlgorithm</td>
<td>ApplyServerSideEncryptionByDefault</td>
<td>Server-side encryption algorithm to be used. Enumerated value: AES256</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Response**

**Response headers**

This API only returns common response headers. For more information, please see [Common Response Headers](#).

**Response body**

The response body return of this request is empty.

**Error codes**

There are no special error messages for this API. For all error messages, please see [Error Codes](#).

**Use Cases**

**Request**

The following sample shows you how to set SSE-COS encryption configuration for the `examplebucket-1250000000` bucket.

```
GET /?encryption HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: signatureValue

<ServerSideEncryptionConfiguration>
  <Rule>
    <ApplyServerSideEncryptionConfiguration>
      <SSEAlgorithm>AES256</SSEAlgorithm>
    </ApplyServerSideEncryptionConfiguration>
  </Rule>
</ServerSideEncryptionConfiguration>
```

**Response**

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Date: Mon, 17 Jun 2019 08:37:36 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****
```
GET Bucket encryption

Last updated: 2019-12-23 18:42:29

Description

This API (GET Bucket encryption) is used to query the default encryption configuration of the specified bucket.

To call this API, you must have the `GetBucketEncryption` permission. By default, the bucket owner has direct permission to use this API and can grant such permission to other users.

Request

Sample request

```
GET /?encryption HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Request parameters

This API does not use any request parameter.

Request headers

This API only uses common request headers. For more information, please see Common Request Headers.

Request body

This API does not have a request body.

Response

Response headers

This API only returns common response headers. For more information, please see Common Response Headers.

Response body

The following response elements will be returned:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServerSideEncryptionConfiguration</td>
<td>None</td>
<td>This contains the default encryption configuration parameters</td>
<td>Container</td>
</tr>
<tr>
<td>Rules</td>
<td>ServerSideEncryptionConfiguration</td>
<td>Default server-side encryption configuration rule</td>
<td>Container</td>
</tr>
<tr>
<td>ApplyServerSideEncryptionByDefault</td>
<td>Rules</td>
<td>Default configuration information of server-side encryption</td>
<td>Container</td>
</tr>
<tr>
<td>Element Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>SSEAlgorithm</td>
<td>ApplyServerSideEncryptionByDefault</td>
<td>Server-side encryption algorithm to be used. Enumerated value: AES256</td>
<td>String</td>
</tr>
</tbody>
</table>

**Error codes**

There are no special error messages for this API. For all error messages, please see Error Codes.

**Use Cases**

**Request**

GET /?encryption HTTP 1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: signatureValue

**Response**

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: xxxx
Date: Mon, 17 Jun 2019 08:37:36 GMT
Server: tencent-cos
x-cos-request-id: NWQwNzUxNTBfMzdiMDJhMDlfOWM0Nl85NDFk****

```xml
<?xml version = "1.0" encoding = "UTF-8">
<ServerSideEncryptionConfiguration>
  <Rule>
    <ApplyServerSideEncryptionConfiguration>
      <SSEAlgorithm>AES256</SSEAlgorithm>
    </ApplyServerSideEncryptionConfiguration>
  </Rule>
</ServerSideEncryptionConfiguration>
```
DELETE Bucket encryption

Description

This API (DELETE Bucket encryption) is used to delete the default encryption configuration of the specified bucket.

To call this API, you must have the `DeleteBucketEncryption` permission. By default, the bucket owner has direct permission to use this API and can grant such permission to other users.

Request

Sample request

```
DELETE /?encryption HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Request parameters

This API does not use any request parameter.

Request headers

This API only uses common request headers. For more information, please see Common Request Headers.

Request body

The request body of this request is empty.

Response

Response headers

This API only returns common response headers. For more information, please see Common Response Headers.

Response body

The response body return of this request is empty.

Error codes

There are no special error messages for this API. For all error messages, please see Error Codes.

Use Cases

Request

The following example shows you how to delete the default SSE-COS encryption configuration of the examplebucket-125000000 bucket.

```
DELETE /?encryption HTTP 1.1
Host: examplebucket-125000000.cos.ap-beijing.myqcloud.com
Date: Mon, 17 Jun 2019 08:37:35 GMT
Authorization: signatureValue
```

Response

```
HTTP/1.1 204 No Content
Server: tencent-cos
Date: Mon, 17 Jun 2019 08:37:36 GMT
x-cos-request-id: NNQwNhz0xXr8NzdINoJihMDg1WW0WNLBSNFk****
```
Global Acceleration (Accelerate)
PUT Bucket Accelerate
Last updated: 2020-02-21 12:15:04

Feature description

This API (PUT Bucket Accelerate) is used to enable or suspend global acceleration for the specified bucket.

Detail analysis

1. If you have never enabled global acceleration for the bucket, the request to the GET Bucket Accelerate API will not return global acceleration configuration status.
2. Once enabled, global acceleration can only be suspended but cannot be disabled.
3. Valid values of the global acceleration configuration status are Enabled and Suspended, indicating that global acceleration is enabled or suspended, respectively.
4. If you are using a sub-account, in order to set the global acceleration feature for the bucket, you should have permission to write the configuration.

Request

Sample request

```
PUT /?accelerate HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

Request header

This API only uses common request headers. For more information, please see Common Request Headers.

Request body

```
<AccelerateConfiguration xmlns="cos xmlns/">
  <Status>Enabled</Status>
</AccelerateConfiguration>
```

Detailed data is as shown below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccelerateConfiguration</td>
<td>None</td>
<td>Detailed information of global acceleration</td>
<td>Container</td>
</tr>
<tr>
<td>Status</td>
<td>AccelerateConfiguration</td>
<td>Indicates whether global acceleration is enabled. Enumerated values: Suspended, Enabled</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Response

Response header

This API only returns common response headers. For more information, please see Common Response Headers.

Response body

The response body return is empty.
Error codes
The following error messages may be returned for this request operation. For common error messages, please see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>400 Bad Request</td>
<td>1. If the XML body of the request is empty, InvalidArgument will be returned. 2. The global acceleration status has only two valid values: Enabled and Suspended. If any other status value is entered, InvalidArgument will be returned.</td>
</tr>
<tr>
<td>InvalidDigest</td>
<td>400 Bad Request</td>
<td>The carried Content-MD5 does not match the request body calculated by the server.</td>
</tr>
</tbody>
</table>

Use cases

Request

PUT /?accelerate HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Authorization: authorization string
Content-Type: text/plain
Content-Length: 83

<AccelerateConfiguration>
  <Status>Enabled</Status>
</AccelerateConfiguration>

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0ZThm
GET Bucket Accelerate

Feature description

This API (GET Bucket Accelerate) is used to query the global acceleration configuration of the specified bucket.

Detail analysis

1. If you have never enabled global acceleration for the bucket, the request to the GET Bucket Accelerate API will not return global acceleration configuration status.
2. Valid return values of the global acceleration configuration status are `Enabled` and `Suspended`, indicating that global acceleration is enabled or suspended, respectively.
3. If you are using a sub-account, in order to query the global acceleration configuration information of the bucket, you should have permission to read the configuration.

Request

Sample request

```
GET /Accelerate HTTP 1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT date
Authorization: Auth String
```

Request header

This API only uses common request headers. For more information, please see Common Request Headers.

Request body

The request body of this request is empty.

Response

Response header

This API only returns common response headers. For more information, please see Common Response Headers.

Response body

```
<AccelerateConfiguration xmlns="cos xmlns:">
  <Status>Enabled</Status>
  <Type>COS</Type>
</AccelerateConfiguration>
```

Detailed data is as shown below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccelerateConfiguration</td>
<td>None</td>
<td>Detailed information of global acceleration</td>
<td>Container</td>
</tr>
<tr>
<td>Status</td>
<td>AccelerateConfiguration</td>
<td>Indicates whether global acceleration is enabled. Enumerated values: Suspended, Enabled</td>
<td>Enum</td>
</tr>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------</td>
<td>------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Type</td>
<td>AccelerateConfig</td>
<td>Global acceleration type. Enumerated value: COS</td>
<td>Enum</td>
</tr>
</tbody>
</table>

**Error codes**

There are no special error messages for this API. For all error messages, please see Error Codes.

**Use cases**

**Request**

GET /?accelerate HTTP/1.1
Host: examplebucket-1250000000.cos.ap-chengdu.myqcloud.com
Connection: keep-alive
Authorization: authorization string
Content-Type: text/plain

**Response 1 (global acceleration enabled)**

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 73
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0ZThm

<AccelerateConfiguration>
  <Status>Enabled</Status>
  <Type>COS</Type>
</AccelerateConfiguration>

**Response 2 (global acceleration suspended)**

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 73
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0ZThm

<AccelerateConfiguration>
  <Status>Disabled</Status>
  <Type>COS</Type>
</AccelerateConfiguration>

**Response 3 (global acceleration not enabled)**

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 73
Connection: keep-alive
Date: Wed, 23 Aug 2019 08:14:53 GMT
Server: tencent-cos
x-cos-request-id: NTk5ZDM5N2RfMjNiMjM1MGFfMmRiX2Y0ZThm

<AccelerateConfiguration/>
Object APIs
Basic Operations
PUT Object

Last updated: 2019-12-16 17:54:13

Description

This API is used to upload a local object to a specified bucket. To make this request, you need to have the permission to write to the bucket.

- If the `Content-Length` value in the request header is smaller than the length of the data in the actual request body, COS will still successfully create a file, but the object size will equal the size defined in `Content-Length`, and the remaining data will be discarded.
- If there is an object in the bucket with the same name as the object to be uploaded, and versioning is not enabled, the old object will be overwritten by the new one and `200 OK` will be returned upon success.

Versioning

- If versioning is enabled for the bucket, COS will automatically generate a unique version ID for the object to be uploaded and return this ID in the response using the `x-cos-version-id` response header.
- If versioning is suspended for the bucket, COS will always use `null` as the version ID of the objects in the bucket and will not return the `x-cos-version-id` response header.

Request

Sample Request

```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Type: Content Type
Content-Length: Content Length
Content-MD5: MD5
Authorization: Auth String

[Object Content]
```

Authorization: Auth String (see Request Signature for details).

Request Parameters

This API does not use any request parameter.

Request Headers

In addition to common request headers, this API also supports the following request headers. For more information on common request headers, see Common Request Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Cache directives as defined in RFC 2616, which will be stored in the object metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td>Filename as defined in RFC 2616, which will be stored in the object metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>Encoding format as defined in RFC 2616, which will be stored in the object metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>Expires</td>
<td>Cache expiration time as defined in RFC 2616, which will be stored in the object metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Transfer-Encoding</td>
<td>If you want to upload the object in parts, you need to specify the Transfer-Encoding: chunked request header. In such a case, the request body must follow the transfer encoding format as defined in RFC 2616 and you cannot specify the Content-Length request header</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-meta-*</td>
<td>Header suffix and information of the user-defined metadata, which will be stored in the object metadata; maximum size: 2 KB. <strong>Note:</strong> User-defined metadata information can contain underscores (_), whereas the header suffixes of user-defined metadata can only contain minus signs (-), not underscores</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-storage-class</td>
<td>Object storage class, such as STANDARD_IA and ARCHIVE. Default value: STANDARD. For enumerated values, see Storage Class</td>
<td>Enum</td>
<td>No</td>
</tr>
</tbody>
</table>

**ACL-related headers**

You can configure access permissions for the object by specifying the following request headers when uploading it:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-acl</td>
<td>Defines the access control list (ACL) attribute of the object. For the enumerated values such as default, private, and public-read, see the Preset ACL section in ACL Overview. Default value: default <strong>Note:</strong> Currently, there can be up to 1,000 entries in one ACL. If you do not need access control for the object, set default for this parameter or simply leave it blank, and the object will inherit the permissions of the bucket</td>
<td>Enum</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read</td>
<td>Allows grantee to read the object; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read-acp</td>
<td>Allows grantee to read the ACL of the object; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-write-acp</td>
<td>Allows grantee to write to the ACL of the object; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-full-control</td>
<td>Grants a user full permission to perform operations on the object; format: id=&quot;[OwnerUin]&quot; , such as id=&quot;100000000001&quot; . You can use comma (,) to separate multiple users, such as id=&quot;100000000001&quot;, id=&quot;100000000002&quot;</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**Server-side encryption-related headers**

You can use server-side encryption when uploading an object. For more information, see Server-side Encryption Headers.

**Request Body**

The request body of this API request is the object (file) content.

**Response**

**Response Headers**

This API only returns common response headers. For more information, see Common Response Headers.

**Versioning-related headers**

When the object is uploaded to a bucket where versioning is enabled, the following response headers will be returned:
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-version-id</td>
<td>Object version ID</td>
<td>string</td>
</tr>
</tbody>
</table>

**Server-side encryption-related headers**

If server-side encryption is used when the object is uploaded, this API will return the server-side encryption headers. For more information, see Server-side Encryption Headers.

**Response Body**

The response body of this API is empty.

**Error Codes**

There is no special error message for this API. For all error messages, see Error Codes.

**Examples**

**Example 1. Basic example (versioning not enabled)**

**Request**

```plaintext
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 21 Jun 2019 09:24:28 GMT
Content-Type: image/jpeg
Content-Length: 13
Content-MD5: ti4QvKtVqIJAvZxDbP/crQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1561109068;1561116268&q-key-time=15611109068;1561116268&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=&q-signature=998bfc8836fc285d90e455c14e3d7e623bd2****
Connection: close

[Object Content]
```

**Response**

```plaintext
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 21 Jun 2019 09:24:28 GMT
ETag: "b62e10bcab55a88240b9c436cfdfc9"
Server: tencent-cos
x-cos-request-id: NWQwY2EyNGNfYThjMDBiMDlfMTA0ZmVfYTJm****
```

**Example 2. Specifying metadata and ACL using request headers**

**Request**

```plaintext
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 21 Jun 2019 09:24:31 GMT
Content-Type: image/jpeg
Cache-Control: max-age=86400
Content-Disposition: attachment; filename=example.jpg
x-cos-meta-example-field: example-value
x-cos-acl: public-read
Content-Length: 13
Content-MD5: ti4QvKtVqIJAvZxDbP/crQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1561109071;1561116271&q-key-time=15611109068;1561116268&q-header-list=cache-control;content-disposition;content-length;content-md5;content-type;date;host;x-cos-acl;x-cos-meta-example-field&q-url-param-list=&q-signature=da483c6b1c259d142a128ba0e6d957811df****
Connection: close

[Object Content]
```

**Response**

```plaintext
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 21 Jun 2019 09:24:28 GMT
ETag: "b62e10bcab55a88240b9c436cfdfc9"
Server: tencent-cos
x-cos-request-id: NWQwY2EyNGNfYThjMDBiMDlfMTA0ZmVfYTJm****
```
Example 3. Using server-side encryption SSE-COS

Request

```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 21 Jun 2019 09:24:35 GMT
Content-Type: image/jpeg
x-cos-server-side-encryption: AES256
Content-Length: 13
Content-MD5: ti4QvKtVqIJAvZxDbP/c+Q==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1561109075;1561116275&q-key-time=1561116275&q-header-list=content-length;content-md5;content-type;date;host;x-cos-server-side-encryption&q-url-param-list=&q-signature=3e21f7ba7e0d5c7f3aee7ff139753b24a****
Connection: close

[Object Content]
```

Response

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 21 Jun 2019 09:24:35 GMT
ETag: "b62e10bcab55a88240bd9c436cffdcf9"
Server: tencent-cos
x-cos-request-id: MNQwY2EyNg2fN0V1JMTJhMD1lTyaXN185MjA2****
```

Example 4. Using server-side encryption SSE-C

Request

```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 21 Jun 2019 09:24:38 GMT
Content-Type: image/jpeg
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: MDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7 frmUG8==
Content-Length: 13
Content-MD5: ti4QvKtVqIJAvZxDbP/c+Q==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1561109078;1561116278&q-key-time=1561116278&q-header-list=content-length;content-md5;content-type;date;host;x-cos-server-side-encryption-customer-algorithm;x-cos-server-side-encryption-customer-key;x-cos-server-side-encryption-customer-key-md5&q-url-param-list=&q-signature=d84a5d70a5f08c7db4f89a91628a7eacf90****
Connection: close

[Object Content]
```

Response

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 21 Jun 2019 09:24:38 GMT
ETag: "492b458ec33eaf0a824a7d6d4803b3"
Server: tencent-cos
x-cos-request-id: MNQwY2EyNg2fN0V1JMTJhMD1lTyaXN185MjA2****
```
Example 5. With versioning enabled

Request

```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 21 Jun 2019 09:24:45 GMT
Content-Type: image/jpeg
Content-Length: 13
Content-MD5: ti4QvKtVqIJAvZxDbP/cQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1561109085;1561116285&q-key-time=1561109085;1561116285&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=&q-signature=28c8b3f8f887cab343124b2330a0280486ef****
Connection: close

[Object Content]
```

Response

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Fri, 21 Jun 2019 09:24:45 GMT
ETag: "b62e10bcab55a88240bdfc36cfdcf9"
Server: tencent-cos
x-cos-request-id: NWQwY2EyNWRfYThjMDBiMDlfMTA1MDlfYTQ1****
x-cos-version-id: MTg0NDUxODI5NjQ2MjM5OTMyNzM
```

Example 6. With versioning suspended

Request

```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 24 Jul 2019 02:51:28 GMT
Content-Type: image/jpeg
Content-Length: 13
Content-MD5: ti4QvKtVqIJAvZxDbP/cQ==
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1563936688;1563943888&q-key-time=1563936688;1563943888&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=&q-signature=aab4bfeb62a7d86725ad6b4a086decba1****
Connection: close

[Object Content]
```

Response

```
HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Wed, 24 Jul 2019 02:51:28 GMT
ETag: "b62e10bcab55a88240bdfc36cfdcf9"
Server: tencent-cos
x-cos-request-id: NWQzN2M3YjBfN2ViMTJhMDlfYTkxMl9iY2Fj****
x-cos-version-id: MTg0NDUxODI5NjU2MjU5OTMyNzW
```

Example 7. Using the chunked transfer encoding for multipart transfer

Request

```
PUT /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 08 Aug 2019 09:15:29 GMT
Content-Type: text/plain
Transfer-Encoding: chunked
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3JO****&q-sign-time=1565255729;1565262929&q-key-time=1565255729;1565262929&q-header-list=content-type;date;host;transfer-encoding&q-url-param-list=&q-signature=0b8f566ba75afbc159ca88a4e4051ec6939****

[Object Content]
```
Response

HTTP/1.1 200 OK
Content-Length: 0
Connection: close
Date: Thu, 08 Aug 2019 09:15:29 GMT
ETag: "aa488b8818a6be87f4a7b936a80752"
Server: tencent-cos
x-cos-request-id: NWQ0YmU4MzFfNzFiNDBiMDlfMWJhYTlfMTY2Njll****
PUT Object - Copy

Description

This API (PUT Object - Copy) is used to create a copy of an object that already exists in COS, i.e., copying an object from the source path (object key) to the destination path (object key). The recommended object size is from 1 MB to 5 GB. For objects over 5 GB, please use the Upload Part - Copy API. You can modify the object metadata and access control list (ACL) during the copying process.

You can use this API to move, rename, and copy an object and modify its metadata.

- For cross-account copy, you need to set the access permissions of the source object to Public Read or authorize the destination account to read the source object. This is not required for intra-account copy.
- An error may be returned when COS receives the copy request or is copying the object. If an error occurs before the copy operation begins, a standard error response will be returned. If an error occurs during the execution of the copy operation, HTTP 200 OK will be returned with the error as the response body. This means that the HTTP 200 OK response can mean both success and error. When you use this API, please pay attention to the content of the response body to determine if the copy request is successful and deal with the result accordingly.

Version

If versioning is enabled for the destination bucket, by default COS will generate a unique version ID for the object copy, which will be different from the version ID of the source object and will be returned in the x-cos-version-id response header.

If versioning is suspended or not enabled for the destination bucket, the version ID generated by COS will always be `null`.

Request

Sample Request

