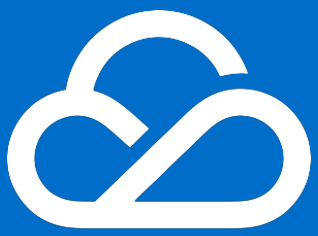
# Cloud Object Storage

# API Documentation

# Product Introduction



Tencent Cloud

## Copyright Notice



## Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

API Documentation

# Introduction

The XML APIs of Tencent Cloud Object Storage (COS) service are a kind of lightweight interfaces without connection state. You can call these APIs to send requests and receive responses directly via http/https, in order to interact with the backend of Tencent Cloud Object Storage. The contents of both requests and responses for these APIs are in XML format.

> Note:
> Currently, the available Regions of COS have different values for XML APIs and JSON APIs, and the corresponding region fields are required when using different APIs and their SDKs. For more information, see the document Available Regions.
> In order to use the XML APIs of Tencent Cloud Object Storage service more efficiently, please read Request Signature carefully before reviewing other API documents.

## Terminology Information

Some main concepts and terms appear in the text:

| Name | Description |
| --- | --- |
| APPID | A unique resource ID in user dimension owned by a developer when accessing COS services, which is used to indicate resources |
| SecretId | The project identity ID owned by a developer, which is used for identity authentication |
| SecretKey | The project identity key owned by a developer |
| Bucket | The container used to store data in COS |
| Object | The specific file stored in COS, which is the basic entity that is stored |
| Region | The region information in domain name. For enumerated values, please see the document Available Regions, such as: ap-beijing, ap- |

| Name | Description |
| --- | --- |
| | hongkong, eu-frankfurt, etc. |
| ACL | Access Control List, which refers to the access control information list of specified Buckets or Objects |
| CORS | Cross-Origin Resource Sharing, which refers to the HTTP request for resources from a different domain |
| Multipart Uploads | Refers to a multipart upload mode provided by Tencent Cloud COS service for uploading files |

## Quick Start

To use the Tencent Cloud object storage APIs, you need to follow these steps first:

1. Purchase the Tencent Cloud Object Storage (COS) service
2. Create a Bucket in Tencent Cloud Object Storage Console
3. Obtain APPID, SecretId, and SecretKey on the console Personal API Key page
4. Write an algorithm program for requesting signature (or use any of server-side SDKs)
5. Calculate the signature and call API to perform operation

## APIs of Other Versions

### JSON APIs

JSON API is the API provided by Tencent Cloud COS service for users to access COS before launching the XML API, and the upload domain name is [Region].file.myqcloud.com. JSON APIs and standard XML APIs have the same underlying infrastructure, and thus data interoperability is possible and they can be cross-used. However, they're not compatible with each other and have different domains. After the XML API service of Tencent Cloud COS is launched, it is recommended that you use the XML API interface. JSON APIs will be kept in a state of maintenance, and they will be available for use but no new features will be added.

# Common Request Headers

## Description

This document describes Common Request Headers to be used when using APIs. The headers described below will not be discussed in later API documents.

### List of Request Headers

| Header Name | Description | Type | Required |
| --- | --- | --- | --- |
| Authorization | Contain authentication information, signature information used to verify the validity of requests. This header is not required for files that can be read by public users. | String | No |
| Content-Length | HTTP request content length defined in RFC 2616 (bytes), commonly used in API operations of PUT type. | String | No |
| Content-Type | HTTP request content type defined in RFC 2616 (MIME), for example: text/plain | String | No |
| Content-MD5 | 128-bit content MD5 check value encoded using Base64, defined in RFC 1864. This header is used to check whether the file content has changed. | String | No |
| | | | |

| Header Name | Description | Type | Required |
| --- | --- | --- | --- |
| Date | GMT time defined in RFC 1123, for example: Wed, 30 Mar. 2016 23:00:00 GMT. | String | No |
| Expect | If Expect: 100-continue is used, the request content will not be sent until the receipt of response from server. This option can be used to check whether a header is valid, without the need to send the data content. Valid value: 100-continue. | String | No |
| Host | Request host, in a form of -.cos..myqcloud.com | String | Yes |

# Common Returned Headders

## Description

This document describes Common Response Headers that will appear when using APIs. The headers described below will not be explained in later API documents.