```plaintext
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
x-cos-copy-source: <BucketName-APPID>.cos.<Region>.myqcloud.com/filepath
```

Authorization: Auth String (see Request Signature for details).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-copy-source</td>
<td>URL path to the source file. You can specify a historical version using the <code>versionid</code> subresource</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-metadata-directive</td>
<td>Whether to copy the metadata of the source file. Enumerated values: <code>Copy</code>, <code>Copy</code> the metadata of the source file; <code>Replaced</code>, modify the metadata according to the header information in the current request. Default value: <code>Copy</code>. If the destination path is the same as the source path, which means you want to modify the object metadata, you must specify this header as <code>Replaced</code></td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-copy-source-If-Modified-Since</td>
<td>If the object is modified after the specified time, the operation will be performed; otherwise, <code>412</code> will be returned. This header can be used</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>--------------</td>
</tr>
<tr>
<td>x-cos-copy-source-If-Unmodified-Since</td>
<td>If the object is not modified after the specified time, the operation will be performed; otherwise, '412' will be returned. This header can be used together with 'x-cos-copy-source-If-Match'. If it is used together with other conditions, a conflict will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-copy-source-If-Match</td>
<td>If the 'Etag' of the object is the same as the specified one, the operation will be performed; otherwise, '412' will be returned. This header can be used together with 'x-cos-copy-source-If-Unmodified-Since'. If it is used together with other conditions, a conflict will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-copy-source-If-None-Match</td>
<td>If the 'Etag' of the object is different from the specified one, the operation will be performed; otherwise, '412' will be returned. This header can be used together with 'x-cos-copy-source-If-Modified-Since'. If it is used together with other conditions, a conflict will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-storage-class</td>
<td>Sets the storage class of the object. Enumerated values: <code>STANDARD</code>, <code>STANDARD_IA</code>. Default value: <code>STANDARD</code></td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-acl</td>
<td>Defines the ACL attribute of the object. Valid values: <code>private</code>, <code>public-read</code>. Default value: <code>private</code></td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read</td>
<td>Grants the grantee Read access. Format: <code>x-cos-grant-read: id=&quot;[OwnerUin]&quot;</code></td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-write</td>
<td>Grants the grantee Write access. Format: <code>x-cos-grant-write: id=&quot;[OwnerUin]&quot;</code></td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-full-control</td>
<td>Grants the grantee full permission. Format: <code>x-cos-grant-full-control: id=&quot;[OwnerUin]&quot;</code></td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-meta-*</td>
<td>Suffix and information of the user-defined header, which will be returned as the object metadata; maximum size: 2 KB. <strong>Note: user-defined header information can contain underscores, whereas user-defined header suffixes cannot</strong></td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

**Server-side encryption-related headers**

This request operation specifies the data encryption policy for COS data storage. COS will automatically encrypt the data written to the IDC and automatically decrypt it when you retrieve it. Currently, AES-256 encryption with the COS master key is supported. If you want to enable server-side encryption for your data, please pass in the following header:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption</td>
<td>Specifies how the server-side encryption is enabled for the object. For encryption with the COS master key, pass in AES256</td>
<td>String</td>
<td>Yes if encryption is needed</td>
</tr>
</tbody>
</table>

**Request Body**

The request body of this request is empty.

**Response**

**Response Headers**

**Common Response Headers**

This response uses common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-version-id</td>
<td>Version of the object copy in the destination bucket. This parameter will be returned only if versioning is enabled or enabled and then suspended for the bucket</td>
<td>String</td>
</tr>
</tbody>
</table>
If the object is stored with COS-managed server-side encryption, the response will contain this header and the value of the encryption algorithm used (AES256).

Response Body

This response body returns `application/xml` data. The following contains all the node data:

```xml
<CopyObjectResult>
  <ETag>"ba82b57cfdfda8bd17ad4e587f9ebf4fa"</ETag>
  <LastModified>2017-08-04T02:41:45Z</LastModified>
</CopyObjectResult>
```

Please find the details below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyObjectResult</td>
<td>Returns the result of the copy operation</td>
<td>String</td>
</tr>
<tr>
<td>ETag</td>
<td>Returns the MD5 checksum of the file. The value of ETag can be used to check whether the object content has changed</td>
<td>String</td>
</tr>
<tr>
<td>LastModified</td>
<td>Returns the time in GMT time when the file is last modified</td>
<td>String</td>
</tr>
</tbody>
</table>

Example

**Request**

```
PUT /exampleobject HTTP/1.1
Host: destinationbucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 04 Aug 2017 02:41:45 GMT
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15IsskiBQKTZbAo6WhgcBqVls9SmuG00&q-sign-time=1480932292;1981012292&q-key-time=1480932292;1981012292&q-url-param-list=&q-header-list=host&q-signature=eacefe8e2a0dc8a1674d9a29787b16aba5a47cc
x-cos-copy-source: sourcebucket-1250000001.cos.ap-beijing.myqcloud.com/picture.jpg
Content-Length: 0
```

**Response**

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 133
Connection: keep-alive
Date: Fri, 04 Aug 2017 02:41:45 GMT
Server: tencent-cos
x-cos-request-id: NTk4M2RlZTlfZDRiMDM1MGFfYTA1ZV8xMzNlYw==

<CopyObjectResult>
  <ETag>"ba82b57cfdfda8bd17ad4e587f9ebf4fa"</ETag>
  <LastModified>2017-08-04T02:41:45Z</LastModified>
</CopyObjectResult>
```
POST Object

Description

This API is used to upload a file (object) to a specified bucket using forms. To make this request, you need to have the permission to write to the bucket. All the API parameters carried by HTTP headers are requested using form fields.

Version

If versioning is enabled for the bucket, the POST operation will automatically generate a unique version ID for the object to be uploaded. COS will return this ID in the response using the `x-cos-version-id` response header.

If you suspend versioning for the bucket, COS will always use `null` as the version ID of the objects in the bucket.

Notes

1. You need to have the permission to write to the bucket.
2. If there is an object in the bucket with the same name as the object to be uploaded, the old object will be overwritten by the new one and the request will return success upon a successful upload.

Request

Sample Request

```
POST / HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Content-Length: length

Headers

Form
```

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

The following request headers are required for the implementation of this operation:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Length</td>
<td>HTTP request length in bytes as defined in RFC 2616</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Form Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>Defines the ACL attribute of the object. Valid values: <code>private</code>, <code>public-read</code>, and <code>default</code> . Default value: <code>default</code> (i.e., inheriting the bucket's permission). Note: currently, there can be up to 1,000 entries in one ACL. If you do not need access control for the object, set <code>default</code> for this parameter or simply leave it blank, and the object will inherit the permissions of the bucket</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires</td>
<td>Headers as defined in RFC 2616. For more information, see PUT Object</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>file</td>
<td>File content, as the last field in the form</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>key</td>
<td>Filename after the file is uploaded, which will be changed if you use ${filename} ; for example, if you use a/b/${filename} and upload a file</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>
If this field is specified, it will take the priority; '303' and the 'Location' header will be returned, and `bucket=bucket&key=key&etag=etag` parameter will be added to the end of the URL.

**success_action_status**

Valid values: '200', '201', and '204'. '204' is returned by default. If 'success_action_redirect' is set, this field will be ignored.

**x-cos-meta-**

Suffix and information of the user-defined header, which will be returned as the object metadata; maximum size: 2 KB. Note: user-defined header information can contain underscores, whereas user-defined header suffixes cannot.

**x-cos-storage-class**

Specifies the storage class of the object. Enumerated values: 'STANDARD', 'STANDARD_IA', and 'ARCHIVE'. Default value: 'STANDARD'.

**policy**

Base64-encoded and used for request verification. If the request content does not match the conditions specified by the field, '403 Access Denied' will be returned.

**x-cos-server-side-encryption**

Specifies how the server-side encryption is enabled for the object. For encryption with a COS master key, enter AES256. Yes if encryption is needed.

**Signature Protection**

If signature protection is needed for a **POST** request to upload an object using forms, the forms must contain the content of the following form-data:

<table>
<thead>
<tr>
<th>Form Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>Base64-encoded policy content, which is used to verify the request content. If the request content does not match the conditions specified by the policy, the request will be denied.</td>
</tr>
<tr>
<td>q-sign-algorithm</td>
<td>Algorithm for signature calculation. COS currently supports SHA1. Enter <code>sha1</code> (in lowercase) for this field.</td>
</tr>
<tr>
<td>q-ak</td>
<td>Your SecretId in Tencent Cloud.</td>
</tr>
<tr>
<td>q-key-time</td>
<td>Start and end of the validity period of the key used to request the signature, which are described in Unix timestamps; unit: second. Format: <code>start-seconds;end-seconds</code>, such as <code>1480932292;1481012298</code>.</td>
</tr>
<tr>
<td>q-signature</td>
<td>Request signature calculated using the fields above. COS will verify the signature against the form fields. If they do not match, the request will be denied.</td>
</tr>
</tbody>
</table>

**Signature Calculation**

The signature **q-signature** is calculated in three steps:

1. Use the key content to encrypt `q-key-time` to get `SignKey`.  
2. Create a **POST** request policy and encrypt its content with `sha1` to get `StringToSign`.  
3. Encrypt `StringToSign` with `SignKey` to generate the signature.

**Policy**

Below is a sample of a complete policy:

```json
{
  "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"acl": "public-read"},
    {"bucket": "examplebucket-1250000000"},
    {"starts-with", "$key", "user/eric/"},
    {"q-sign-algorithm": "sha1"},
    {"q-ak": "AKIDOiz3ItomoVJbn51L.tikXMFEPwko0b5q"},
    {"q-sign-time": "1480932292;1481012298"}
  ]
}
```

**Expiration**

Sets the expiration time of the **POST** policy in ISO8601 GMT time, such as `2017-12-01T12:00:00.000Z`.  

©2013-2019 Tencent Cloud. All rights reserved.  
Page 158 of 259
### Conditions Rules

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full match</td>
<td>Format: (&quot;key&quot;: &quot;value&quot;) or [&quot;eq&quot;, &quot;$key&quot;, &quot;value&quot;]</td>
</tr>
<tr>
<td>Prefix match</td>
<td>Format: [&quot;starts-with&quot;, &quot;$key&quot;, &quot;value&quot;], where value can be left empty</td>
</tr>
<tr>
<td>Range match</td>
<td>Format: [&quot;content-length-range&quot;, int1, int2], which means that the number of file bytes must be between int1 and int2</td>
</tr>
</tbody>
</table>

### Conditions Parameters

All parameters are optional. If they are left empty, the verification can be skipped.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Matching Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>Allowed range of the file's ACL attribute; optional</td>
<td>Full, prefix</td>
</tr>
<tr>
<td>bucket</td>
<td>Specifies the destination bucket</td>
<td>Full</td>
</tr>
<tr>
<td>content-length-range</td>
<td>Specifies the size range of the file to be uploaded</td>
<td>Range</td>
</tr>
<tr>
<td>key</td>
<td>Object storage path</td>
<td>Full, prefix</td>
</tr>
<tr>
<td>success_action_redirect</td>
<td>URL returned after a successful upload</td>
<td>Full, prefix</td>
</tr>
<tr>
<td>success_action_status</td>
<td>Status returned after a successful upload</td>
<td>Full</td>
</tr>
<tr>
<td>x-cos-meta-*</td>
<td>Suffix and information of the user-defined header, which will be returned as the object metadata; maximum size: 2 KB. Note: user-defined header information can contain underscores, whereas user-defined header suffixes cannot</td>
<td>Full, prefix</td>
</tr>
<tr>
<td>x-cos-*</td>
<td>Other COS headers that need to be signed</td>
<td>Full</td>
</tr>
</tbody>
</table>

### Response

#### Response Headers

##### Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

##### Special Response Headers

This request may return the following response headers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-version-id</td>
<td>Version of the copied object in the destination bucket. This header will be returned only if versioning is enabled or enabled and then suspended for the bucket</td>
<td>String</td>
</tr>
<tr>
<td>x-cos-server-side-encryption</td>
<td>If the object is stored with COS-managed server-side encryption, the response will contain this header and the value of the encryption algorithm used (AES256)</td>
<td>string</td>
</tr>
</tbody>
</table>

#### Response Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETag</td>
<td>Returns the MDS checksum of the file. The value of ETag can be used to check whether the object gets corrupted during the upload</td>
<td>String</td>
</tr>
<tr>
<td>Location</td>
<td>If success_action_redirect is specified, the corresponding value will be returned; otherwise, the full path to the object will be returned</td>
<td>String</td>
</tr>
</tbody>
</table>

#### Response Body
This response body returns application/xml data. The following contains all the node data:

```xml
<PostResponse>
  <Location>http://examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/photo.jpg</Location>
  <Bucket>examplebucket-1250000000</Bucket>
  <Key>photo.jpg</Key>
  <ETag>d41d8cd98f00b204e9800998ecf8427e</ETag>
</PostResponse>
```

Below are the details:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostResponse</td>
<td>None</td>
<td>Container storing the response of the POST Object request</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node PostResponse:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>PostResponse</td>
<td>Full path to the object</td>
<td>String</td>
</tr>
<tr>
<td>Bucket</td>
<td>PostResponse</td>
<td>Bucket where the object is stored</td>
<td>String</td>
</tr>
<tr>
<td>Key</td>
<td>PostResponse</td>
<td>Object key name</td>
<td>String</td>
</tr>
<tr>
<td>ETag</td>
<td>PostResponse</td>
<td>Etag content</td>
<td>String</td>
</tr>
</tbody>
</table>

**Error Codes**

The following describes some frequent special errors that may occur when you make this request. For more COS error codes, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidDigest</td>
<td>400 Bad Request</td>
<td>If the request carries the Content-MD5 header when the file is uploaded, COS will check whether the MD5 of the body is the same as the one carried, and if not, InvalidDigest will be returned</td>
</tr>
<tr>
<td>KeyTooLong</td>
<td>400 Bad Request</td>
<td>If a user-defined header beginning with x-cos-meta is carried in the request when the file is uploaded, the total size of the key and value of the header cannot exceed 4 KB; otherwise, the KeyTooLong error will be returned</td>
</tr>
<tr>
<td>MissingContentLength</td>
<td>411 Length Required</td>
<td>If the Content-Length header is not carried in the request when the file is uploaded, this error code will be returned</td>
</tr>
<tr>
<td>NoSuchBucket</td>
<td>404 Not Found</td>
<td>If the bucket to which you want to upload the object does not exist, the NoSuchBucket (404 Not Found) error will be returned</td>
</tr>
<tr>
<td>EntityTooLarge</td>
<td>400 Bad Request</td>
<td>If the file to be uploaded is larger than 5 GB, the EntityTooLarge error will be returned with the error message Your proposed upload exceeds the maximum allowed object size</td>
</tr>
<tr>
<td>InvalidURI</td>
<td>400 Bad Request</td>
<td>The object key cannot exceed 850 bytes in length; otherwise, the InvalidURI error will be returned</td>
</tr>
</tbody>
</table>
Example

Request

```
POST / HTTP/1.1
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 1352
Content-Type: multipart/form-data; boundary=e07f2a7876ea4755ae18d300807ad879

--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="key"
photo.jpg
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="success_action_status"
201
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="Acl"
public-read
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="x-cos-storage-class"
STANDARD
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="Signature"
q-sign-algorithm=sha1&ak=AKIDZfbOAo7cllgPvF9cXFrJD0a1ICvR98LJM&q-sign-time=1512983814;1512984814&q-key-time=1512983814;1512984814&q-url-param-list=&q-header-list=host&q-signature=2ff2daee714e47445a88a000ec5d51771ff5856500
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="policy"
eeyJjb25kaXRpb25zIjogW3siYnVja2V0IjogImtpdG1hbnMzdGVzdDEifSwgWyJjb250ZW50LWxlbmd0aC1yYW5nZSIwIiwgIjEyIl1dLCAiZXhwaXJhdGlvbiI6ICIyMDQ3LTEyLTAxVDEyOjAwOjAwLjAwMFoifQ==
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="x-Cos-meta-bb"
124
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="key1"
1
--e07f2a7876ea4755ae18d300807ad879
Content-Disposition: form-data; name="file"; filename="empty:a"

--e07f2a7876ea4755ae18d300807ad879--
```

Response

```
HTTP/1.1 204
Content-Type: application/xml
Content-Length: 232
Connection: keep-alive
Date: Mon, 11 Dec 2017 09:16:56 GMT
ETag: "d41d8cd98f00b204e9800998ecf8427e"
Location: http://examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com/photo.jpg
Server: tencent-cos
x-cos-request-id: MWEyZTQwMDZfMjQ0G0Y3MGFyNTE4Y1B1
```
GET Object

Last updated : 2019-12-16 17:55:33

Description

This API is used to download an object in a COS bucket to a local file system. To make this request, you need to have Read access to the target object or the target object allows Public Read.

If the `response-*` request parameter is used, this request operation will not support anonymous request and must carry a signature.

Versioning

If versioning is enabled, the GET operation will return the latest version of the object. If you want to download a historical version of the object, please specify the `versionId` request parameter.

Request

Sample Request

```
GET /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

Request Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>response-cache-control</td>
<td>Value of the Cache-Control header in the response</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>response-content-disposition</td>
<td>Value of the Content-Disposition header in the response</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>response-content-encoding</td>
<td>Value of the Content-Encoding header in the response</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>response-content-language</td>
<td>Value of the Content-Language header in the response</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>response-content-type</td>
<td>Value of the Content-Type header in the response</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>response-expires</td>
<td>Value of the Expires header in the response</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>versionId</td>
<td>Version ID of the object to be downloaded</td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Headers

In addition to common request headers, this API also supports the following request headers. For more information on common request headers, see Common Request Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>Byte range as defined in RFC 2616. The range value must be in the format of bytes=first-last, where both first and last are offsets starting from 0. For example, bytes=0-9 means downloading the first 10-byte data of the object, and HTTP status code 206 (Partial Content) and the Content-Range response header will be returned. If this parameter is not specified, the entire object will be downloaded</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>If-Modified-Since</td>
<td>If the object is modified after the specified time, the object will be returned; otherwise, HTTP status code 304 (Not Modified) will be returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>If-Unmodified-Since</td>
<td>If the object is not modified after the specified time, the object will be returned; otherwise, HTTP status 412 (Precondition Failed) will be returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>If-Match</td>
<td>If the ETag of the object is the same as the specified value, the object will be returned; otherwise, HTTP status code 412 (Precondition Failed) will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>If-None-Match</td>
<td>If the ETag of the object is different from the specified value, the object will be returned; otherwise, HTTP status code 304 (Not Modified) will be returned</td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

### Server-side encryption-related headers

If server-side encryption is used for the specified object and the encryption method is SSE-C, you will need to specify the headers related to server-side encryption to decrypt the object. For more information, see [Server-side Encryption Headers](#).

### Request Body

This API does not have a request body.

### Response

#### Response Headers

In addition to common response headers, this API also returns the following response headers. For more information on common response headers, see [Common Response Headers](#).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Cache directives as defined in RFC 2616, which will be returned only if it is contained in the object metadata or specified through the request parameter</td>
<td>string</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td>Filename as defined in RFC 2616, which will be returned only if it is contained in the object metadata or specified through the request parameter</td>
<td>string</td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>Encoding format as defined in RFC 2616, which will be returned only if it is contained in the object metadata or specified through the request parameter</td>
<td>string</td>
</tr>
<tr>
<td>Content-Range</td>
<td>Byte range of the return content as defined in RFC 2616, which will be returned only if the <code>Range</code> request header is specified in the request</td>
<td>string</td>
</tr>
<tr>
<td>Expires</td>
<td>Cache expiration time as defined in RFC 2616, which will be returned only if it is contained in the object metadata or specified through the request parameter</td>
<td>string</td>
</tr>
<tr>
<td>x-cos-meta-*</td>
<td>Header suffix and information of the user-defined metadata</td>
<td>string</td>
</tr>
<tr>
<td>x-cos-storage-class</td>
<td>Object storage class, such as STANDARD_IA and ARCHIVE. For enumerated values, see <a href="#">Storage Class</a>. This header will be returned only if the storage class of the object is not STANDARD</td>
<td>Enum</td>
</tr>
</tbody>
</table>

### Versioning-related headers

For versioned objects, the following response headers will be returned:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-version-id</td>
<td>Object version ID</td>
<td>string</td>
</tr>
</tbody>
</table>
Server-side encryption-related headers
If server-side encryption is used for the specified object, this API will return the server-side encryption headers. For more information, see Server-side Encryption Headers.

Response Body
The response body of this API request is the object (file) content.

Error Codes
There is no special error message for this API. For all error messages, see Error Codes.

Examples

Example 1. Basic example (versioning not enabled)

Request
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jul 2019 11:33:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNom820Y1g1QDF0c3JO#####&q-sign-time=1562239980;1562247180&q-key-time=1562239980;1562247180&q-header-list=date;host&q-url-param-list=&q-signature=fa5552e4c94ab474c9b6b6bebe0fb5c15d9#####
Connection: close

Response
HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Accept-Ranges: bytes
Date: Thu, 04 Jul 2019 11:33:00 GMT
ETag: "b62e10bcab55a88240b9c436c5fde99"
Last-Modified: Thu, 04 Jul 2019 11:32:55 GMT
Server: tencent-cos
x-cos-request-id: NWQxZGUzZWNfN2RiZTBiMDlfM2EzZF8yMGYx#####

[Object Content]

Example 2. Specifying response headers through request parameters

Request
GET /exampleobject?response-content-type=application%2Foctet-stream&response-cache-control=max-age%3D86400&response-content-disposition=attachment%3B%20filename%3Dexample.jpg HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jul 2019 11:33:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNom820Y1g1QDF0c3JO#####&q-sign-time=1562239980;1562247180&q-key-time=1562239980;1562247180&q-header-list=date;host&q-url-param-list=response-cache-control;response-content-disposition;response-content-type&q-signature=a079419b6f0cd4ac1bc55bce14b544a9a3#####
Connection: close

Response
HTTP/1.1 200 OK
Content-Type: application/octet-stream
Content-Length: 13
Connection: close
Accept-Ranges: bytes
Cache-Control: max-age=86400
Content-Disposition: attachment; filename=example.jpg
Date: Thu, 04 Jul 2019 11:33:00 GMT
ETag: "b62e10bcab55a88240b9c436c5fde99"
Last-Modified: Thu, 04 Jul 2019 11:32:55 GMT
Server: tencent-cos
x-cos-request-id: NWQxZGUzZWNfNjI4NWQ2NF9lMWYyXzk1NjFj#####

©2013-2019 Tencent Cloud. All rights reserved.
Example 3. Using server-side encryption SSE-C

Request

GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jul 2019 11:33:05 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1562239985;1562247185&q-key-time=1562239985;1562247185&q-header-list=date;host&q-url-param-list=&q-signature=2f7fbbaba6f59483878ebe73ca4a829a8e****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Accept-Ranges: bytes
Date: Thu, 04 Jul 2019 11:33:06 GMT
ETag: "b62e18bca5a88240b9e4386c1d9f9"
Last-Modified: Thu, 04 Jul 2019 11:33:00 GMT
Server: tencent-cos
x-cos-request-id: NWQxZGUzZjdfZDkyNzVkNjRfZDA0Y185OGJj****
x-cos-server-side-encryption: AES256

Example 4. Using server-side encryption SSE-C

Request

GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jul 2019 11:33:11 GMT
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: NDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7fMrG8g=
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1562239991;1562247191&q-key-time=1562239991;1562247191&q-header-list=date;host;x-cos-server-side-encryption-customer-algorithm;x-cos-server-side-encryption-customer-key;x-cos-server-side-encryption-customer-key-md5&q-url-param-list=&q-signature=6e50f8adca8fbd840869a8d691abfbd2a8e****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Accept-Ranges: bytes
Date: Thu, 04 Jul 2019 11:33:11 GMT
ETag: "492b458ec33eaf0a824e7d1bd64b3b3"
Last-Modified: Thu, 04 Jul 2019 11:33:06 GMT
Server: tencent-cos
x-cos-request-id: NWQxZGUzZjdfZDkyNzVkNjRfZDA0Y185OGJj****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7fMrG8g=

Example 5. Downloading the latest version of the object (with versioning enabled)

Request

GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 04 Jul 2019 11:33:11 GMT
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key: NDEyMzQ1Njc4OUFCQ0RFRjAxMjM0NTY3ODlBQkNERUY=
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7fMrG8g=
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1562239991;1562247191&q-key-time=1562239991;1562247191&q-header-list=date;host;x-cos-server-side-encryption-customer-algorithm;x-cos-server-side-encryption-customer-key;x-cos-server-side-encryption-customer-key-md5&q-url-param-list=&q-signature=6e50f8adca8fbd840869a8d691abfbd2a8e****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Accept-Ranges: bytes
Date: Thu, 04 Jul 2019 11:33:11 GMT
ETag: "492b458ec33eaf0a824e7d1bd64b3b3"
Last-Modified: Thu, 04 Jul 2019 11:33:06 GMT
Server: tencent-cos
x-cos-request-id: NWQxZGUzZjdfZDkyNzVkNjRfZDA0Y185OGJj****
x-cos-server-side-encryption-customer-algorithm: AES256
x-cos-server-side-encryption-customer-key-MD5: U5L61r7jcwdNvT7fMrG8g=

[Object Content]
GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 05 Jul 2019 03:30:31 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1562297431;1562304631&q-key-time=1562297431;1562304631&q-header-list=host&q-url-param-list=&q-signature=Sbk2166e208606d6a5988bf6caca3df159b6****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 15
Connection: close
Accept-Ranges: bytes
Date: Fri, 05 Jul 2019 03:30:31 GMT
ETag: "60c7644eb1ae8918a7fe7e13a352712c"
Last-Modified: Fri, 05 Jul 2019 03:30:26 GMT
Server: tencent-cos
x-cos-request-id: NWQxZWM0NTdfZGEyNzVkMjYuNjQ1OTg0ODI2MjA****
x-cos-version-id: MTg0NDUxODE3NzYyODg0ODI2MjA

[Object Content]

Example 6. Downloading a specific version of the object (with versioning enabled)

Request

GET /exampleobject?versionId=MTg0NDUxODE3NzYyODg0ODI2MjA HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 05 Jul 2019 03:30:31 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1562297431;1562304631&q-key-time=1562297431;1562304631&q-header-list=host&q-url-param-list=versionid&q-signature=3ea1162e56d0b43f7398a39b99b72bb58ad****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Accept-Ranges: bytes
Date: Fri, 05 Jul 2019 03:30:31 GMT
ETag: "b62e10bcab55a88240b9d5436cfdcf9"
Last-Modified: Fri, 05 Jul 2019 03:30:21 GMT
Server: tencent-cos
x-cos-request-id: NWQxZWM0NTdfOTNjMjJhMDlfZDBjNV95ZTMz****
x-cos-version-id: MTg0NDUxODE3NzYyODg0ODI2MjA

[Object Content]

Example 7. Downloading partial content by specifying the Range request header

Request

GET /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 09 Aug 2019 04:02:17 GMT
Range: bytes=2-4
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0****&q-sign-time=1565323337;1565308537&q-key-time=1565323337;1565308537&q-header-list=host&range&q-url-param-list=&q-signature=8bc84427bbb1b5cfba8c456c9b3c85d8e3****
Connection: close

Response

HTTP/1.1 206 Partial Content
Content-Type: image/jpeg
Content-Length: 3
HEAD Object

Last updated : 2019-12-16 17:56:24

Description

This API is used to determine whether the specified object exists and whether you have permission to access it. You can use this API to get the metadata of the object if you can access it. To make this request, you need to have the permission to read the target object or the target object allows Public Read.

Versioning

If versioning is enabled, the HEAD operation will return the latest version of the metadata. To get a previous version of the metadata, please use the versionId request parameter.

Request

Sample Request

HEAD /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for details).

Request Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>versionId</td>
<td>Specifies the version ID of the object to be queried</td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Headers

In addition to common request headers, this API also supports the following request headers. For more information on common request headers, see Common Request Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>If-Modified-Since</td>
<td>If the object is modified after the specified time, HTTP status code 200 (OK) will be returned; otherwise, HTTP status code 304 (Not Modified) will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>If-Unmodified-Since</td>
<td>If the object is not modified after the specified time, HTTP status code 200 (OK) will be returned; otherwise, HTTP status code 412 (Precondition Failed) will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>If-Match</td>
<td>If the ETag of the object is the same as the specified value, HTTP status code 200 (OK) will be returned; otherwise, HTTP status code 412 (Precondition Failed) will be returned</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>If-None-Match</td>
<td>If the ETag of the object is different from the specified value, HTTP status code 200 (OK) will be returned; otherwise, HTTP status code 304 (Not Modified) will be returned</td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

Server-side encryption-related headers

If server-side encryption is used for the specified object and the encryption method is SSE-C, you will need to specify the headers related to server-side encryption to decrypt the object. For more information, see Server-side Encryption Headers.

Request Body

This API does not have a request body.
Response

Response Headers

In addition to common response headers, this API returns the following response headers. For more information, see Common Response Headers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Cache directives as defined in RFC 2616, which will be returned only if it is contained in the object metadata</td>
<td>string</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td>Filename as defined in RFC 2616, which will be returned only if it is contained in the object metadata</td>
<td>string</td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>Encoding format as defined in RFC 2616, which will be returned only if it is contained in the object metadata</td>
<td>string</td>
</tr>
<tr>
<td>Expires</td>
<td>Cache expiration time as defined in RFC 2616, which will be returned only if it is contained in the object metadata</td>
<td>string</td>
</tr>
<tr>
<td>x-cos-meta-*</td>
<td>Header suffix and information of the user-defined metadata</td>
<td>string</td>
</tr>
<tr>
<td>x-cos-storage-class</td>
<td>Object storage class, such as STANDARD_IA and ARCHIVE. For enumerated values, see Storage Class. This header will be returned only if the storage class of the object is not STANDARD</td>
<td>Enum</td>
</tr>
</tbody>
</table>

Versioning-related Headers

For versioned objects, the following response headers will be returned:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-version-id</td>
<td>Object version ID</td>
<td>string</td>
</tr>
</tbody>
</table>

Server-side encryption-related headers

If server-side encryption is used for the specified object, this API will return the server-side encryption headers. For more information, see Server-side Encryption Headers.

Response Body

The response body of this API is empty.

Error Codes

There is no special error message for this API. For all error messages, see Error Codes.

Example

Request

HEAD /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 09 Aug 2019 10:21:01 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKID8A0fBVtYFrNm02oY1g1JQQF0c3J0*****&q-sign-time=1565346061;1565353261&q-key-time=1565346061;1565353261&q-header-list=date;host&q-url-param-list=&q-signature=82f401cf54cd6ad8331d1c8b8c827bf8f2f9*****
Connection: close

Response

HTTP/1.1 200 OK
Content-Type: image/jpeg
Content-Length: 13
Connection: close
Date: Fri, 09 Aug 2019 10:21:01 GMT
ETag: "b62e10bcab55a88240b9c436cffdcf9"
Last-Modified: Fri, 09 Aug 2019 10:20:56 GMT
DELETE Object

Last updated: 2019-12-16 17:49:23

<table>
<thead>
<tr>
<th>Server: tencent-cos</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-request-id: NMD0ZQS5GfNwRjMDJhMl5l0ThjMl5xNzE2****</td>
</tr>
</tbody>
</table>
This API is used to delete an object from a COS bucket. To make this request, you need to have the permission to write to the bucket.

**Notes**
- If an object you want to delete in a `DELETE Object` request does not exist, the request will still be successful and `204 No Content` will be returned.
- To use the `DELETE Object` API, you need to have Write access to the object.

**Request**

**Sample Request**

```
DELETE /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: length
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

**Request Headers**

**Common Headers**
The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

**Special Headers**
This request does not use any special request header.

**Request Body**
The request body of this request is empty.

**Response**

**Response Headers**

**Common Response Headers**
This response uses common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**
This request does not use any special response header.

**Response Body**
The response body of this request is empty.

**Error Analysis**
The following describes some frequent special errors that may occur when you make this request:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchBucket</td>
<td>404 Not Found</td>
<td>The bucket does not exist</td>
</tr>
</tbody>
</table>

For more COS error codes or a complete list of errors, see Error Codes.

**Example**

**Request**
DELETE /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 23 Oct 2016 21:32:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDMtTC8jfM50wL9CawaAI0b2ThTSUjyGF0&q-sign-time=1484213409;325571894098&q-key-time=1484213409;325571894098&q-header-list=host&q-url-param-list=&q-signature=1c24fe260ffe79b8603f932c2e016a6c6bb8af44a

Response

HTTP /1.1 204 No Content
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Wed, 23 Oct 2016 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRjYTRYnRjMzNzFhOF06MmN3Yg==
DELETE Multiple Objects

**Description**

This API is used to delete multiple objects from a specified bucket in a single request. You can delete up to 1,000 objects in a single request. For the response, there are two modes for you to choose from: Verbose and Quiet. Verbose mode returns the information on the deletion of each object, whereas Quiet mode only returns the information on objects for which errors are reported.

This request must carry `Content-MD5` for an integrity check on the request body.

**Notes**

1. You can delete up to 1,000 objects in a single DELETE Multiple Object request.
2. This API supports response in two modes: Verbose (default) and Quiet. Verbose mode returns the information on the deletion of each key, whereas Quiet mode only returns the information on the keys which the request fails to delete.
3. This request must carry a `Content-MD5` header for the integrity check on the request body.
4. If a key you want to delete in a request does not exist, the request can still be successful.

**Request**

Syntax example:

```
POST /?delete HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: length
Content-Type: application/xml
Content-MD5: MD5
Authorization: Auth String

<Delete>
    <Quiet></Quiet>
    <Object>
        <Key></Key>
    </Object>
    <Object>
        <Key></Key>
    </Object>
    ...
</Delete>
```

Authorization: Auth String (see Request Signature for details).

**Request Line**

```
POST /?delete HTTP/1.1
```

This API allows POST requests.

**Request Headers**

**Common Headers**

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

**Special Headers**
Required headers

The implementation of this operation uses the following required request headers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Length</td>
<td>HTTP request length in bytes as defined in RFC 2616</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Content-MD5</td>
<td>Base64-encoded 128-bit MD5 checksum as defined in RFC 1864. This header is used to verify whether the file content has changed</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Body

The node content of this request body is:

```xml
<Delete>
  <Quiet></Quiet>
  <Object>
    <Key></Key>
  </Object>
  <Object>
    <Key></Key>
  </Object>
  ...
</Delete>
```

The content is described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE</td>
<td>None</td>
<td>The result return mode and the objects to be deleted</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Quiet</td>
<td>DELETE</td>
<td>A boolean value which determines whether to use the Quiet mode. If its value is True, the Quiet mode will be used; if it is False, Verbose mode will be used. Default value: False</td>
<td>Boolean</td>
<td>No</td>
</tr>
<tr>
<td>Object</td>
<td>DELETE</td>
<td>Information on the objects to be deleted</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Key</td>
<td>DELETE.Object</td>
<td>Filenames of the objects to be deleted</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Response

Response Headers

Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers
This request does not use any special response header.

**Response Body**

This response body returns **application/xml** data. The following contains all the node data:

```xml
<DeleteResult>
  <Deleted>
    <Key></Key>
  </Deleted>
  <Error>
    <Key></Key>
    <Code></Code>
    <Message></Message>
  </Error>
</DeleteResult>
```

The content is described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteResult</td>
<td>None</td>
<td>The result return mode and target objects of this deletion</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node **DeleteResult** :**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleted</td>
<td>DeleteResult</td>
<td>Information on objects that have been successfully deleted in this operation</td>
<td>Boolean</td>
</tr>
<tr>
<td>Error</td>
<td>DeleteResult</td>
<td>Information on objects that have not been deleted in this operation</td>
<td>Container</td>
</tr>
</tbody>
</table>

**Content of the Container node **Deleted** :**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>DeleteResult.Deleted</td>
<td>Object name</td>
<td>String</td>
</tr>
</tbody>
</table>

**Content of the Container node **Error** :**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>DeleteResult.Error</td>
<td>Names of the objects that have not been deleted</td>
<td>String</td>
</tr>
<tr>
<td>Code</td>
<td>DeleteResult.Error</td>
<td>Error code of the deletion failure</td>
<td>String</td>
</tr>
<tr>
<td>Message</td>
<td>DeleteResult.Error</td>
<td>Error message for the deletion failure</td>
<td>String</td>
</tr>
</tbody>
</table>

**Error Analysis**

The following describes some frequent special errors that may occur when you make this request:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidRequest</td>
<td>400 Bad Request</td>
<td>The required Content-MD5 field is not carried, and Missing required header for this request: Content-MD5 will be returned</td>
</tr>
<tr>
<td>Error Code</td>
<td>HTTP Status Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>400 Bad Request</td>
<td>If the number of keys in a request exceeds 1,000, a MalformedXML error will be returned along with the error message. Delete key size is greater than 1000.</td>
</tr>
<tr>
<td>InvalidDigest</td>
<td>400 Bad Request</td>
<td>The Content-MD5 carried does not match the request body calculated by the server.</td>
</tr>
</tbody>
</table>

For more COS error codes or a complete list of errors, see Error Codes.

### Examples

#### Request

```
POST /?delete HTTP/1.1
Host: lelu06-1252400000.cos.ap-guangzhou.myqcloud.com
Date: Wed, 23 Oct 2016 21:32:00 GMT
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15Isski8QK72ba60Wmpc8oV1o9Smu098&q-sign-time=1480932292;1981012292&q-key-time=1480932292;1981012292&q-url-param-list=delete&q-header-list=host&q-signature=c54f22fd92232a7672ba5996a5a8a7352d2ef
Content-MD5: yoLiNjQuvB7lu8cEmPafrQ==
Content-Length: 125

<Delete>
  <Quiet>true</Quiet>
  <Object>
    <Key>aa</Key>
  </Object>
  <Object>
    <Key>aaa</Key>
  </Object>
</Delete>
```

#### Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 17
Connection: keep-alive
Date: Tue, 22 Aug 2017 12:00:48 GMT
Server: tencent-cos
x-cos-request-id: NTk5YzFjZjBjZWFHsZ011G9fWJmZV91ZGMSZ0==

<DeleteResult/>
```

```
POST /?delete HTTP/1.1
Host: lelu06-1252400000.cos.ap-guangzhou.myqcloud.com
Date: Tue, 22 Aug 2017 12:16:35 GMT
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: */*
User-Agent: python-requests/2.12.4
Authorization: q-sign-algorithm=sha1&q-ak=AKID15Isski8QK72ba60Wmpc8oV1o9Smu098&q-sign-time=1480932292;1981012292&q-key-time=1480932292;1981012292&q-url-param-list=delete&q-header-list=host&q-signature=c54f22fd92232a7672ba5996a5a8a7352d2ef
Content-MD5: V0XuU8V7aqMYeWyD3BC2nQ==
Content-Length: 126
```
<Delete>
  <Quiet>false</Quiet>
  <Object>
    <Key>aa</Key>
  </Object>
  <Object>
    <Key>aaa</Key>
  </Object>
</Delete>

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 111
Connection: keep-alive
Date: Tue, 22 Aug 2017 12:16:35 GMT
Server: tencent-cos
x-cos-request-id: NTk5YzIwYTNfMzFhYzM1M25fMmNhNzZfZWNhNjQ=

<DeleteResult>
  <Deleted>
    <Key>aa</Key>
  </Deleted>
  <Deleted>
    <Key>aaa</Key>
  </Deleted>
</DeleteResult>
OPTIONS Object

Description

This API is used to implement a pre-request for cross-origin object access configuration. Before making a real cross-origin resource sharing (CORS) request, you can make an OPTIONS request carrying the specific source origin, HTTP method, and header information to COS for it to determine whether a real request can be made. If there is no CORS configuration, 403 Forbidden will be returned. You can enable CORS for a bucket using the PUT Bucket cors API.

Request

Sample Request

```plaintext
OPTIONS /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Origin: Origin
Access-Control-Request-Method: HTTPMethod
Access-Control-Request-Headers: RequestHeader
Authorization: Auth String
```

Request Headers

**Common Headers**

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

**Special Headers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>string</td>
<td>Source origin of the simulated cross-origin access</td>
<td>Yes</td>
</tr>
<tr>
<td>Access-Control-Request-Method</td>
<td>string</td>
<td>HTTP request method of the simulated cross-origin access</td>
<td>Yes</td>
</tr>
<tr>
<td>Access-Control-Request-Headers</td>
<td>string</td>
<td>Request headers of the simulated cross-origin access</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Body

The request body of this request is empty.

Response

Response Headers

**Common Response Headers**

This response contains common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**
The special response headers used by this operation include:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access-Control-Allow-Origin</td>
<td>string</td>
<td>Source origin of the simulated cross-origin access. This header will not be returned if the origin is not allowed</td>
</tr>
<tr>
<td>Access-Control-Allow-Methods</td>
<td>string</td>
<td>HTTP request method of the simulated cross-origin access. This header will not be returned if the request method is not allowed</td>
</tr>
<tr>
<td>Access-Control-Allow-Headers</td>
<td>string</td>
<td>Request headers of the simulated cross-origin access. If any request header is not allowed, it will not be returned by this response header</td>
</tr>
<tr>
<td>Access-Control-Expose-Headers</td>
<td>string</td>
<td>HTTP request method of the simulated cross-origin access. This header will not be returned if the request method is not allowed</td>
</tr>
<tr>
<td>Access-Control-Max-Age</td>
<td>string</td>
<td>Sets the validity period of the result of the OPTIONS request</td>
</tr>
</tbody>
</table>

**Response Body**

This response body is empty.

**Example**

**Request**

```plaintext
OPTIONS /exampleobject HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Thu, 12 Jan 2017 17:25:53 GMT
Origin: http://www.qq.com
Access-Control-Request-Method: PUT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDDNEycgLPI2axw9xa2Hx87WZ39QOCn&q-sign-time=1487070734;3246654734&q-key-time=1487070734;32559966734&q-header-list=host&q-url-param-list=&q-signature=2ac3ada19910f44668ae9df72a0ec1003f34324b
```

**Response**

```plaintext
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 16087
Connection: keep-alive
x-cos-request-id: NTg3NzRiZGRfYmRjMzVfM2Y2OF81N2YzNA==
Date: Thu, 12 Jan 2017 17:26:53 GMT
ETag: "9a4802d5c99dafe1c04da0e7e166bfV"
Access-Control-Allow-Origin: http://www.qq.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-cos-request-id
Server: tencent-cos
```
POST Object restore

Last updated: 2019-12-16 17:52:25

Description

This API is used to restore an object archived by COS. The restored readable object is temporary, and you can make configuration to keep it readable and set the time when you want it to be deleted. You can use the Days parameter to specify the expiration time of the temporary object. If the object expires and you have not initiated any operation to copy the object or extend its validity period before the expiration, the temporary object will be automatically deleted. A temporary object is only a copy of the source archived object and the source object will exist throughout the period.

Request

Sample Request

POST /<ObjectKey>?restore HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The implementation of this operation requires the following request body.

<RestoreRequest>
  <Days>2</Days>
  <CASJobParameters>
    <Tier>Bulk</Tier>
  </CASJobParameters>
</RestoreRequest>

The detailed data are described as follows:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestoreRequest</td>
<td>None</td>
<td>Container used for data restoration</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Days</td>
<td>None</td>
<td>Sets expiration time of the temporary copy</td>
<td>integer</td>
<td>Yes</td>
</tr>
<tr>
<td>CASJobParameters</td>
<td>None</td>
<td>Container of the parameters of CAS jobs</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>
When restoring data, you can specify `Tier` as one of the three modes supported by CAS: `Standard`, standard mode where a restoration job can be completed in 3-5 hours; `Expedited`, expedited mode where a restoration job can be completed in 15 minutes; and `Bulk`, batch mode where a restoration job can be completed in 5-12 hours.

<table>
<thead>
<tr>
<th>Tier</th>
<th>None</th>
</tr>
</thead>
</table>

**Response**

**Response Headers**

**Common Response Headers**
This response contains common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**
This response does not use any special response header.

**Response Body**
This response body is empty.

**Error Codes**
The following error messages may be returned for this request operation. For common error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Restoration succeeded</td>
<td>202 Accepted</td>
</tr>
<tr>
<td>RestoreAlreadyInProgress</td>
<td>Object being restored</td>
<td>409 Conflict</td>
</tr>
</tbody>
</table>

**Example**

**Request**