## List of Response Headers

| Header Name | Description | Type |
| --- | --- | --- |
| Content-Length | HTTP request content length defined in RFC 2616 (Bytes) | String |
| Content-Type | HTTP request content type defined in RFC 2616 (MIME) | String |
| Connection | Declare connection status between client and server. Enumerated values: keep-alive, close | Enum |
| Date | Response time of the server, subject to the GMT time defined in RFC 1123. | String |
| Etag | ETag (Entity Tag) is an information tag used to identify Object content upon creation of Object. This parameter may return values other than MD5, depending on different situations of the requests. ETag value can be used to check whether the Object content has changed. | String |
| Server | Name of the server that created the request. Default value: tencent-cos | String |
| x-cos-request-id | When a request is sent, the server | String |

| Header Name | Description | Type |
| --- | --- | --- |
| | will automatically generate an ID for the request. | |
| x-cos-trace-id | When a request encounters an error, the server will automatically generate an ID for the error. | String |

# Error Codes

## Overview

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

## Format of Returned Error Message

### Response Header

Content-Type: application/xml

Corresponding HTTP status code: 3XX, 4XX, 5XX

### Response Content

Syntax Format

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>[Error code]</Code>
  <Message>[Error message]</Message>
  <Resource>[Resource Address]</Resource>
  <RequestId>[Request ID]</RequestId>
  <TraceId>[Error ID]</TraceId>
</Error>
```

Element Description

| Element Name | Description | Type |
| --- | --- | --- |
| Error | Contain all error information. | Container |
| Code | Error codes are used to locate a | String |

| Element Name | Description | Type |
| --- | --- | --- |
| | unique error condition and determine scenario of the error. Error codes are described in detail below. | |
| Message | Contain detailed error information. | String |
| Resource | Resource address: Bucket address or Object address. | String |
| RequestId | The server will automatically generate a unique ID for the request when the request is sent. When a problem occurs, request-id can help COS locate the problem faster. | String |
| TraceId | When a request encounters an error, the server will automatically generate a unique ID for the error. When a problem occurs, trace-id can help COS locate the problem faster. When a request encounters an error, one trace-id corresponds to one request-id. | String |

## Error Codes

3XX Errors

| Error Code | Description | HTTP Status Code |
| --- | --- | --- |
| PermanentRedirect | This resource has been moved to another location permanently, please use HTTP Location to redirect to the new location | 301 Moved Permanently |
| | | |

| Error Code | Description | HTTP Status Code |
| --- | --- | --- |
| TemporaryRedirect | This resource has been moved to another location temporarily, please use HTTP Location to redirect to the new location | 302 Moved Temporarily |
| Redirect | Temporary redirection | 307 Moved Temporarily |
| TemporaryRedirect | You will be redirected temporarily during DNS update process | 307 Moved Temporarily |

4XX Errors

| Error Code | Description | HTTP Status Code |
| --- | --- | --- |
| BadDigest | The provided x-cos-SHA-1 value is different from the SHA-1 value of the file received by the server end | 400 Bad Request |
| EntityTooSmall | The size of the file to be uploaded is smaller than the required minimum size, which is common for multipart upload | 400 Bad Request |
| EntityTooLarge | The size of the file to be uploaded is larger than the required maximum size | 400 Bad Request |
| ImcompleteBody | The actual content length of the request is inconsistent with the specified Conent-Length | 400 Bad Request |
| IncorrectNumberOfFilesInPostRequest | Only one file is allowed to be uploaded at a time for a Post request | 400 Bad Request |
| InlineDataTooLarge | The size of inline data is larger than the required maximum size | 400 Bad Request |
| InvalidArgument | URI is invalid | 400 Bad Request |
| InvalidBucketName | Bucket name is invalid | 400 Bad Request |
| | | |

| Error Code | Description | HTTP Status Code |
| --- | --- | --- |
| InvalidDigest | x-cos-SHA-1 value is invalid | 400 Bad Request |
| InvalidPart | Part is missing or SectionID is invalid | 400 Bad Request |
| InvalidPolicyDocunment | Policy configuration file is invalid | 400 Bad Request |