```
POST /exampleobject?restore HTTP/1.1
Accept: */*
Authorization: q-sign-algorithm=sha1&q-ak=AKIDZfb0A07cIgPwF9cXFrJ0B811G98J8M&q-sign-time=1497530202;1497610202&q-key-time=1497530202;1497610202&q-header-list=&q-url-param-list=&q-signature=28e9a4986df1fbed8255e9719b580557ae0u057
Host: examplebucket-1250000000.cos.ap-guangzhou.myqcloud.com
Content-Length: 105
Content-Type: application/x-www-form-urlencoded
```
<RestoreRequest>
  <Days>2</Days>
  <CASJobParameters>
    <Tier>Bulk</Tier>
  </CASJobParameters>
</RestoreRequest>

Response

HTTP/1.1 202 Accepted
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Thu, 15 Jun 2017 12:37:29 GMT
Server: tencent-cos
x-cos-request-id: NTk0MjdmODlfMjQ4OGY3XzYzYzhfMjc=
Overview

This API (COS Select) is used to extract the content from the specified object (in CSV or JSON format) using Structured Query Language (SQL) statements. During the extraction process, you need to specify the content delimiter and use an appropriate SQL function. COS Select will return the matching extraction result, and you can specify the save format of the result.

For more information on COS Select, see COS Select Overview. For more information on the SQL expressions of COS Select, see SELECT Command in the Developer Guide.

Permission Restrictions

To use COS Select, you must have the permission to `cos:GetObject`.

- If you are using a root account, you have the permission by default.
- If you are using a sub-account, contact your root account to get the permission to this operation. For more information on permission settings, see Granting a Sub-account Access to COS.

Object Data Format

COS Select supports extracting object data in the following formats:

- CSV: An object is stored in CSV format with its data records separated with a specific delimiter.
- JSON: An object is stored in JSON format, which can be either a JSON file or a JSON list.

- CSV and JSON objects need to be encoded in UTF-8.
- COS Select supports extracting CSV and JSON objects compressed by gzip or bzip2.
- COS Select supports extracting CSV and JSON objects encrypted with SSE-COS.

Request

Sample Request

```
POST /<ObjectKey>?select&select-type=2 HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: date
Authorization: Auth String

Request body
```

- Authorization: Auth String (see Request Signature for details).
- The request parameters `select` and `select-type=2` are required, where the former represents the initiation of a select request, and the latter represents the version information of the API.

Request Header

This API uses only a common request header. For more information on common request headers, see Common Request Headers.

Request Body

The following request shows how to initiate a COS Select request to extract all the content of a CSV object and save the result as a CSV object.

```
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>SELECT * from COSObject</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <SelectObjectId>object-identifier</SelectObjectId>
    <SelectObjectContent>object-content</SelectObjectContent>
  </InputSerialization>
</SelectRequest>
```
The following request shows how to initiate a COS Select request to extract all the content of a JSON object and save the result as a JSON object.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
  <Expression>Select * from COSObject</Expression>
  <ExpressionType>SQL</ExpressionType>
  <InputSerialization>
    <CompressionType>GZIP</CompressionType>
    <JSON>
      <Type>DOCUMENT</Type>
    </JSON>
  </InputSerialization>
  <OutputSerialization>
    <JSON>
      <RecordDelimiter>
      </RecordDelimiter>
    </JSON>
  </OutputSerialization>
  <RequestProgress>
    <Enabled>FALSE</Enabled>
  </RequestProgress>
</SelectRequest>
```

The `InputSerialization` element describes the format of the object to be extracted. It is a required parameter and can be specified in CSV or JSON format.

The `OutputSerialization` element describes the format in which the extraction result is saved. This parameter can be specified in CSV or JSON format.

The format of the object to be extracted can be different from that in which the extraction result is saved, so you can extract an object in JSON format and save the extraction result in CSV format, and vice versa.

The following table shows the composition of the elements in a request body:

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>InputSerialization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OutputSerialization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Expression</td>
<td>SelectRequest</td>
<td>An SQL expression that represents the extraction operation which you need to initiate, such as <code>SELECT s._1 FROM COSObject s</code>. This expression extracts the first column of content from a CSV object. For more information on SQL expressions, see <code>SELECT Command</code>.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>ExpressionType</td>
<td>SelectRequest</td>
<td>Expression type, which is an extension. Currently, only SQL expressions and parameters are supported.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>InputSerialization</td>
<td>SelectRequest</td>
<td>Describes the format of the object to be extracted.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>OutputSerialization</td>
<td>SelectRequest</td>
<td>Describes the output format of the extraction result.</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>RequestProgress</td>
<td>SelectRequest</td>
<td>Whether to return the query progress information (<code>QueryProgress</code>). If this parameter is selected, COS Select will periodically return the query progress information.</td>
<td>Container</td>
<td>No</td>
</tr>
</tbody>
</table>

**InputSerialization container element**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| CompressionType | InputSerialization | Describes the compression format of the object to be extracted:  
- If the object has not been compressed, this parameter is NONE  
- If the object has been compressed, COS Select currently supports gzip and bzip2 formats. Value range: NONE, GZIP, and BZIP2. Default value: NONE. | String   | No       |
| CSV/JSON        | InputSerialization | Describes the required file parameters under the corresponding object format. For example, for CSV format, the delimiter needs to be specified.                                                           | Container| Yes CSV or JSON       |

**CSV container element (InputSerialization sub-element)**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>CSV</td>
<td>A character that separates the data records in a CSV object into different rows, which is \n by default. You can specify any octal character such as a comma, semicolon, and tab. This parameter supports up to 2 bytes, i.e., you can enter a delimiter in the format of  \r\n . Default value: \n .</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>FieldDelimiter</td>
<td>CSV</td>
<td>Specifies the character separating rows in a CSV object, which is , by default. You can specify any octal character. This parameter supports up to 1 byte. Default value: , .</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>CSV</td>
<td>If there is a string in the CSV object to be extracted that contains delimiters, you can use QuoteCharacter to escape it so as to prevent it from being cut into several parts. For example, if there is a string &quot;a, b&quot; in the CSV object, double quotation mark &quot; can prevent this string from being separated into two characters ( a and b ). Default value: &quot;.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>QuoteEscapeCharacter</td>
<td>CSV</td>
<td>If the string to be extracted contains &quot;&quot;, then you need to use &quot; &quot; to escape it so as to ensure that the string can be escaped normally. For example, your string &quot;&quot;&quot;a, b&quot;&quot; will be parsed as &quot; a , b &quot; . Default value: &quot;.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>AllowQuotedRecordDelimiter</td>
<td>CSV</td>
<td>Specifies whether the object to be extracted contains any character that is the same as the delimiter and needs to be escaped using &quot;&quot;. When this parameter is set to TRUE, COS Select will escape the character during extraction, which will result in a decrease in the extraction performance. When it is set to FALSE, no escaping will be performed. Default value: FALSE.</td>
<td>Boolean</td>
<td>No</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>FileHeaderInfo</td>
<td>CSV</td>
<td>Whether the object to be extracted contains a column header. Value range: NONE, USE, IGNORE. NONE indicates that the object contains no column headers, USE indicates that the object contains a column header and you can use the head for extraction (such as SELECT &quot;name&quot; FROM COSObject), and IGNORE indicates that the object contains a column header and you do not intend to use the header for extraction (but you can still use the column index for extraction, such as SELECT s._1 FROM COSObject s).</td>
<td>Enum</td>
<td>No</td>
</tr>
<tr>
<td>Comments</td>
<td>CSV</td>
<td>Specifies a record as a comment line. This character is added to the first character of the record. If a record is specified as a comment, COS Select will not perform any analysis on it. Default value: #.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**JSON container element (InputSerialization sub-element)**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| Type      | JSON        | JSON file type:  
- DOCUMENT indicates that the JSON file contains only an independent JSON object which can be cut into multiple rows  
- LINES indicates that each row in the JSON object contains an independent JSON object  
Value range: DOCUMENT, LINES | Enum | Yes |

OutputSerialization container element

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV /JSON</td>
<td>OutputSerialization</td>
<td>Specifies the output format of the extraction result. Value range: CSV, JSON</td>
<td>Container</td>
<td>Yes, which must be CSV or JSON</td>
</tr>
</tbody>
</table>

CSV container element (OutputSerialization sub-element)

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuoteFields</td>
<td>CSV</td>
<td>Specifies whether the output result needs to be escaped with &quot; if it is a file. Value range: ALWAYS, ASNEEDED. ALWAYS indicates applying &quot; to all the output extracted files, and ASNEEDED indicates using it only when needed. Default value: is ASNEEDED.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>RecordDelimiter</td>
<td>CSV</td>
<td>A character that separates the data records in the output result into different rows, which is \n by default. You can specify any octal character such as a comma, semicolon, and tab. This parameter supports up to 2 bytes, i.e., you can enter a delimiter in the format of \r\n. Default value: \n.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>FieldDelimiter</td>
<td>CSV</td>
<td>A character that separates each row in the output result into different columns, which is , by default. You can specify any octal character. This parameter supports up to 1 byte. Default value: , .</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>QuoteCharacter</td>
<td>CSV</td>
<td>If there is a string in the output result that contains delimiters, you can use QuoteCharacter to escape it so as to ensure that the string will not be cut in subsequent analysis. For example, if there is a string a, b in the output result, double quotation mark &quot; can prevent this string from being separated into two characters ( a and b ), and COS Select will convert it to &quot;a, b&quot; to be written to the file. Default value: &quot; .</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>
## Cloud Object Storage

### QuoteEscapeCharacter

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuoteEscapeCharacter</td>
<td>CSV</td>
<td>If the string to be output contains &quot;&quot;, then you need to use &quot;&quot; to escape it so as to ensure that the string can be escaped normally. For example, your string &quot;a, b&quot; will be converted to &quot;&quot;a, b&quot;&quot; when written to the file. Default value: &quot;&quot;.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

### JSON container element (OutputSerialization sub-element)

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>RecordDelimiter</td>
<td>JSON</td>
<td>A character that separates the data records in the output result into different rows, which is \n by default. You can specify any octal character such as a comma, semicolon, and tab. This parameter supports up to 2 bytes, i.e., you can enter a delimiter in the format of \r\n. Default value: \n.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

### RequestProgress container element

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>RequestProgress</td>
<td>Specifies whether COS Select should periodically return the query progress information. Default value: FALSE</td>
<td>Boolean</td>
<td>No</td>
</tr>
</tbody>
</table>

### Response

A successful extraction operation will return a **200 OK** status code.

**Response Header**

This API only returns a common response header. For more information, see [Common Response Headers](#).

**Response Body**
As the size of the response body is unpredictable, COS presents the response body in serialized form, i.e., dividing the response body into multiple parts to be returned. The following shows an overview of the returned response body:

- Message 1
- Message 2
- Message 3
- .......
- Message n

### Pre-response (Prelude) and Response Result (Data)

COS cuts a extraction result into multiple parts, each of which is a message. Each message consists the pre-response (prelude) and response result (data).

- The prelude consists of two parts:
  - Total length of the message.
  - Total length of all headers.
- The data consists of two parts:
  - Header.
  - Payload.

Both the prelude and data end with a 4-byte CRC code encoded in Big Endian. COS Select uses CRC32 to calculate the CRC code. For more information on CRC32, see the [RFC documentation](https://tools.ietf.org/html/rfc8484). In addition to data, COS Select additionally spends a total of 16 bytes in transferring the prelude and code.

The integer values in all the messages are transferred in network byte order (i.e., encoded in Big Endian).

The figure below shows what a message and a header consist of. One message may contain multiple headers.

As shown above, each message consists of a prelude, a prelude CRC code (composed of two pieces of information that record the number of bytes), header(s), a payload, and a message CRC code. As can be seen from the above figure, the length of the entire response body is calculated as follows:

\[
\text{Total length of a response body} = \text{Length of the prelude} + \text{length of the prelude CRC code} + \text{length of the payload} + \text{length of the header(s)} + \text{length of the message CRC code}
\]

As the total length of the prelude, the prelude CRC code and the message CRC code is always 16 bytes, the total length of the response body can also be quickly calculated as follows:

\[
\text{Total length of a response body} = \text{Length of the payload} + \text{length of the header(s)} + 16
\]

The following describes the components of the response body in detail:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| prelude   | Records the total length of each message and the total length of all headers separately. Each record is of 4 bytes, and the total length is 8 bytes:  
  1. total byte-length: the total length of the message, which is encoded in Big Endian and of 4 bytes in total with the capacity of the record itself included.  
  2. headers byte-length: the total length of all headers, which is encoded in Big Endian and of 4 bytes in total with the capacity occupied by the record excluded. |
| prelude CRC | A prelude CRC is encoded in Big Endian and contains a total of 4 bytes. It helps the program quickly determine whether the prelude information is correct so as to reduce blocking during buffering. |
| header    | Metadata of the extraction result recorded by the message, such as data type and body format. The length of this part in bytes varies by data type. A header is stored as a key-value (KV) pair and encoded in UTF-8. The metadata recorded in a header can be displayed in any order, but each metadata entry is recorded only once. Depending on the data type, the following headers may appear in the result returned by COS Select:  
  1. MessageType Header: This header represents the response type, where the key is "message-type" and the value can be "error" or "event". "error" indicates that this record is an error message, and "event" indicates that this record is a specific event.  
  2. EventType Header: This header records the event type, where the key is "event-type" and the value can be "Records", "Cont", "Progress", "Stats" or "End". "Records" indicates that the event is the returned extraction record, "Cont" the TCP connection hold, "Progress" the periodically returned extraction result, "Stats" the statistics of the query, and "End" the end of query.  
  3. ErrorCode Header: This header records the error code, where the key is "error-code" and the value can be an error code listed in Special Error Codes.  
  4. ErrorMessage Header: This header records the error message, where the key is "error-message" and the value can be an error message returned by the server, which can be used to locate the error. |
| Payload   | Records the extraction result or official information related to the request. |
| Message CRC | CRC code encoded in Big Endian containing a total of 4 bytes. |

A message may record multiple headers, and each header consists of the following parts:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Name Byte-Length</td>
<td>Records the length of the header name in bytes</td>
</tr>
</tbody>
</table>
| Header Name             | Header type. Value range: "message-type", "event-type", "error-code", "error-message"  
  - "message-type" indicates that the header records the response type  
  - "event-type" indicates that the header records the event type  
  - "error-code" indicates that the header records the error code  
  - "error-message" indicates that the header records the error message. |
| Header Value Type       | Type of the header value, which is always 7 for COS Select, indicating that the type is String |
| Value String Byte-Length| Length of the header value in bytes, which is always 2 bytes               |
| Header Value String     | Body of the header, i.e., the metadata of the payload, where the length of the header value in bytes depends on the response type |

COS Select supports the following response types:

<table>
<thead>
<tr>
<th>Response Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records message</td>
<td>An extraction record message, which can contain a single record, a partial record, or multiple records, depending on the number of extraction results. A response body may contain multiple records messages</td>
</tr>
<tr>
<td>Continuation message</td>
<td>A connection continuation message that is sent by COS Select periodically to maintain the TCP connection and appears randomly in the response body. It is recommended to make your client able to automatically identify this type of messages and filter them out so as to avoid smudging the extraction results</td>
</tr>
<tr>
<td>Progress message</td>
<td>A progress message returned by COS Select periodically to indicate the current query progress</td>
</tr>
<tr>
<td>Stats message</td>
<td>A statistics message about the query returned by COS Select after the query ends</td>
</tr>
<tr>
<td>End message</td>
<td>An end message indicating that the query has ended and there is no subsequent response data. The query can be considered to have ended only when a message of this type is received</td>
</tr>
<tr>
<td>Response Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RequestLevelError message</td>
<td>An error message which will be returned by COS Select when an error occurs during the query process and contains the cause of the error. If COS Select returns this message, it will not return an end message</td>
</tr>
</tbody>
</table>

These response types are as detailed below.

**Records Message**

- **Header format**
  A records message contains three types of headers: `:message-type`, `:event-type`, and `:content-type`, as shown below:

- **Body format**
  The body of a records message may contain a single record, a partial record, or multiple records, depending on the number of extraction results.

**Continuation Message**

- **Header format**
  A continuation message contains two types of headers: `:message-type` and `:event-type`, as shown below:
Body format
A continuation message contains no body content.

Progress Message
- Header format
A progress message contains three types of headers: ":message-type", ":event-type", and ":content-type", as shown below:
Body format
The body of a progress message is XML text which contains the current query progress, mainly including:
- BytesScanned: If the file is compressed, this value represents the size of the file in bytes before it is decompressed; otherwise, this value represents the size of the file in bytes.
- BytesProcessed: If the file is compressed, this value represents the size of the file in bytes after it is decompressed; otherwise, this value represents the size of the file in bytes.
- BytesReturned: Size of the extraction result currently returned by COS Select in bytes.

Below is a sample:
```xml
<?xml version="1.0" encoding="UTF-8"?>
<Progress>
  <BytesScanned>512</BytesScanned>
  <BytesProcessed>1024</BytesProcessed>
  <BytesReturned>1024</BytesReturned>
</Progress>
```

Stats Message
Header format
A stats message contains three types of headers: "message-type", "event-type", and "content-type", as shown below:
### Body format

The body of a stats message is XML text which contains the statistics of the current query, mainly including:

- **BytesScanned**: If the file is compressed, this value represents the size of the file in bytes before it is decompressed; otherwise, this value represents the size of the file in bytes.
- **BytesProcessed**: If the file is compressed, this value represents the size of the file in bytes after it is decompressed; otherwise, this value represents the size of the file in bytes.
- **BytesReturned**: Size of the extraction result returned by COS Select in the query in bytes.

Below is a sample:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Stats>
  <BytesScanned>512</BytesScanned>
  <BytesProcessed>1024</BytesProcessed>
  <BytesReturned>1024</BytesReturned>
</Stats>
```

### End Message

- **Header format**

  An end message contains two types of headers: `:message-type` and `:event-type`, as shown below:

### Request Level Error Message

- **Header format**

  A request level error message contains three types of headers: `:error-code`, `:error-message`, and `:message-type`, as shown below:
For more information on the error code in a request level error message, see Special Error Codes.

- Body format
  A request level error message contains no body content.

### Special Error Codes

For common error messages for this request, see Error Codes. Special error codes are as shown below:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Message</th>
<th>Meaning</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidXML</td>
<td>The XML is invalid</td>
<td>The XML format is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingRequiredParameter</td>
<td>The SelectRequest entity is missing a required parameter</td>
<td>A required parameter is missing in the extraction request.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingExpectedExpression</td>
<td>The SQL expression is missing</td>
<td>An SQL expression is missing.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingInputSerialization</td>
<td>The input serialization is missing</td>
<td>No data serialization format of the input CSV object is specified.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Message</td>
<td>Meaning</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>InvalidCompressionFormat</td>
<td>The file is not in a supported compression format. Only GZIP and BZIP2 are supported.</td>
<td>The file is in an invalid compression format. Only gzip and bzip2 are supported.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingInputFormat</td>
<td>The input format is missing.</td>
<td>The input format is missing.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidFileHeaderInfo</td>
<td>The input FileHeaderInfo is invalid. Only NONE, USE, and IGNORE are supported</td>
<td>The input file header information is invalid. Only NONE, USE, and IGNORE are supported.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The input RecordDelimiter of CSV is invalid</td>
<td>The record delimiter of the CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The input FieldDelimiter of CSV is invalid</td>
<td>The field delimiter of the CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The input QuoteCharacter of CSV is invalid</td>
<td>The quote character of the CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The input AllowQuoteRecordDelimiter of CSV is invalid. Only TRUE and FALSE are supported</td>
<td>The configuration for enabling the escape character in the input CSV file is invalid. Only TRUE and FALSE are supported.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidJsonType</td>
<td>The JsonType is invalid. Only DOCUMENT and LINES are supported</td>
<td>The JSON type is invalid. Only DOCUMENT and LINES are supported.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingOutputSerialization</td>
<td>The output serialization is missing.</td>
<td>No data serialization format of the output CSV object is specified.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>MissingOutputFormat</td>
<td>The output format is missing.</td>
<td>The output format is missing.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidQuoteFields</td>
<td>The QuoteFields is invalid. Only ALWAYS and ASNEEDED are supported</td>
<td>The escaping rule is invalid. Only ALWAYS and ASNEEDED are supported.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The output RecordDelimiter of CSV is invalid</td>
<td>The record delimiter of the output CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The output FieldDelimiter of CSV is invalid</td>
<td>The field delimiter of the output CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The output QuoteCharacter of CSV is invalid</td>
<td>The escape character of the output CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The output QuoteEscapeCharacter of CSV is invalid</td>
<td>The double quotation mark escape character of the output CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The output RecordDelimiter of JSON is invalid</td>
<td>The record delimiter of the output JSON file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Message</td>
<td>Meaning</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>Encountered an error parsing the SQL expression</td>
<td>An error occurred when parsing the SQL expression.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>Other expressions are not allowed in the SELECT list when '*' is used without dot notation.</td>
<td>The SELECT list does not allow '*' to be used without dot notation.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>The SQL expression contains an empty SELECT</td>
<td>The SQL expression contains an empty SELECT clause.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>GROUP is not supported in the SQL expression</td>
<td>The GROUP clause is not supported in the SQL expression.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>UNION is not supported in the SQL expression</td>
<td>The UNION clause is not supported in the SQL expression.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>FROM is missing in the SQL expression</td>
<td>The FROM clause is missing in the SQL expression.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>ORDER is not supported in the SQL expression</td>
<td>The ORDER clause is not supported in the SQL expression.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>The column index is invalid in the SQL expression</td>
<td>The column index specified in the SQL expression is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>SQLParsingError</td>
<td>The table alias is invalid in WHERE</td>
<td>The table alias in the WHERE clause is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Bzip2DecompressError</td>
<td>Encountered an error decompressing the bzip2 file</td>
<td>An error occurred when decompressing the bzip2 file.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Bzip2DecompressError</td>
<td>BZIP2 is not applicable to the queried object</td>
<td>The bzip2 format is not applicable to decompressing the object to be queried.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>GzipDecompressError</td>
<td>Encountered an error decompressing the gzip file</td>
<td>An error occurred when decompressing the gzip file.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>GzipDecompressError</td>
<td>GZIP is not applicable to the queried object</td>
<td>The gzip format is not applicable to decompressing the object to be queried.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Busy</td>
<td>The service is busy. Please retry later</td>
<td>The backend service is busy. Please retry later.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Overload</td>
<td>The service is overload. Please retry later</td>
<td>The backend service is overloaded. Please retry later.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>AmbiguousFieldName</td>
<td>Field name matches to multiple fields in the file</td>
<td>The specified header name has multiple identical values.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Message</td>
<td>Meaning</td>
<td>HTTP Status Code</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ComparisonFailed</td>
<td>Attempt to compare failed</td>
<td>The attempt to compare failed. Please retry.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>CastFailed</td>
<td>Attempt to convert from one data type to another using CAST failed in the SQL expression.</td>
<td>The attempt to convert from one data type to another using the CAST function failed in the SQL expression.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>OverMaxRecordSize</td>
<td>The length of a record in the input or result is greater than maxCharsPerRecord of 1 MB</td>
<td>The size of a record in the input or output exceeds the limit of 1 MB.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>LastRecordParseFail</td>
<td>Please check the last record in the input</td>
<td>Please check the last record in the input file.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>CSVParsingError</td>
<td>Encountered an error parsing the CSV file</td>
<td>An error occurred when parsing the CSV file.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>JSONParsingError</td>
<td>Encountered an error parsing the JSON file</td>
<td>An error occurred when parsing the JSON file.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>ErrorWritingRow</td>
<td>Encountered an error parsing the SELECT result. Please try again</td>
<td>Unable to format your query result. Please check the file and retry.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidRequestParameter</td>
<td>The input Comment of CSV is invalid</td>
<td>The comment of the CSV file is invalid.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>InvalidTextEncoding</td>
<td>UTF-8 encoding is required. Please check the file and try again.</td>
<td>The file and the result can only be encoded in UTF-8.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>NoSuchKey</td>
<td>The specified key does not exist</td>
<td>The specified object key does not exist.</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>AccessDenied</td>
<td>Access Denied</td>
<td>Access is denied due to incorrect signature or permission.</td>
<td>403 Forbidden</td>
</tr>
<tr>
<td>MethodNotAllowed</td>
<td>The specified method is not allowed against this resource</td>
<td>The current resource does not support the HTTP method.</td>
<td>405 Method Not Allowed</td>
</tr>
<tr>
<td>InternalError</td>
<td>We encountered an internal error. Please try again</td>
<td>The server encountered an internal error.</td>
<td>500 Internal Server</td>
</tr>
</tbody>
</table>

**Samples**

**Sample 1. Extracting Content from an Object in CSV Format**

The following sample shows the process of calling this API to extract all the content from a CSV object and outputting the extraction result in CSV format. The object to be extracted is named `exampleobject.csv` and stored in the bucket `examplebucket-1250000000` in Beijing (ap-beijing).

```xml
POST /exampleobject.csv?select&select-type=2 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 12 Jan 2019 11:49:52 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
<SelectRequest>
```
If you need to execute different extraction commands, you can modify the SQL command in the **Expression** element. For more information on commands, see [SELECT Command](#). Some common extraction scenarios are described below.

- Suppose that you use a column index to filter the content of an object. You can use `s.n` to filter out the data in the `n` column, with the minimum value of `n` being 1. The following command will filter out the records in column 3 with a value greater than 100 from the object and return columns 1 and 2 of those records:

  ```sql
  SELECT s._1, s._2 FROM COSObject s WHERE s._3 > 100
  ```

- If your CSV object has a column header and you want to filter the content of the object using the name of the header (by setting `FileHeaderInfo` to `Use`), you can use `s.name` for indexing. The following command will filter out records with a header named `Id` or `FirstName`:

  ```sql
  SELECT s.Id, s.FirstName FROM COSObject s
  ```

- You can also specify a function in the SQL expression. The following command will count the number of records in the first column that are smaller than 1:

  ```sql
  SELECT count(*) FROM COSObject s WHERE s._1 < 1
  ```

The following is a sample response:

```
HTTP/1.1 200 OK
x-cos-id-2: cos_id_demo
x-cos-request-id: cos_request_id_demo
Date: Tue, 12 Jan 2019 11:50:29 GMT

A series of messages
```

### Sample 2. Extracting Content from an Object in JSON Format

The following sample shows the process of calling this API to extract all the content from a JSON object and outputting the extraction result in CSV format. The object to be extracted is named `exampleobject.json` and stored in the bucket `examplebucket-1250000000` in Beijing (ap-beijing).

```
POST /exampleobject.json?select&select-type=2 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 12 Jan 2019 11:52:29 GMT
Authorization: authorization string
Content-Length: content length

<?xml version="1.0" encoding="UTF-8"?>
```
Similarly, you can also perform different extraction commands on JSON objects by modifying the SQL command in the `Expression` element. For more information on commands, see SELECT Command. Some common extraction scenarios are described below.

- You can extract the corresponding data using the JSON attribute name. The following command will filter out records whose `city` value is Seattle from the object and return the `country` and `city` information of those records:

  ```
  SELECT s.country, s.city from COSObject s where s.city = 'Seattle'
  ```

- You can also specify a function in the SQL expression. The following command will count the total number of records in the JSON object:

  ```
  SELECT count(*) FROM COSObject s
  ```

**Notes**

Unlike the GET Object API, SELECT Object Content does not support the following features:

- Returning a part of an object: You cannot use parameters such as `Range` to specify to return a part of an object.
- Manipulating archived objects (in ARCHIVE storage class). COS Select cannot directly manipulate archived objects. To do so, you need to retrieve the data first.
Access Control List (acl)
PUT Object acl
Last updated: 2019-12-16 17:58:14

Description
This API is used to make ACL configuration for a specified object in a specified bucket. You can pass in the ACL configuration through the `x-cos-acl`, `x-cos-grant-read`, and `x-cos-grant-full-control` headers or through the request body in XML format.

Request

Sample Request

```
PUT /<ObjectKey>?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers
The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-acl</td>
<td>Defines the ACL attribute of the object. Valid values: <code>private</code>, <code>public-read</code>, and <code>default</code>. Default value: <code>default</code> (i.e., inheriting the bucket's permission). Note: currently, there can be up to 1,000 entries in one ACL. If you do not need access control for the object, set <code>default</code> for this parameter or simply leave it blank, and the object will inherit the permissions of the bucket</td>
<td>string</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-read</td>
<td>Grants the grantee Read access. Format: <code>x-cos-grant-read: id=&quot;[OwnerUin]&quot;</code></td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-grant-full-control</td>
<td>Grants the grantee full permission. Format: <code>x-cos-grant-full-control: id=&quot;[OwnerUin]&quot;</code></td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Request Body
This response body returns application/xml data. The following contains all the node data:

```
<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
    <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
      <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <Permission>READ</Permission>
      </Grantee>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```
Detailed data are shown below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlPolicy</td>
<td>None</td>
<td>Container storing the result of GET Object acl</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node **AccessControlPolicy**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>AccessControlPolicy</td>
<td>Information on the object owner</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>AccessControlList</td>
<td>AccessControlPolicy</td>
<td>Information on the grantee and permissions</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node **Owner**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>AccessControlPolicy.Owner</td>
<td>Object owner ID. Format: qcs:cam:uin/&lt;OwnerUin&gt;:uin/&lt;SubUin&gt; For root accounts, &lt;OwnerUin&gt; and &lt;SubUin&gt; have the same value</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>DisplayName</td>
<td>AccessControlPolicy.Owner</td>
<td>Object owner name</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node **AccessControlList**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>AccessControlPolicy.AccessControlList</td>
<td>Permissions on a single object. One AccessControlList can have 100 Grant entries</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node **Grant**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Name (Keyword)</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Grantee</td>
<td>AccessControlPolicy.AccessControlList.Grant</td>
<td>Describes the information on the grantee. The type can be RootAccount or Subaccount. - If the type is RootAccount, the ID specifies a root account. - If the type is Subaccount, the ID specifies a sub-account</td>
<td>Container</td>
<td>Yes</td>
</tr>
<tr>
<td>Permission</td>
<td>AccessControlPolicy.AccessControlList.Grant</td>
<td>Specifies the permission granted to the grantee. Enumerated values: READ, FULL_CONTROL</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node **Grantee**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Specifies all users</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>ID</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>User ID in the format of qcs:cam:uin/&lt;OwnerUin&gt;:uin/&lt;SubUin&gt;; For root accounts, &lt;OwnerUin&gt; and &lt;SubUin&gt; have the same value</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>DisplayName</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Username</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Response**

**Response Headers**

**Common Response Headers**

This response uses common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**

This request operation does not use any special response header.

**Response Body**

The response body of this request is empty.

**Error Codes**

This response may contain the following error codes. For common error messages, see Error Codes.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>HTTP Status Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SignatureDoesNotMatch</td>
<td>This error code will be returned if the provided signature does not conform to the rule</td>
<td>403 Forbidden</td>
</tr>
</tbody>
</table>
NoSuchBucket  If the bucket to which you want to add the rule does not exist, this error code will be returned  404 Not Found

MalformedXML  Invalid XML format. Please check against the RESTful API documentation  400 Bad Request

InvalidRequest  Invalid request. If the error message shows "header acl and body acl conflict", it means that you cannot set the ACL parameters in the header and the body at the same time  400 Bad Request

Example

Request

PUT /exampleobject?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 25 Feb 2017 04:10:22 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA10b2ThTSUjGF0&sign-time=1484724784;32557628784&key-time=1484724784;32557628784&q-header-list=host&q-url-param-list=acl&q-signature=785d9075b8154119e6a075713c1b9e56ff0bddd4
Content-Length: 229
Content-Type: application/x-www-form-urlencoded

<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
    <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
        <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Fri, 25 Feb 2017 04:10:22 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjFjMmJfOWIxZjRlXzZmNDhfMjIw

©2013-2019 Tencent Cloud. All rights reserved.  Page 208 of 259
GET Object acl

Description

This API is used to get the access permissions on a specified object in a specified bucket. Only the bucket owner has the permission to perform this operation.

Version

By default, the GET operation returns the current version of the object. If you want to get a different version, use the versionId subresource.

Request

Sample Request

GET /<ObjectKey>?acl HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

Required headers

The implementation of this request operation uses the following required request headers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>Signature string</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This response does not use any special response header.

Response Body

This response body returns application/xml data. The following contains all the node data:

```xml
<AccessControlPolicy>
  <Owner>
    <LastUpdated>2019-12-16 17:58:48</LastUpdated>
  </Owner>
</AccessControlPolicy>
```
Detailed data are shown below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlPolicy</td>
<td>None</td>
<td>Container storing the result of GET Object acl</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node **AccessControlPolicy**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>AccessControlPolicy</td>
<td>Information on the object owner</td>
<td>Container</td>
</tr>
<tr>
<td>AccessControlList</td>
<td>AccessControlPolicy</td>
<td>Information on the grantee and permissions</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node **Owner**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>AccessControlPolicy.Owner</td>
<td>Object owner ID. Format: qcs::cam::uin/&lt;OwnerUin&gt;&lt;uin&gt;/&lt;SubUin&gt;. For root accounts, &lt;OwnerUin&gt; and &lt;SubUin&gt; have the same value</td>
<td>String</td>
</tr>
<tr>
<td>DisplayName</td>
<td>AccessControlPolicy.Owner</td>
<td>Object owner name</td>
<td>String</td>
</tr>
</tbody>
</table>

Content of the Container node **AccessControlList**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant</td>
<td>AccessControlPolicy.AccessControlList</td>
<td>Permissions on a single object. One AccessControlList can have 100 Grant entries</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node **Grant**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
</table>
### Node Name (Keyword) | Parent Node | Description | Type
--- | --- | --- | ---
Grantee | AccessControlPolicy.AccessControlList.Grant | Describes the information on the grantee. The type can be RootAccount or Subaccount. If the type is RootAccount, the ID specifies a root account; if the type is Subaccount, the ID specifies a sub-account | Container
Permission | AccessControlPolicy.AccessControlList.Grant | Specifies the permission granted to the grantee. Enumerated values: READ, FULL_CONTROL | String

#### Content of the Container node  
**Grantee**:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Specifies all users</td>
<td>String</td>
</tr>
<tr>
<td>ID</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>User ID in the format of qcs::cam::uin/&lt;OwnerUin&gt;:uin/&lt;SubUin&gt;. For root accounts, &lt;OwnerUin&gt; and &lt;SubUin&gt; have the same value</td>
<td>String</td>
</tr>
<tr>
<td>DisplayName</td>
<td>AccessControlPolicy.AccessControlList.Grant.Grantee</td>
<td>Username</td>
<td>String</td>
</tr>
</tbody>
</table>

### Example

#### Request

GET /exampleobject?acl HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB&q-sign-time=1484213027;32557109027&q-key-time=1484213027;32557109027&q-header-list=host&q-url-param-list=acl&q-signature=dcc1eb2022b79cb2a7b8b1062d3a48e12b4065

#### Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 266
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg3NzRiMjVfYmRjMzVfMTViMl82ZGZmNw==

```xml
<AccessControlPolicy>
  <Owner>
    <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
    <DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://cam.qcloud.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CanonicalUser">
        <ID>qcs::cam::uin/100000000001:uin/100000000001</ID>
      </Grantee>
      <!-- Permission details -->
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```
<DisplayName>qcs::cam::uin/100000000001:uin/100000000001</DisplayName>
</Grantee>
<Permission>FULL_CONTROL</Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>

Multipart Upload
Initiate Multipart Upload
Description

An Initiate Multipart Upload API request initiates a multipart upload. After the request is fulfilled, the system will return to UploadId for the subsequent Upload Part requests.

Request

Request Example

```
POST /<ObjectKey>?uploads HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

We implement this request with a common request header. For more information about common request headers, see Common Request Headers.

Special Request Headers

Recommended Headers

We implement this request with the following recommended request header information:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control</td>
<td>Cache policy defined in RFC 2616, which will be stored as the object's metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Content-Disposition</td>
<td>File name defined in RFC 2616, which will be stored as the object's metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Content-Encoding</td>
<td>Encoding format defined in RFC 2616, which will be stored as the object's metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Content-Type</td>
<td>Content type (MIME) defined in RFC 2616, which will be stored as the object's metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Expires</td>
<td>File date and time defined in RFC 2616, which will be stored as the object's metadata</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-meta-*</td>
<td>This includes the suffix and information of the user-defined header, which will be returned as the object metadata of up to 2 KB. <strong>Note:</strong> User-defined header information contains underscores, but user-defined header suffixes does not</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-storage-class</td>
<td>This sets the storage class of the object. Enumerated values: STANDARD, STANDARD_IA, and ARCHIVE. Default value: STANDARD</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Permission-related Headers

For more information about request ACL, see ACL Overview.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
</table>

Last updated : 2019-08-05 12:29:09
Cloud Object Storage

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-acl</td>
<td>It defines the ACL attribute of the object. Value range: private, public-</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>read, and default; default value: default (i.e., inheriting the bucket's</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>permissions);</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Currently, there can be up to 1,000 entries in one ACL. If you do</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>not need access control for the object, use &quot;default&quot; for this parameter or</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>simply leave it blank, so that the object will inherit the permissions of the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>bucket</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x-cos-grant-read</td>
<td>This grants the READ permission in the format of x-cos-grant-read: id=&quot;</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>[OwnerUin]&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x-cos-grant-full-control</td>
<td>This grants the grantee all permissions in the format of x-cos-grant-full-</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>control: id=&quot;[OwnerUin]&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Server-side encryption-related headers**

This request operation specifies the data encryption policy during COS data upload. COS will automatically encrypt the data written to the IDC and automatically decrypt it when you retrieve it. Currently, AES-256 encryption with the COS master key is supported. If you want to enable server-side encryption for your data, upload the following header:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-</td>
<td>This specifies how the server-side encryption is enabled for the object.</td>
<td>String</td>
<td>Yes if encryption is needed</td>
</tr>
<tr>
<td>encryption</td>
<td>For encryption with the COS master key, enter AES256</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Request Body**

The request body of this request is null.

**Response**

**Response Headers**

**Common Response Headers**

This response uses a common response header. For more information about the common response header, see **Common Response Headers**.

**Special Response Headers**

**Server-side encryption-related response**

If it is specified to use server-side encryption when the file is uploaded, the response header will contain the following information:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption</td>
<td>If the object is stored with COS-managed server-side encryption, the response</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td>will contain the values of this header and the encryption algorithm used (AES256)</td>
<td></td>
</tr>
</tbody>
</table>

**Response Body**

The return of this response body is **application/xml** data. Below is an example containing all the node data:

```xml
<InitiateMultipartUploadResult>
  <Bucket>examplebucket-1250000000</Bucket>
  <Key>exampleobject</Key>
  <UploadId>1484727270323ddb949d526c629235314a9ead00f0b5d993a3d76d46de6a9cceb9d3b88e</UploadId>
</InitiateMultipartUploadResult>
```

**Detailed data is as shown below:**

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitiateMultipartUploadResult</td>
<td>None</td>
<td>This describes all the returned information</td>
<td>Container</td>
</tr>
</tbody>
</table>
Content of the Container node InitiateMultipartUploadResult:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>InitiateMultipartUploadResult</td>
<td>Destination bucket for the multipart upload, which is formed by connecting the user-defined string and system-generated APPID with a dash, such as examplebucket-1250000000</td>
<td>Container</td>
</tr>
<tr>
<td>Key</td>
<td>InitiateMultipartUploadResult</td>
<td>Object name</td>
<td>Container</td>
</tr>
<tr>
<td>Uploadid</td>
<td>InitiateMultipartUploadResult</td>
<td>ID used in subsequent uploads</td>
<td>Container</td>
</tr>
</tbody>
</table>

Examples

Request

POST /exampleobject?uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDtTCBYJW50wLB9CAwA10q27his1GF0&q-sign-time=1484727259;325576232598&q-key-time=1484727259;325576232598&q-header-list=host&q-url-param-list=uploads&q-signature=b5f46c47579eae0e74be7778f8b133c01be2b045

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 230
Connection: keep-alive
Date: Fri, 10 Mar 2016 09:45:46 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjIzZTZfOWIxZjRlXzZmMzhfMWRj

<InitiateMultipartUploadResult>
  <Bucket>examplebucket-1250000000</Bucket>
  <Key>exampleobject</Key>
  <UploadId>1484727270323ddb949d528c629235314a9ead80f8ba5d993a3d7f6b46e6a9cceb9633b08e</UploadId>
<InitiateMultipartUploadResult>
Upload Part

Last updated: 2020-02-28 18:21:52

Description

API Upload Part requests the implementation to upload objects to COS in the way of part. A maximum of 10000 part is supported, each part is 1MB-5GB in size, and the last part can be smaller than 1MB.

Detail analysis

1. Multipart upload first needs to initialize it and use the Initiate Multipart Upload API to implement it. After initialization, you will get a uploadId that uniquely identifies this upload.
2. When you request a Upload Part, you need to carry the numbers of partNumber and uploadId, partNumber as blocks, and you can upload them out of order.
3. When the incoming uploadId and partNumber are the same, the later incoming block will overwrite the previously passed block. A 404 error, NoSuchUpload, is returned when uploadId does not exist.

Request

Request Sample

```
PUT /<ObjectKey>partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: Size
Authorization: Auth String

[Object]
```

For more information on Authorization: Auth String, please see Request Signature Documentation.

Request header

Public head

The implementation of this request operation uses a common request header. For more information on common request headers, see Common Request Headers Document.

Non-public head

Required header

The request operation requires the request header to use a required header, as shown below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Length</td>
<td>HTTP request content length defined in RFC 2616 (bytes)</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Recommended head

This request operation recommends that the request header be used as the recommended header. The details are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expect</td>
<td>HTTP request content length defined in RFC 2616 (bytes)</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Content-MD5</td>
<td>The Base64-encoded MD5 hash value of the request body content defined in RFC 1864, which is used for integrity checking to verify whether the request body has changed during transmission.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

Request parameter

The details are as follows:
<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartNumber</td>
<td>Identify the number of multipart upload this time. PartNumber should be</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>greater than or equal to 1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UploadId</td>
<td>When the ID, that identifies multipart upload uses the Initiate Multipart</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Upload interface to initialize the multipart upload, it will get a uploadId,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The ID not only uniquely identifies this part data, but also identifies the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>relative location of the part data in the whole file.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Request body**

The request body of the request is the data content of the part.

**Response**

**Response header**

**Public response header**

The response includes Public response header. For more information on Public response header, please see Common Response Header Document.

**Unique response header**

The response will probably return the following response header information:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-cos-server-side-encryption</td>
<td>If server encryption is specified when uploading, the response header will</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>return the response header. Enumerated value: AES256</td>
<td></td>
</tr>
<tr>
<td>X-cos-storage-class</td>
<td>Returns the storage class information of the object. COS returns this</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>response header for all objects except Standard storage class. Enumerated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>values: STANDARD_IA and ARCHIVE</td>
<td></td>
</tr>
</tbody>
</table>

**Response body**

The response body for the request is empty.

**Error Codes**

There is no special error message for this request operation. For common error messages, see Error Codes Document.

**Actual case**

**Request**

```none
PUT /exampleobject?partNumber=1&uploadId=1484727278323dd0949d528c029235314a9ead80fba5d993a3d76b460e6a9e95c6403e HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj****&q-sign-time=1484727403;32557623403&q-key-time=1484727403;32557623403&q-header-list=host&q-url-param-list=partNumber;uploadId&q-signature=bfc54518ca8fc31b3ea287f1e2a8dd8c8e8e8****
Content-Length: 10485760

[Object]
```

**Response**

```none
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Wed, 18 Jan 2017 16:17:03 GMT
Etag: "e1e5b4965bc7d30880ed6d226f78a5390f1c09fc"
Server: tencent-cos
x-cos-request-id: NTg3ZjI0NzlfOWIxZjRlXzZmMGYfMWYy
```
Upload Part - Copy

Last updated : 2019-12-16 18:11:51

Description

This API is used to copy the parts of an object from the source path to the destination path. You can use `x-cos-copy-source` to specify the source object and use `x-cos-copy-source-range` to specify the byte range (allowed part size: 5 MB - 5 GB).

- If the destination object and the source object are in different regions, and the part size of the destination object will exceed 5 GB, you will need to use multipart upload or multipart copy API to copy the object.
- To upload an object in parts, you must first initialize the multipart upload. A unique descriptor (upload ID) will be returned in the response of multipart upload initialization, which needs to be carried in the multipart upload request.

Version

If versioning is enabled for the bucket, `x-cos-copy-source` identifies the current version of the object to be copied. If the current version is a delete marker and no version is specified for `x-cos-copy-source`, COS will consider the object as a deleted object and return a 404 error. If you specify a `versionId` in `x-cos-copy-source` and the `versionId` is a delete marker, COS will return an HTTP 400 error as it is not allowed to use a delete marker as the version of `x-cos-copy-source`.

Request

Sample Request

```plaintext
PUT /examplebucket?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: <Bucketname-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
x-cos-copy-source: <Bucketname>-<APPID>.cos.<Region>.myqcloud.com/filepath
x-cos-copy-source-range: bytes=first-last
x-cos-copy-source-if-match: etag
x-cos-copy-source-if-none-match: etag
x-cos-copy-source-if-unmodified-since: time_stamp
x-cos-copy-source-if-modified-since: time_stamp
```

Authorization: Auth String (see Request Signature for more information).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

Required Headers

The implementation of this operation uses the following required request headers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-copy-source</td>
<td>URL path to the source object. A historical version can be specified using the versionId subresource</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Recommended Headers

The implementation of this operation uses the following recommended request headers:
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-copy-source-range</td>
<td>Byte range of the source object. The range value must be in the format of</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>bytes=first-last, where both first and last are offsets starting from 0. For example, bytes=0-9 means that you want to copy the first 10 bytes of data of the source object. If this parameter is not specified, the entire object will be copied</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x-cos-copy-source-If-Modified-Since</td>
<td>If the object is modified after the specified time, the operation will be performed; otherwise, 412 will be returned. This header can be used together with x-cos-copy-source-If-None-Match. If it is used together with other conditions, a conflict will be returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-copy-source-If-Unmodified-Since</td>
<td>If the object is not modified after the specified time, the operation will be performed; otherwise, 412 will be returned. This header can be used together with x-cos-copy-source-If-None-Match. If it is used together with other conditions, a conflict will be returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-copy-source-If-Match</td>
<td>If the Etag of the object is the same as the specified one, the operation will be performed; otherwise, 412 will be returned. This header can be used together with x-cos-copy-source-If-Unmodified-Since. If it is used together with other conditions, a conflict will be returned</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-copy-source-If-None-Match</td>
<td>If the Etag of the object is different from the specified one, the operation will be performed; otherwise, 412 will be returned. This header can be used together with x-cos-copy-source-If-Unmodified-Since. If it is used together with other conditions, a conflict will be returned</td>
<td>string</td>
<td>No</td>
</tr>
</tbody>
</table>

**Request Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>partNumber</td>
<td>Part number</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>uploadId</td>
<td>To upload a file in parts, you must first initialize the multipart upload. A unique descriptor (upload ID) will be returned in the response of multipart upload initialization, which needs to be carried in the multipart upload request</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Request Body**

The request body of this request is empty.

**Response**

**Response Headers**

**Common Response Headers**

This response contains common response headers. For more information on common response headers, see Common Response Headers.

**Special Response Headers**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-copy-source-version-id</td>
<td>Version of the copied source object if versioning is enabled for the source bucket</td>
<td>String</td>
</tr>
<tr>
<td>x-cos-server-side-encryption</td>
<td>If the object is stored with COS-managed server-side encryption, the response will contain this header and the value of the encryption algorithm used (AES256)</td>
<td>String</td>
</tr>
</tbody>
</table>

**Response Body**

This response body returns application/xml data. The following contains all the node data:
<?xml version="1.0" encoding="UTF-8" ?>
<CopyPartResult>
<ETag>"ba82b57cfdfda8bd17ad4e5879ebb4f6"</ETag>
<LastModified>2017-09-04T04:45:45</LastModified>
</CopyPartResult>

The data are described in details below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyPartResult</td>
<td>Returns the result of the copy</td>
<td>String</td>
</tr>
<tr>
<td>ETag</td>
<td>Returns the MD5 checksum of the object. The value of ETag can be used to check whether the object content has changed</td>
<td>String</td>
</tr>
<tr>
<td>LastModified</td>
<td>Returns the last modified time of the object in GMT time</td>
<td>String</td>
</tr>
</tbody>
</table>

**Example**

**Request**

PUT /exampleobject?partNumber=1&uploadId=150576248ca8373f8a5cd52cb129f4bcf86a1dce833df34f4f5bccc466c99a42cd1ffa21f9 HTTP/1.1
User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.13.1.0 zlib/1.2.3 libidn/1.18 libssh2/1.2.2
Accept: */*
x-cos-copy-source:examplebucket-1250000000.cos.ap-shanghai.myqcloud.com/exampleobject1
x-cos-copy-source-range: bytes=10-100
Host: examplebucket-1250000000.cos.ap-shanghai.myqcloud.com
Authorization:q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3Mq0Cn&q-sign-time=1507530223;1508530223&q-key-time=1507530223;1508530223&q-header-list=&q-url-param-list=&q-signature=d02640c0821c49293e5c289fa07290e6b2f95cb3

**Response**

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 133
Connection: keep-alive
Date: Mon, 04 Sep 2017 04:45:45 GMT
Server: tencent-cos
x-cos-request-id: NTlkYjFjYWJfMjQ4OGY3MGFfNGIzZV9k

<?xml version="1.0" encoding="UTF-8" ?>
<CopyPartResult>
<ETag>"ba82b57cfdfda8bd17ad4e5879ebb4f6"</ETag>
<LastModified>2017-09-04T04:45:45</LastModified>
</CopyPartResult>
Complete Multipart Upload

Last updated: 2019-12-16 18:01:38

Description

This API is used to complete a multipart upload. After all parts are uploaded via the Upload Part API, you need to call this API to complete the multipart upload. When using this API, you need to specify the PartNumber and ETag of each part in the request body for the part information to be verified.

As the parts need to be merged after they are uploaded, and the merge takes several minutes, COS will immediately return status code 200 when the merge starts, and periodically return spaces during the merge to keep the connection active until the merge is complete. After that, COS will return the content of the merged object in the response body.

- If any uploaded part is below 1 MB in size, 400 EntityTooSmall will be returned when this API is called.
- If the numbers of the uploaded parts are not continuous, 400 InvalidPart will be returned when this API is called.
- If the part information entries in the request body are not sorted by number in ascending order, 400 InvalidPartOrder will be returned when this API is called.
- If the uploadId does not exist, 404 NoSuchUpload will be returned when this API is called.

Note:

It is recommended to either complete or abort a multipart upload as early as possible, as the uploaded parts of an incomplete multipart upload will take up storage capacity and incur storage fees.

Version

If versioning is enabled, the GET operation will return the current version of the object. To obtain a different version, please use the versionId parameter.

Request

Syntax example:

```
POST /ObjectName?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-length: Size
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

Request Line

```
POST /ObjectName?uploadId=UploadId HTTP/1.1
```

This API allows POST requests.

Request Parameters

See the details below:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>uploadId</td>
<td>ID of this multipart upload. When the Initiate Multipart Upload API is used to initialize a multipart upload, an uploadId will be returned. The ID uniquely identifies the data of the part and its position in the entire file</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Request Headers

Common Headers
The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers
This request operation does not use any special request header.

Request Body
The node content of this request body is:

```xml
<CompleteMultipartUpload>
  <Part>
    <PartNumber><PartNumber />
    <ETag><ETag />
  </Part>
...</CompleteMultipartUpload>
```

The data are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompleteMultipartUpload</td>
<td>None</td>
<td>describes all information on this multipart upload</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node CompleteMultipartUpload:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td>CompleteMultipartUpload</td>
<td>describes information on each part in this multipart upload</td>
<td>Container</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Content of the Container node Part:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartNumber</td>
<td>CompleteMultipartUpload.Part</td>
<td>part number</td>
<td>Integer</td>
<td>Yes</td>
</tr>
<tr>
<td>ETag</td>
<td>CompleteMultipartUpload.Part</td>
<td>MD5 checksum of each part</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## Response

Response Headers

Common Response Headers
This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-server-side-encryption</td>
<td>Specifies how the server-side encryption is enabled for the object. For encryption with a COS master key, pass in AES256</td>
<td>String</td>
</tr>
<tr>
<td>x-cos-version-id</td>
<td>Version ID of the new object if versioning is enabled for the bucket</td>
<td>String</td>
</tr>
</tbody>
</table>
Response Body
This response body returns application/xml data. The following contains all the node data:

```xml
<CompleteMultipartUploadResult>
  <Location></Location>
  <Bucket></Bucket>
  <Key></Key>
  <ETag></ETag>
</CompleteMultipartUploadResult>
```

The data are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompleteMultipartUploadResult</td>
<td>None</td>
<td>Describes all the returned information</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node  CompleteMultipartUploadResult:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>CompleteMultipartUploadResult</td>
<td>Public network domain name used to access the new object</td>
<td>URL</td>
</tr>
<tr>
<td>Bucket</td>
<td>CompleteMultipartUploadResult</td>
<td>Destination bucket for the multipart upload, which is formed by connecting a user-defined string and the system-generated appid with a hyphen, such as <code>mybucket-1250000000</code></td>
<td>String</td>
</tr>
<tr>
<td>Key</td>
<td>CompleteMultipartUploadResult</td>
<td>Object name</td>
<td>String</td>
</tr>
<tr>
<td>ETag</td>
<td>CompleteMultipartUploadResult</td>
<td>The unique tag value of the merged object, which is not the MD5 checksum of the object's content and can only be used to check the object's uniqueness.</td>
<td>String</td>
</tr>
</tbody>
</table>

Example
Request

```plaintext
POST /ObjectName?uploadId=1484728886e63106c87d8d207536aa8521c89c42a436fe23bb58854a7bb5e87b7d77d4ddc48 HTTP/1.1
Host: arlenhuangtestsgnoversion-1251665777.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwL9CAwA1Qb2Th1SUjyGF0&q-sign-time=1484729794;325576257948&q-key-time=1484729794;325576257948&q-header-list=host&q-url-param-list=uploadId&q-signature=23627c8fddb3823cce425763c663f083f94828d
Content-Length: 138

<CompleteMultipartUpload><Part><PartNumber>1</PartNumber><ETag>"fc392e6589be447ff4e2d256489a9773"</ETag></Part></CompleteMultipartUpload>
```

Response

```plaintext
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 277
Connection: keep-alive
Date: Wed, 18 Jan 2017 16:17:03 GMT
Server: tencent-cos
x-cos-request-id: NFg3ZjJmYVNYyM8RkXzWBYzRFMJc1

<CompleteMultipartUploadResult>
```
Abort Multipart Upload

Last updated : 2019-12-16 18:09:10
Description

This API is used to abort a multipart upload and delete the uploaded parts. If you call this API and there is an Upload Part request that is using the multipart upload, the request will fail. If the uploadId does not exist, 404 NoSuchUpload will be returned.

It is recommended to either complete or abort a multipart upload as early as possible, as the uploaded parts of an incomplete multipart upload will take up storage capacity and incur storage fees.

Request

Sample Request

DELETE /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String

Authorization: Auth String (see Request Signature for details).

Request Parameters

See the details below:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>uploadId</td>
<td>ID of this multipart upload. When the Initiate Multipart Upload API is used to initialize a multipart upload, an uploadId will be returned. The ID uniquely identifies the data of the part and its position in the entire file</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Body

The request body of this request is empty.

Response

Response Headers

Common Response Headers

This response uses common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers

This request does not use any special response header.

Response Body

The response body of this request is empty.
Example

Request

DELETE /exampleobject?uploadId=1484727278232330b49d5528e629235314a9ead80f0ba5d93a3d76b460e6a3cceb9633b08e HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Tue, 26 Oct 2013 21:22:00 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1qThTSUjGF0&q-sign-time=1484728626;32557624626&q-key-time=1484728626;32557624626&q-header-list=host&q-url-param-list=uploadId&q-signature=2d3036b57cade4a257b48a356c922779a562b18

Response

HTTP/1.1 204 OK
Content-Type: application/xml
Content-Length: 0
Connection: keep-alive
Date: Tue, 26 Oct 2013 21:22:00 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjI5MzlfOTgxZjRlXzZhYjNfMjBh
List Multipart Uploads

Description

This API is used to query the ongoing multipart uploads. A single request operation can list up to 1,000 multipart uploads.

To make the request, you need to have the permission to read the bucket.

Request

Sample Request

```plaintext
GET /uploads HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for more information)

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Parameters

The content is described in details below:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>delimiter</td>
<td>The delimiter is a symbol. The identical paths between prefix or, if no prefix is specified, the beginning and the first delimiter are grouped and defined as a common prefix.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>encoding-type</td>
<td>Specifies the encoding type of the returned value; valid value: url</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>prefix</td>
<td>Sets that the response will only contain object keys with the specified prefix. When you make a query with prefix specified, the returned keys will still contain Prefix.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>max-uploads</td>
<td>Sets the maximum number of multipart uploads returned. Value range: [1, 1,000]. Default value: 1,000.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>key-marker</td>
<td>Used together with upload-id-marker. If upload-id-marker is not specified, only the multipart uploads whose ObjectName is lexicographically greater than key-marker will be listed; If upload-id-marker is specified, the multipart uploads whose ObjectName is lexicographically greater than the specified key-marker will be listed, and any multipart upload whose ObjectName lexicographically equals key-marker and whose UploadID is greater than upload-id-marker will also be listed</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>upload-id-marker</td>
<td>Used together with key-marker. If key-marker is not specified, upload-id-marker will be ignored; if key-marker is specified, the multipart uploads whose ObjectName is lexicographically greater than the specified key-marker will be listed, and any multipart upload whose ObjectName lexicographically equals key-marker and whose UploadID is greater than upload-id-marker will also be listed</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

**Request Body**

The request body of this request is empty.

**Response**

**Response Headers**

**Common Response Headers**

This response contains common response headers. For more information on common response headers, see [Common Response Headers](#).

**Special Response Headers**

This response does not use any special response header.

**Response Body**

This response body returns `application/xml` data. The following contains all the node data:

```
<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>
      <ID></ID>
      <DisplayName></DisplayName>
    </Initiator>
    <Owner>
      <ID></ID>
      <DisplayName></DisplayName>
    </Owner>
    <Initiated></Initiated>
  </Upload>
  <CommonPrefixes>
    <Prefix></Prefix>
  </CommonPrefixes>
</ListMultipartUploadsResult>
```

The data are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListMultipartUploadsResult</td>
<td>None</td>
<td>Information on all multipart uploads</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node `ListMultipartUploadsResult`:
<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>InitiateMultipartUploadResult</td>
<td>Destination bucket for a multipart upload, which is formed by connecting a user-defined string and the system-generated APPID with a hyphen, such as examplebucket-1250000000</td>
<td></td>
</tr>
<tr>
<td>Encoding-Type</td>
<td>ListMultipartUploadsResult</td>
<td>Specifies the encoding type of the returned value; valid value: url</td>
<td>String</td>
</tr>
<tr>
<td>KeyMarker</td>
<td>ListMultipartUploadsResult</td>
<td>The key value where the entry list starts</td>
<td>String</td>
</tr>
<tr>
<td>UploadIdMarker</td>
<td>ListMultipartUploadsResult</td>
<td>The uploadId value where the entry list starts</td>
<td>String</td>
</tr>
<tr>
<td>NextKeyMarker</td>
<td>ListMultipartUploadsResult</td>
<td>If the returned list is truncated, the NextKeyMarker returned will be the starting point of the subsequent list</td>
<td>String</td>
</tr>
<tr>
<td>NextUploadIdMarker</td>
<td>ListMultipartUploadsResult</td>
<td>If the returned list is truncated, the UploadId returned will be the starting point of the subsequent list</td>
<td>String</td>
</tr>
<tr>
<td>MaxUploads</td>
<td>ListMultipartUploadsResult</td>
<td>Sets the maximum number of multipart uploads returned. Value range: [0, 1,000]</td>
<td>String</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>ListMultipartUploadsResult</td>
<td>Whether the returned list is truncated, which is a boolean value. Valid values: TRUE, FALSE</td>
<td>Boolean</td>
</tr>
<tr>
<td>Prefix</td>
<td>ListMultipartUploadsResult</td>
<td>Sets that the response will only contain object keys with the specified Prefix. When you make a query with prefix specified, the returned keys will still contain Prefix.</td>
<td>String</td>
</tr>
<tr>
<td>Delimiter</td>
<td>ListMultipartUploadsResult</td>
<td>The delimiter is a symbol. The identical paths between prefix or, if no prefix is specified, the beginning and the first delimiter are grouped and defined as a common prefix.</td>
<td>String</td>
</tr>
<tr>
<td>Upload</td>
<td>ListMultipartUploadsResult</td>
<td>Information on each upload</td>
<td>Container</td>
</tr>
<tr>
<td>CommonPrefixes</td>
<td>ListMultipartUploadsResult</td>
<td>The identical paths between prefix and delimiter are grouped and defined as a common prefix</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node **Upload**:
### Node Name (Keyword) | Parent Node | Description | Type
---|---|---|---
Key | ListMultipartUploadsResult.Upload | Object name | String
UploadID | ListMultipartUploadsResult.Upload | ID of a multipart upload | String
StorageClass | ListMultipartUploadsResult.Upload | Indicates the storage class of the parts; enumerated values: STANDARD, STANDARD_IA, ARCHIVE | String
Initiator | ListMultipartUploadsResult.Upload | Information on the initiator of an upload | Container
Owner | ListMultipartUploadsResult.Upload | Information on the owner of the parts | Container
Initiated | ListMultipartUploadsResult.Upload | Starting time of a multipart upload | Date

#### Content of the Container node Initiator:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ListMultipartUploadsResult.Upload.Initiator</td>
<td>Unique CAM ID of the user</td>
<td>String</td>
</tr>
<tr>
<td>DisplayName</td>
<td>ListMultipartUploadsResult.Upload.Initiator</td>
<td>User ID (UIN)</td>
<td>String</td>
</tr>
</tbody>
</table>

#### Content of the Container node Owner:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ListMultipartUploadsResult.Upload.Owner</td>
<td>Unique CAM ID of the user</td>
<td>String</td>
</tr>
<tr>
<td>DisplayName</td>
<td>ListMultipartUploadsResult.Upload.Owner</td>
<td>User ID (UIN)</td>
<td>String</td>
</tr>
</tbody>
</table>

#### Content of the Container node CommonPrefixes:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>ListMultipartUploadsResult.CommonPrefixes</td>
<td>Displays a specific common prefix</td>
<td>String</td>
</tr>
</tbody>
</table>

### Error Analysis

The following describes some frequent special errors that may occur when you make this request:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>HTTP Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>400 Bad Request</td>
<td>The value of max-uploads must be an integer between 0 and 1,000; otherwise, InvalidArgument will be returned. The value of encoding-type can only be url; otherwise, InvalidArgument will be returned</td>
</tr>
</tbody>
</table>

For more COS error codes or a complete list of errors, see Error Codes.

### Example

Request:

```
GET /uploads HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2015 21:32:00 GMT
```
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTC8YjMJ5o6LB9CAwA1Q27h7SUjGFG0&q-sign-time=1484727508;32557623508&q-key-time=1484727508;32557623508&q-header-list=host&q-url-param-list=uploads&q-signature=5bd4759a738f7daa9a0550c224d86c61589c9dbbf

Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 1203
Date: Wed, 18 Jan 2015 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg3ZjI0ZGRfNDQyMDRlXzNhZmRfMjRl

<ListMultipartUploadsResult>
  <Bucket>examplebucket-1250000000</Bucket>
  <Encoding-Type/>
  <KeyMarker/>
  <UploadIdMarker/>
  <MaxUploads>1000</MaxUploads>
  <Prefix/>
  <Delimiter/>
  <IsTruncated>false</IsTruncated>
  <Upload>
    <Key>Object</Key>
    <UploadID>1484726657932bcb5b17f7a9a8cad9fc36a340ff204c7b9edf51e7dddf06d1da6220528c</UploadID>
    <Initiator>
      <ID>qcs::cam::uin/100000000001</ID>
      <DisplayName>100000000001</DisplayName>
    </Initiator>
    <Owner>
      <ID>qcs::cam::uin/100000000001</ID>
      <DisplayName>100000000001</DisplayName>
    </Owner>
    <StorageClass>Standard</StorageClass>
    <Initiated>Wed Jan 18 16:04:17 2017</Initiated>
  </Upload>
  <Upload>
    <Key>exampleobject</Key>
    <UploadID>1484727270323ddb949d528c629235314a9ead801fba5d993a3d76b460e6a9cceb9633b88e</UploadID>
    <Initiator>
      <ID>qcs::cam::uin/100000000001</ID>
      <DisplayName>100000000001</DisplayName>
    </Initiator>
    <Owner>
      <ID>qcs::cam::uin/100000000001</ID>
      <DisplayName>100000000001</DisplayName>
    </Owner>
    <StorageClass>Standard</StorageClass>
    <Initiated>Wed Jan 18 16:14:30 2017</Initiated>
  </Upload>
</ListMultipartUploadsResult>
List Parts

Last updated: 2019-12-16 18:03:37

Description

This API is used to query the uploaded parts of a specified multipart upload, i.e., listing all successfully uploaded parts of a multipart upload whose uploadId is specified.

Request

Sample Request

```
GET /<ObjectKey>?uploadId=UploadId HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Authorization: Auth String
```

Authorization: Auth String (see Request Signature for details).

Request Headers

Common Headers

The implementation of this operation uses common request headers. For more information on common request headers, see Common Request Headers.

Special Headers

This request operation does not use any special request header.

Request Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uploadId</td>
<td>String</td>
<td>Yes</td>
<td>ID of this multipart upload. When the Initiate Multipart Upload API is used to initialize a multipart upload, an uploadId will be returned. The ID uniquely identifies the data of the part and its position in the entire file</td>
</tr>
<tr>
<td>encoding-type</td>
<td>string</td>
<td>No</td>
<td>Specifies the encoding type of the returned value</td>
</tr>
<tr>
<td>max-parts</td>
<td>string</td>
<td>No</td>
<td>Maximum number of entries returned at a time. Default value: 1,000</td>
</tr>
<tr>
<td>part-number-marker</td>
<td>string</td>
<td>No</td>
<td>By default, entries are listed in UTF-8 binary order starting from marker</td>
</tr>
</tbody>
</table>

Request Body

The request body of this request is empty.

Response
Response Headers

Common Response Headers
This response contains common response headers. For more information on common response headers, see Common Response Headers.

Special Response Headers
This response does not use any special response header.

Response Body
A successful query returns application/xml data which include the information on the successfully uploaded parts.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<ListPartsResult>
  <Bucket>examplebucket-1250000000</Bucket>
  <Encoding-type/></Encoding-type>
  <Key>exampleobject</Key>
  <UploadId>1484642062bf1381e5d7b057692a1311d8d72df3a28f2633cfbe4d0a0e8bd8719f9335458b</UploadId>
  <Initiator>
    <ID>1250000000</ID>
    <Displayname>1250000000</Displayname>
  </Initiator>
  <Owner>
    <ID>qcs::cam::uin/1000000000:uin/1000000000</ID>
    <Displayname>1000000000</Displayname>
  </Owner>
  <PartNumberMarker>0</PartNumberMarker>
  <Part>
    <PartNumber>1</PartNumber>
    <LastModified>Tue Jan 17 16:43:37 2017</LastModified>
    <ETag>"a1f8e5e4d3ac6970a002a6277e191fe80a1382"</ETag>
    <Size>5242880</Size>
    <NextPartNumberMarker>1</NextPartNumberMarker>
    <StorageClass>STANDARD</StorageClass>
    <MaxParts>1</MaxParts>
    <IsTruncated>true</IsTruncated>
  </Part>
</ListPartsResult>
```

The data are described in details below:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListPartsResult</td>
<td>None</td>
<td>Stores the result of the List Parts request</td>
<td>Container</td>
</tr>
</tbody>
</table>

Content of the Container node ListPartsResult:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>ListPartsResult</td>
<td>Name of the destination bucket for the multipart upload, which is formed by connecting a user-defined string and the system-generated APPID with a hyphen, such as examplebucket-1250000000</td>
<td>string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encoding-Type</th>
<th>ListPartsResult</th>
<th>Encoding type</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>ListPartsResult</td>
<td>Object name</td>
<td>string</td>
</tr>
<tr>
<td>UploadId</td>
<td>ListPartsResult</td>
<td>ID which identifies this multipart upload</td>
<td>string</td>
</tr>
<tr>
<td>Initiator</td>
<td>ListPartsResult</td>
<td>Information on the creator of the parts</td>
<td>Container</td>
</tr>
</tbody>
</table>
### Table 1: Cloud Object Storage - Node Name (Keyword) and Description

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>ListPartsResult</td>
<td>Information on the owner of the parts</td>
<td>Container</td>
</tr>
<tr>
<td>StorageClass</td>
<td>ListPartsResult</td>
<td>Indicates the storage class of the parts; enumerated values: STANDARD, STANDARD_IA, ARCHIVE</td>
<td>string</td>
</tr>
<tr>
<td>PartNumberMarker</td>
<td>ListPartsResult</td>
<td>By default, entries are listed in UTF-8 binary order starting from marker</td>
<td>string</td>
</tr>
<tr>
<td>NextPartNumberMarker</td>
<td>ListPartsResult</td>
<td>If the returned list is truncated, the NextMarker returned will be the starting point of the subsequent list</td>
<td>string</td>
</tr>
<tr>
<td>MaxParts</td>
<td>ListPartsResult</td>
<td>Maximum number of entries returned at a time</td>
<td>string</td>
</tr>
<tr>
<td>IsTruncated</td>
<td>ListPartsResult</td>
<td>Whether the returned list is truncated, which is a boolean value. Valid values: true, false</td>
<td>boolean</td>
</tr>
<tr>
<td>Part</td>
<td>ListPartsResult</td>
<td>Metadata</td>
<td>Container</td>
</tr>
</tbody>
</table>

### Content of the Container node Initiator:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ListPartsResult.Initiator</td>
<td>Unique ID of the creator</td>
<td>string</td>
</tr>
<tr>
<td>DisplayName</td>
<td>ListPartsResult.Initiator</td>
<td>Creator's username</td>
<td>string</td>
</tr>
</tbody>
</table>

### Content of the Container node Owner:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>ListPartsResult.Owner</td>
<td>Unique ID of the owner</td>
<td>string</td>
</tr>
<tr>
<td>DisplayName</td>
<td>ListPartsResult.Owner</td>
<td>Owner's username</td>
<td>string</td>
</tr>
</tbody>
</table>

### Content of the Container node Part:

<table>
<thead>
<tr>
<th>Node Name (Keyword)</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartNumber</td>
<td>ListPartsResult.Part</td>
<td>Part number</td>
<td>string</td>
</tr>
<tr>
<td>LastModified</td>
<td>ListPartsResult.Part</td>
<td>Last modified time of a part</td>
<td>string</td>
</tr>
<tr>
<td>ETag</td>
<td>ListPartsResult.Part</td>
<td>MD5 checksum of a part</td>
<td>string</td>
</tr>
<tr>
<td>Size</td>
<td>ListPartsResult.Part</td>
<td>Part size in bytes</td>
<td>string</td>
</tr>
</tbody>
</table>

### Example

**Request**

```
GET /exampleobject?uploadId=14846428620b1f381e5d7b857692e131dd872dafa28f2633c1b5bea9e8bd87199335453b0&max-parts=1 HTTP/1.1
Host: examplebucket-1250000000.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDDNMEycglRP12xmmX9a2Hhx87w3Mrq0Cn&q-sign-time=1484643123;1484646723&q-key-time=1484643123;1484646723&q-header-list=host&q-url-param-list=max-parts;uploadid&q-signature=b8b4055724e64c9a0848198a21f025fd3f6ab3e87
```

©2013-2019 Tencent Cloud. All rights reserved.
Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 661
Connection: keep-alive
Date: Wed, 18 Jan 2017 16:17:03 GMT
x-cos-request-id: NTg3ZGRiMzhfMmM4OGY3XzdhY2NfYW==

<ListPartsResult>
  <Bucket>examplebucket-1250000000</Bucket>
  <Encoding-type/>
  <Key>exampleobject</Key>
  <UploadId>14846420620b1f381e5d7b657692a131dd8d72dafa28f2633cfbba4d0a88d9d719933545b0</UploadId>
  <Initiator>
    <ID>1250000000</ID>
    <Display Name>1250000000</Display Name>
  </Initiator>
  <Owner>
    <ID>qcs::cam::uin/10000000001:uin/10000000001</ID>
    <Display Name>10000000001</Display Name>
  </Owner>
  <PartNumberMarker>0</PartNumberMarker>
  <Part>
    <PartNumber>1</PartNumber>
    <LastModified>Tue Jan 17 16:43:37 2017</LastModified>
    <ETag>"a1f8e5e4d63ace6970a0062e627e191fe09a1362"</ETag>
    <Size>5242880</Size>
  </Part>
  <NextPartNumberMarker>1</NextPartNumberMarker>
  <StorageClass>STANDARD</StorageClass>
  <MaxParts>1</MaxParts>
  <IsTruncated>true</IsTruncated>
</ListPartsResult>
Batch Operation APIs
CreateJob

Feature Description

This API is used to create a batch operation job in a bucket. For more information, please see Batch Operation Overview.

When calling this API, please make sure that you have necessary permissions to perform operations on the objects in the bucket. The bucket owner has such permissions by default. If you do not have the permissions, please apply for them to the bucket owner first.

Request

Sample Request

```
POST /jobs HTTP/1.1
x-cos-appid: <appid>
<?xml version="1.0" encoding="UTF-8"?>
<CreateJobRequest>
...
</CreateJobRequest>
```

Request Parameters

Calling the CreateJob API requires the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-cos-appid</td>
<td>User APPID with a length of 1-64 bytes.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information, see Common Request Headers.

Request Body

You can set the configuration information for the inventory job in the request body with the XML language. Such information includes the objects to be analyzed by the inventory job, frequency and dimension of the analysis, and format and storage location of the analysis result.

```
<CreateJobRequest>
  <ClientRequestToken>string</ClientRequestToken>
  <ConfirmationRequired>boolean</ConfirmationRequired>
  <Description>string</Description>
  <Manifest>
    <Location>
      <ETag>string</ETag>
      <ObjectArn>string</ObjectArn>
      <ObjectVersionId>string</ObjectVersionId>
    </Location>
    <Spec>
      <Fields>
        <member>string</member>
      </Fields>
      <Format>string</Format>
    </Spec>
  </Manifest>
  <Operation>
    <COSPutObjectCopy>
      <AccessControlGrants>
        <COSGrant>
          <Grantee>
            <Identifier>string</Identifier>
          </Grantee>
        </COSGrant>
      </AccessControlGrants>
      <Identifier>string</Identifier>
    </COSPutObjectCopy>
  </Operation>
</CreateJobRequest>
```
The elements in the request body are described as below. For other elements, please see [CommonElements](#).

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientRequestToken</td>
<td>CreateJobRequest</td>
<td>Token unique to each request, which can prevent the frontend from initiating one batch operation job multiple times. The token can be 1–64 bytes long, and a UUID is recommended.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------</td>
<td>----------</td>
</tr>
<tr>
<td>ConfirmationRequired</td>
<td>CreateJobRequest</td>
<td>Specifies whether confirmation is required before a job is carried out. Default value: false.</td>
<td>Boolean</td>
<td>No</td>
</tr>
<tr>
<td>Description</td>
<td>CreateJobRequest</td>
<td>Job description; 0–256 bytes long. This parameter will be returned if it is configured when the job is created.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Manifest</td>
<td>CreateJobRequest</td>
<td>Inventory of the objects to be processed. You need to record the objects you want to process in the inventory.</td>
<td>Manifest Object</td>
<td>Yes</td>
</tr>
<tr>
<td>Operation</td>
<td>CreateJobRequest</td>
<td>Operation to be performed on the objects in the inventory. Currently, COS supports the PUT Object-Copy operation, which allows you to copy the existing data in a bucket.</td>
<td>Operation Object</td>
<td>Yes</td>
</tr>
<tr>
<td>Priority</td>
<td>CreateJobRequest</td>
<td>Job priority. The higher the value, the higher the priority. Value range: 0–2,147,483,647.</td>
<td>Integer</td>
<td>Yes</td>
</tr>
<tr>
<td>Report</td>
<td>CreateJobRequest</td>
<td>Job report. You can configure this parameter to output a report upon job completion for the evaluation of job execution.</td>
<td>Report Object</td>
<td>Yes</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>RoleArn</td>
<td>CreateJobRequest</td>
<td>COS resource ID used to identify the role you have created. It is required for identity verification.</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Response

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body

```xml
<CreateJobResult>
  <JobId>string</JobId>
</CreateJobResult>
```

The content is described in detail as below:

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>CreateJobResult</td>
<td>Job ID, which is automatically returned by COS after you create a job successfully. Length: 1-64 bytes.</td>
<td>String</td>
</tr>
</tbody>
</table>

Error Codes

This request operation may return the following error messages. For other errors, please see ErrorResponse.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidRequest</td>
<td>Duplicate request</td>
<td>400</td>
<td>CreateJob</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Priority must be an integer between 0 and 2,147,483,647</td>
<td>400</td>
<td>CreateJob</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>The XML Manifest field in the request body does not conform to the XML syntax</td>
<td>400</td>
<td>CreateJob</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>The XML Operation field in the request body does not conform to the XML syntax</td>
<td>400</td>
<td>CreateJob</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>The XML Report field in the request body does not conform to the XML syntax</td>
<td>400</td>
<td>CreateJob</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>The service is temporarily unavailable and unable to create a job</td>
<td>500</td>
<td>CreateJob</td>
</tr>
<tr>
<td>TooManyJobs</td>
<td>The number of jobs has reached the upper limit, making the server unavailable</td>
<td>500</td>
<td>CreateJob</td>
</tr>
</tbody>
</table>

Example

Request

```xml
POST /jobs HTTP/1.1
x-cos-appid: 1250000000
<?xml version="1.0" encoding="UTF-8"?>
<CreateJobRequest>
  <ClientRequestToken>1829b6c7-3141-42f1-9fe4-17082b841646</ClientRequestToken>
  <ConfirmationRequired>false</ConfirmationRequired>
</CreateJobRequest>
```
<Description>example job</Description>
<Manifest>
<Location>
<ETag>ec75a30f3af000e9b31d62bed75cbcad</ETag>
<ObjectArn>qcs::cos:ap-chengdu::manifest-1250000000/1250000000/source/manifest/20190715/manifest.json</ObjectArn>
</Location>
</Manifest>
<Spec>
<Format>COSInventoryReport_CSV_V1</Format>
</Spec>
<Operation>
<COSPutObjectCopy>
<MetadataDirective>Copy</MetadataDirective>
<StorageClass>STANDARD</StorageClass>
<TargetResource>qcs::cos:ap-chengdu::target-1250000000</TargetResource>
</COSPutObjectCopy>
</Operation>
<Priority>10</Priority>
<Report>
<Bucket>qcs::cos:ap-beijing::result-1250000000</Bucket>
<Enabled>true</Enabled>
<Format>Report_CSV_V1</Format>
<Prefix>example-job-result</Prefix>
<ReportScope>AllTasks</ReportScope>
</Report>
<RoleArn>qcs::cam::uin/100000000001:roleName/examplerole</RoleArn>
</CreateJobRequest>

Response
After the request above is made, COS will return the following response, indicating that the inventory job has been successfully configured.

HTTP/1.1 200
<CreateJobResult>
<JobId>65f2e4cf-83f5-42f1-9aa2-14720613da29</JobId>
</CreateJobResult>
DescribeJob

Feature Description

This API is used to get the parameters of a batch operation job and the job status. For more information, please see Batch Operation Overview.

Request

Sample Request

```
GET /jobs/<JobId> HTTP/1.1
x-cos-appid: <appid>
```

Request Parameters

Calling the DescribeJob API requires the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>Job ID.</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-appid</td>
<td>User APPID with a length of 1–64 bytes.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information, see Common Request Headers.

Request Body

This request does not have a request body.

Response

Sample Response

```
HTTP/1.1 200
<DescribeJobResult>
  ...
</DescribeJobResult>
```

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body

```
<DescribeJobResult>
  <Job>
  <ConfirmationRequired>false</ConfirmationRequired>
  <CreationTime>timestamp</CreationTime>
  <Description>string</Description>
  <FailureReasons>
    <JobFailure>
      <FailureCode>string</FailureCode>
      <FailureReason>string</FailureReason>
    </JobFailure>
  </FailureReasons>
  <JobId>string</JobId>
  <Manifest>
    <Location>
      <ETag>string</ETag>
      <ObjectArn>string</ObjectArn>
    </Location>
  </Manifest>
</Job>
```
The content is described in detail as below:

**DescribeJobResult**

This node includes the parameters and status information of the specified batch operation job.

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job</td>
<td>DescribeJobResult</td>
<td>Parameters and status information of the specified batch operation job.</td>
<td>Job Object</td>
</tr>
</tbody>
</table>

**Job**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
</table>

©2013-2019 Tencent Cloud. All rights reserved.
<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientRequestToken</td>
<td>job</td>
<td>Token unique to each request, which can prevent the frontend from initiating one batch operation job multiple times. The token can be 1–64 bytes long, and a UUID is recommended.</td>
<td>String</td>
</tr>
<tr>
<td>CreationTime</td>
<td>job</td>
<td>Job creation time.</td>
<td>Timestamp</td>
</tr>
<tr>
<td>Description</td>
<td>job</td>
<td>Job description; 1–256 bytes long. This parameter will be returned if it is configured when the job is created.</td>
<td>String</td>
</tr>
<tr>
<td>FailureReasons</td>
<td>job</td>
<td>Describes the failure reason if a job fails.</td>
<td>FailureReasons Object</td>
</tr>
<tr>
<td>JobId</td>
<td>job</td>
<td>Job ID generated after the job is successfully created; length: 1–64 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>Manifest</td>
<td>job</td>
<td>Inventory of the objects to be processed. You need to record the objects you want to process in the inventory.</td>
<td>Manifest Object</td>
</tr>
<tr>
<td>Operation</td>
<td>job</td>
<td>Operation to be performed on the objects in the inventory.</td>
<td>Operation Object</td>
</tr>
<tr>
<td>Priority</td>
<td>job</td>
<td>Job priority. The higher the value, the higher the priority. Value range: 0–2,147,483,647.</td>
<td>Integer</td>
</tr>
<tr>
<td>ProgressSummary</td>
<td>job</td>
<td>Overview of job execution, which describes how many operations were performed in the job, how many succeeded, and how many failed.</td>
<td>ProgressSummary Object</td>
</tr>
<tr>
<td>Report</td>
<td>job</td>
<td>Specifies configurations for an inventory report.</td>
<td>Report Object</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>RoleArn</td>
<td>Job</td>
<td>Identifier of the role assigned to the job; length: 1-1024 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>Status</td>
<td>Job</td>
<td>Job status. Valid values: Active, Canceled, Canceling, Complete, Completing, Failed, Failing, New, Paused, Pausing, Preparing, Ready, Suspended.</td>
<td>String</td>
</tr>
<tr>
<td>StatusUpdateReason</td>
<td>Job</td>
<td>Reason for a status update; length: 0-256 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>SuspendedCause</td>
<td>Job</td>
<td>Cause of job suspension. A job is suspended when you are creating it in the console; it will only be carried out after your confirmation. The value of this parameter can be 0-1,024 bytes long.</td>
<td>String</td>
</tr>
<tr>
<td>SuspendedDate</td>
<td>Job</td>
<td>Time when the job is suspended; the time will be recorded upon job suspension</td>
<td>Timestamp</td>
</tr>
<tr>
<td>TerminationDate</td>
<td>Job</td>
<td>Job end time.</td>
<td>Timestamp</td>
</tr>
</tbody>
</table>

**FailureReasons**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobFailure</td>
<td>FailureReason</td>
<td>Job failure code and cause.</td>
<td>JobFailure Object</td>
</tr>
</tbody>
</table>

**FailureCode**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FailureCode</td>
<td>JobFailure</td>
<td>Job failure code; length: 0-64 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>FailureReason</td>
<td>JobFailure</td>
<td>Cause of job failure; length: 0-256 bytes.</td>
<td>String</td>
</tr>
</tbody>
</table>

For other elements, see [CommonElements](#).

**Error Codes**

The following describes some frequent special errors that may occur when you make this request. For other errors, see [ErrorResponse](#).
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchJob</td>
<td>The specified job does not exist</td>
<td>404</td>
<td>DescribeJob</td>
</tr>
</tbody>
</table>

ListJobs

Last updated: 2020-01-15 17:14:02
Feature Description

This API is used to list your batch operation jobs. For more information on batch operation, please see Batch Operation Overview.

Request

Sample Request

GET /jobs?jobStatuses=<JobStatuses>&maxResults=<MaxResults>&nextToken=<NextToken> HTTP/1.1
x-cos-appid: <appid>

Request Parameters

Calling the ListJobs API requires the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>jobStatuses</td>
<td>Status of the job to be queried. If no status is specified, COS will return all jobs, including those in progress; otherwise, only jobs in the specified status will be returned. Valid values: Active, Cancelled, Cancelling, Complete, Completing, Failed, Failing, New, Paused, Pausing, Preparing, Ready, Suspended.</td>
<td>No</td>
</tr>
<tr>
<td>maxResults</td>
<td>Maximum number of jobs to be returned by COS. If this parameter is configured, the number of jobs returned each time will not exceed its value. If it is used together with the nextToken parameter, COS can return the results in multiple pages. Value range: 1–1,000. Default value: 1,000.</td>
<td>No</td>
</tr>
<tr>
<td>nextToken</td>
<td>Page break with a length of 1–64 bytes. Each ListJobs call will return the last JobId in the job list as nextToken. You can pass in the nextToken in the next ListJobs call for COS to list jobs starting from the end of the previous list. In this way, the results can be returned in multiple pages.</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-appid</td>
<td>User APPID with a length of 1–64 bytes.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information, see Common Request Headers.

Request Body

This request does not have a request body.

Response

Sample Response

HTTP/1.1 200
<ListJobsResult>
...
</ListJobsResult>

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body

<ListJobsResult>
<Jobs>
<member>
<CreationTime>timestamp</CreationTime>
<Description>string</Description>
<JobId>string</JobId>
<Operation>string</Operation>
<Priority>integer</Priority>
</member>

©2013-2019 Tencent Cloud. All rights reserved. Page 246 of 259
<ProgressSummary>
<NumberOfTasksFailed>integer</NumberOfTasksFailed>
<NumberOfTasksSucceeded>integer</NumberOfTasksSucceeded>
<TotalNumberOfTasks>integer</TotalNumberOfTasks>
</ProgressSummary>
>Status>string</Status>
<TerminationDate>timestamp</TerminationDate>
</member>
</Jobs>
</ListJobsResult>

The content is described in detail as below:

ListJobsResult
This node contains information on the batch operation jobs returned by COS.

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs</td>
<td>ListJobsResult</td>
<td>Information on the batch operation jobs returned by COS.</td>
<td>Jobs Object</td>
</tr>
<tr>
<td>NextToken</td>
<td>ListJobsResult</td>
<td>Page break with a length of 1–64 bytes. Each ListJobs call will return the last JobId in the job list as nextToken. You can pass in the nextToken in the next ListJobs call for COS to list jobs starting from the end of the previous list. In this way, the results can be returned in multiple pages.</td>
<td>String</td>
</tr>
</tbody>
</table>

Jobs

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>member</td>
<td>jobs</td>
<td>Information on one of the batch operation jobs returned by COS.</td>
<td>member Object</td>
</tr>
</tbody>
</table>

member

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationTime</td>
<td>member</td>
<td>Job creation time.</td>
<td>Timestamp</td>
</tr>
<tr>
<td>Description</td>
<td>member</td>
<td>Job description; length: 0–256 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>JobId</td>
<td>member</td>
<td>Job ID; length: 1–64 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>Operation</td>
<td>member</td>
<td>Operation performed on objects in a batch operation job, e.g. COSPutObjectCopy</td>
<td>String</td>
</tr>
<tr>
<td>Priority</td>
<td>member</td>
<td>Job priority. The higher the value, the higher the priority. Value range: 0–2,147,483,647.</td>
<td>Integer</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>ProgressSummary</td>
<td>member</td>
<td>Overview of job execution, which describes how many operations were performed in the job, how many succeeded, and how many failed.</td>
<td>ProgressSummary Object</td>
</tr>
<tr>
<td>Status</td>
<td>member</td>
<td>Job status. Valid values: Active, Cancelled, Cancelling, Complete, Completing, Failed, Failing, New, Paused, Pausing, Preparing, Ready, Suspended.</td>
<td>String</td>
</tr>
<tr>
<td>TerminationDate</td>
<td>member</td>
<td>Job end time.</td>
<td>Timestamp</td>
</tr>
</tbody>
</table>

For other elements, see [CommonElements](#).  

## Error Codes

The following describes some frequent special errors that may occur when you make this request.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>The <code>jobStatuses</code> parameter is invalid</td>
<td>400</td>
<td>ListJobs</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>The <code>maxResults</code> parameter must be an integer</td>
<td>400</td>
<td>ListJobs</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>The <code>nextToken</code> parameter is invalid</td>
<td>400</td>
<td>ListJobs</td>
</tr>
</tbody>
</table>

For other errors, see [ErrorResponse](#).
UpdateJobPriority

Feature Description

This API is used to upgrade the priority of a job. The greater the value, the higher the priority. Jobs with a higher priority will be carried out first. For more information on batch operation jobs, see Batch Operation Overview.

Request

Sample Request

POST /jobs/<JobId>/priority?priority=<Priority> HTTP/1.1
x-cos-appid: <appid>

Request Parameters

Calling the UpdateJobPriority API requires the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>ID of the batch operation job to be updated; length: 1–64 bytes.</td>
<td>Yes</td>
</tr>
<tr>
<td>priority</td>
<td>New job priority. Value range: 0-2,147,483,647.</td>
<td>Yes</td>
</tr>
<tr>
<td>x-cos-appid</td>
<td>User APPID with a length of 1–64 bytes.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information, see Common Request Headers.

Request Body

This request does not have a request body.

Response

Sample Response

HTTP/1.1 200
<UpdateJobPriorityResult>
<JobId>string</JobId>
.Priority>integer</Priority>
</UpdateJobPriorityResult>

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body

The content is described in detail as below:

UpdateJobStatusResult

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>UpdateJobPriorityResult</td>
<td>ID of the job you update; length: 1–64 bytes.</td>
<td>String</td>
</tr>
</tbody>
</table>
### Node Name

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>UpdateJobPriorityResult</td>
<td>Current job priority. Value range: 0-2,147,483,647.</td>
<td>Integer</td>
</tr>
</tbody>
</table>

## Error Codes

The following describes some frequent special errors that may occur when you make this request.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidRequest</td>
<td>No job priority is provided</td>
<td>400</td>
<td>UpdateJobPriority</td>
</tr>
<tr>
<td>NoSuchJob</td>
<td>The specified job does not exist or has already been completed</td>
<td>404</td>
<td>UpdateJobStatus, UpdateJobPriority</td>
</tr>
</tbody>
</table>

For other errors, see [ErrorResponse](#).
UpdateJobStatus

Last updated: 2020-01-15 17:17:41

Feature Description

This API is used to update the status of a job. You can use this API to start a job or cancel an ongoing job. For more information on batch operation jobs, see Batch Operation Overview.

Request

Sample Request

POST /jobs/<JobId>/status?requestedJobStatus=<RequestedJobStatus>&statusUpdateReason=<StatusUpdateReason> HTTP/1.1
x-cos-appid: <appid>

Request Parameters

Calling the UpdateJobStatus API requires the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>ID of the batch operation job to be updated.</td>
<td>Yes</td>
</tr>
<tr>
<td>requestedJobStatus</td>
<td>Your desired job status. If you change the job status to Ready, COS will think that you have confirmed the job and will execute it. If you change the job status to Cancelled, COS will cancel the job. Valid values: Ready, Cancelled.</td>
<td>Yes</td>
</tr>
<tr>
<td>statusUpdateReason</td>
<td>Reason for the status update; length: 0–256 bytes.</td>
<td>No</td>
</tr>
<tr>
<td>x-cos-appid</td>
<td>User APPID with a length of 1–64 bytes.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request Headers

This API only uses common request headers. For more information, see Common Request Headers.

Request Body

This request does not have a request body.

Response

Sample Response

HTTP/1.1 200
<UpdateJobStatusResult><JobId>string</JobId><Status>string</Status><StatusUpdateReason>string</StatusUpdateReason></UpdateJobStatusResult>

Response Headers

This API only returns common response headers. For more information, see Common Response Headers.

Response Body

The content is described in detail as below:
### UpdateJobStatusResult

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobId</td>
<td>UpdateJobStatusResult</td>
<td>ID of the job you update; length: 5–36 bytes.</td>
<td>String</td>
</tr>
<tr>
<td>Status</td>
<td>UpdateJobStatusResult</td>
<td>Current job status. Valid values: Active, Cancelled, Cancelling, Complete, Completing, Failed, Failing, New, Paused, Pausing, Preparing, Ready, Suspended.</td>
<td>String</td>
</tr>
<tr>
<td>StatusUpdateReason</td>
<td>UpdateJobStatusResult</td>
<td>Reason for the status update; length: 0–256 bytes.</td>
<td>String</td>
</tr>
</tbody>
</table>

### Error Codes

The following describes some frequent special errors that may occur when you make this request.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>The value of the <code>requestedJobStatus</code> parameter must be Cancelled or Ready</td>
<td>400</td>
<td>UpdateJobStatus</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>The specified job has already been completed</td>
<td>400</td>
<td>UpdateJobStatus</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Error with the job status change</td>
<td>400</td>
<td>UpdateJobStatus</td>
</tr>
<tr>
<td>NoSuchJob</td>
<td>The specified job does not exist or has already been completed</td>
<td>404</td>
<td>UpdateJobStatus, UpdateJobPriority</td>
</tr>
</tbody>
</table>

For other errors, see ErrorResponse.
This document describes the common elements used in the batch operation feature.

### Manifest

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Manifest</td>
<td>Location of the object inventory.</td>
<td>Location Object</td>
<td>Yes</td>
</tr>
<tr>
<td>Spec</td>
<td>Manifest</td>
<td>Format of the object inventory. If the inventory is a CSV file, this element will describe the fields it contains.</td>
<td>Spec Object</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Location

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETag</td>
<td>Location</td>
<td>Specifies the ETag of the object inventory; length: 1–1,024 bytes.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>ObjectArn</td>
<td>Location</td>
<td>Specifies the unique resource ID of the object inventory; length: 1–1,024 bytes.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>ObjectVersionId</td>
<td>Location</td>
<td>Specifies the version ID of the object inventory; length: 1–1,024 bytes.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>

### Spec

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fields</td>
<td>Spec</td>
<td>Describes the fields in the inventory. If Format is <code>COSBatchOperations.CSV_V1</code>, this element is required to specify the fields in the CSV file. Valid values: <code>Ignore</code>, <code>Bucket</code>, <code>Key</code>, <code>VersionId</code>.</td>
<td>Array of Strings</td>
<td>No</td>
</tr>
</tbody>
</table>
### Cloud Object Storage

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Spec</td>
<td>Specifies the format of the object inventory. Valid values: COSBatchOperations_CSV_V1, COSInventoryReport_CSV_V1</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Operation

Operation involves multiple operations, but you can only specify one at a time. Currently, only the PUT Object-Copy operation is supported, so only the COSPutObjectCopy element can be contained.

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSPutObjectCopy</td>
<td>Operation</td>
<td>Specifies the parameters for batch replication of objects in the inventory.</td>
<td>COSPutObjectCopy Object</td>
<td>No</td>
</tr>
</tbody>
</table>

### COSPutObjectCopy

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessControlGrants</td>
<td>COSPutObjectCopy</td>
<td>Controls the access permissions of an object.</td>
<td>AccessControlGrants Object</td>
<td>No</td>
</tr>
<tr>
<td>CannedAccessControlList</td>
<td>COSPutObjectCopy</td>
<td>Defines the ACL attribute of an object. Valid values: private, public-read.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>MetadataDirective</td>
<td>COSPutObjectCopy</td>
<td>Whether to copy the source file metadata. Enumerated values: Copy, Replaced. Default value: Copy. If the flag is Copy, the source file metadata will be copied. If the flag is Replaced, the metadata will be modified according to the header information of the request.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>ModifiedSinceConstraint</td>
<td>COSPutObjectCopy</td>
<td>If the object is modified after the specified time, the operation will be performed; otherwise, 412 will be returned.</td>
<td>Timestamp</td>
<td>No</td>
</tr>
<tr>
<td>UnModifiedSinceConstraint</td>
<td>COSPutObjectCopy</td>
<td>If the object is not modified after the specified time, the operation will be performed; otherwise, 412 will be returned.</td>
<td>Timestamp</td>
<td>No</td>
</tr>
<tr>
<td>NewObjectMetadata</td>
<td>COSPutObjectCopy</td>
<td>Configures the metadata of an object.</td>
<td>NewObjectMetadata</td>
<td>No</td>
</tr>
<tr>
<td>StorageClass</td>
<td>COSPutObjectCopy</td>
<td>Sets the storage class of an object. Enumerated values: STANDARD, STANDARD_IA. Default value: STANDARD.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>TargetResource</td>
<td>COSPutObjectCopy</td>
<td>Sets the destination bucket for the replication. Please specify it with “qcs”, e.g. qcs::cos::ap-beijing::result-1250000000.</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### AccessControlGrants

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSGrant</td>
<td>AccessControlGrants</td>
<td>Configures access control.</td>
<td>COSGrant</td>
<td>No</td>
</tr>
</tbody>
</table>

### COSGrant

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grantee</td>
<td>COSGrant</td>
<td>Specifies to which user a permission is granted.</td>
<td>Grantee</td>
<td>Yes</td>
</tr>
<tr>
<td>Node Name</td>
<td>Parent Node</td>
<td>Description</td>
<td>Type</td>
<td>Required</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>Permission</td>
<td>COSGrant</td>
<td>Specifies the permission to be granted. Enumerated values: READ, WRITE, FULL_CONTROL.</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Grantee**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayName</td>
<td>Grantee</td>
<td>Username</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>Identifier</td>
<td>Grantee</td>
<td>User ID in qcs format (UIN), e.g. qcs::cam::uin/100000000001:uin/100000000001.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>TypeIdentifier</td>
<td>Grantee</td>
<td>Specifies an identifier type. Currently, only user ID is supported. Enumerated value: ID.</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**NewObjectMetadata**

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>CacheControl</td>
<td>NewObjectMetadata</td>
<td>Cache directives as defined in RFC 2616, which will be stored in the object metadata.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ContentDisposition</td>
<td>NewObjectMetadata</td>
<td>Filename as defined in RFC 2616, which will be stored in the object metadata.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ContentEncoding</td>
<td>NewObjectMetadata</td>
<td>Encoding format as defined in RFC 2616, which will be stored in the object metadata.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ContentType</td>
<td>NewObjectMetadata</td>
<td>Content type as defined in RFC 2616, which will be stored in the object metadata.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>HttpExpiresDate</td>
<td>NewObjectMetadata</td>
<td>Cache expiration time as defined in RFC 2616, which will be stored in the object metadata.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>SSEAlgorithm</td>
<td>NewObjectMetadata</td>
<td>Server-side encryption algorithm. Currently only AES256 is supported.</td>
<td>String</td>
<td>No</td>
</tr>
</tbody>
</table>
### UserMetadata

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserMetadata</td>
<td>NewObjectMetadata</td>
<td>Includes user-defined object metadata.</td>
<td>Array of Key and Value</td>
<td>No</td>
</tr>
</tbody>
</table>

### Report

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>Report</td>
<td>Bucket to which a job report is delivered.</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Enabled</td>
<td>Report</td>
<td>Specifies whether to output a job report.</td>
<td>Boolean</td>
<td>Yes</td>
</tr>
<tr>
<td>Prefix</td>
<td>Report</td>
<td>Job report prefix; length: 0-256 bytes.</td>
<td>String</td>
<td>No</td>
</tr>
<tr>
<td>ReportScope</td>
<td>Report</td>
<td>Determines whether to record information on all operations or only failed operations in a job report. Valid values: AllTasks, FailedTasksOnly.</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### ProgressSummary

<table>
<thead>
<tr>
<th>Node Name</th>
<th>Parent Node</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumberOfTasksFailed</td>
<td>ProgressSummary</td>
<td>Number of failed operations.</td>
<td>Integer</td>
</tr>
<tr>
<td>NumberOfTasksSucceeded</td>
<td>ProgressSummary</td>
<td>Number of successful operations.</td>
<td>Integer</td>
</tr>
<tr>
<td>TotalNumberOfTasks</td>
<td>ProgressSummary</td>
<td>Total number of operations.</td>
<td>Integer</td>
</tr>
</tbody>
</table>
Error Response

This document describes the error responses you may see when using the batch operation feature.

Error Response Format

HTTP/1.1 400
<Error>
  <Code>string</Code>
  <Message>string</Message>
  <Resource>string</Resource>
  <RequestId>string</RequestId>
  <TraceId>string</TraceId>
</Error>

Common Error Codes

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Status Code</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidArgument</td>
<td>Empty parameter</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Invalid parameter</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>x-cos-appid cannot be empty</td>
<td>400</td>
<td>Any</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>Priority must be an integer between 0 and 2,147,483,647</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>The value of the requestedJobStatus parameter must be Cancelled or Ready</td>
<td>400</td>
<td>UpdateJobStatus</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>The jobStatuses parameter is invalid</td>
<td>400</td>
<td>ListJobs</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>The maxResults parameter must be a positive integer</td>
<td>400</td>
<td>ListJobs</td>
</tr>
<tr>
<td>InvalidArgument</td>
<td>The nextToken parameter is invalid</td>
<td>400</td>
<td>ListJobs</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Invalid request</td>
<td>400</td>
<td>Any</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Empty request body</td>
<td>400</td>
<td>Any</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>No job ID is provided</td>
<td>400</td>
<td>Any</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>No job priority is provided</td>
<td>400</td>
<td>UpdateJobPriority</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>The ClientRequestToken parameter already exists</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>The specified job has already been completed</td>
<td>400</td>
<td>UpdateJobStatus</td>
</tr>
<tr>
<td>InvalidRequest</td>
<td>Error with the job status change</td>
<td>400</td>
<td>UpdateJobStatus</td>
</tr>
<tr>
<td>InternalError</td>
<td>Failed to format the XML response</td>
<td>500</td>
<td>Any</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>Invalid request format</td>
<td>400</td>
<td>Any</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>Incorrect Manifest format</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>Incorrect Operation format</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>MalformedXML</td>
<td>Incorrect Report format</td>
<td>400</td>
<td>Createjob</td>
</tr>
<tr>
<td>NoSuchJob</td>
<td>The specified job does not exist</td>
<td>404</td>
<td>Describejob</td>
</tr>
<tr>
<td>NoSuchJob</td>
<td>The specified job has already been completed</td>
<td>404</td>
<td>UpdateJobStatus, UpdateJobPriority</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
<td>Status Code</td>
<td>API</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>Error with data persistence</td>
<td>500</td>
<td>Any</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>Error loading persistent data</td>
<td>500</td>
<td>Any</td>
</tr>
<tr>
<td>ServiceUnavailable</td>
<td>Unable to create a job</td>
<td>500</td>
<td>CreateJob</td>
</tr>
<tr>
<td>TooManyJobs</td>
<td>The number of jobs has reached the upper limit, making the server unavailable</td>
<td>500</td>
<td>CreateJob</td>
</tr>
</tbody>
</table>