| InvalidURI | URI is invalid | 400 Bad Request |
| KeyTooLong | File path is too long | 400 Bad Request |
| MalformedACLError | Described ACL policy does not comply with XML syntax | 400 Bad Request |
| MalformedPOSTRequest | The Body content of the POST request is invalid | 400 Bad Request |
| MalformedXML | "body" in XML format does not comply with XML syntax | 400 Bad Request |
| MaxMessageLengthExceeded | Request is too long | 400 Bad Request |
| MaxPostPreDataLengthExceeded Error | The data prefix of the POST request is too long, this usually happens for multipart upload operations | 400 Bad Request |
| MatadataTooLarge | The size of metadata is larger than the required maximum size | 400 Bad Request |
| MissingRequestBodyError | Request Body is missing | 400 Bad Request |
| MissingSecurityHeader | Required Header is missing | 400 Bad Request |
| MissingContentMD5 | Content-MD5 is missing in request header | 400 Bad Request |
| MissingAppid | Appid is missing in request header | 400 Bad Request |
| MissingHost | Host is missing in request header | 400 Bad Request |
| RequestIsNotMultiPartContent | The Content-Type of the POST request is invalid | 400 Bad Request |
| RequestTimeOut | Read timeout. Check whether the network is too slow or number of concurrent file uploads is too large | 400 Bad Request |
| TooManyBucket | The number of Buckets exceeded | 400 Bad Request |

| Error Code | Description | HTTP Status Code |
| --- | --- | --- |
| | the limit (200) | |
| UnexpectedContent | Relevant content is not supported for the request | 400 Bad Request |
| UnresolvableGrantByUID | The provided UID does not exist | 400 Bad Request |
| UserKeyMustBeSpecified | The path must be specified for the POST operation performed against a Bucket | 400 Bad Request |
| AccessDenied | Access denied due to invalid signature or permission | 403 Forbidden |
| AccountProblem | This operation has been denied by your account | 403 Forbidden |
| InvaildAccessKeyId | AccessKey does not exist | 403 Forbidden |
| InvalidObjectState | Request content is in conflict with Object attribute | 403 Forbidden |
| InvalidSecurity | Signature string is invalid | 403 Forbidden |
| RequestTimeTooSkewed | Request time is beyond the valid period of permission | 403 Forbidden |
| SignatureDoesNotMatch | Incorrect signature | 403 Forbidden |
| NoSuchBucket | Specified Bucket does not exist | 404 Not Found |
| NoSuchUpload | Specified multipart upload does not exist | 404 Not Found |
| NoSuchBucket | Specified Bucket policy does not exist | 404 Not Found |
| MethodNotAllowed | The HTTP method is not supported by this resource | 405 Method Not Allowed |
| BucketAlreadyExists | BucketName specified by CreateBucket is already in use. Select another BucketName | 409 Conflict |
| BucketNotEmpty | Delete files and unfinished multipart upload tasks before performing DeleteBucket operation | 409 Conflict |
| InvalidBucketState | Bucket status conflicts with | 409 Conflict |

| Error Code | Description | HTTP Status Code |
| --- | --- | --- |
| | operation request, for example, multi-version management conflicts with cross-region duplication | |
| actionAborted | This operation is not supported by specified resource | 409 Conflict |
| MissingContentLength | Header Content-Length is missing | 411 Length Required |
| PreconditionFailed | Precondition matching failed | 412 Precondition |
| InvalidRange | Requested file range is invalid | 416 Requested Range Not Satisfiable |
| InvalidSHA1Digest | sha1 of the request content is invalid | 400 Bad Request |
| NoSuchUpload | "uploadid" specified when performing multipart upload operation does not exist | 400 Bad Request |
| InvalidPart | Part is missing | 400 Bad Request |
| InvalidPartOrder | The numbers of uploaded parts are discontinuous | 400 Bad Request |
| ObjectNotAppendable | Specified file is not appendable | 400 Bad Request |
| AppendPositionErr | Append: file length is inconsistent with position | 400 Bad Request |
| NoSuchVersion | Specified version does not exist | 400 Bad Request |
| NoLifecycle | Lifecycle does not exist | 400 Bad Request |
| PreconditionFailed | Precondition matching failed | 400 Bad Request |
| UnexpectedContent | Relevant content is not supported for the request | 400 Bad Request |
| MultiBucketNotSupport | Only one bucket is configured for cross-region duplication | 400 Bad Request |
| NotSupportedStorageClass | Specified storage type is invalid | 400 Bad Request |
| InvalidAccessKeyId | AccessKey does not exist | 403 Forbidden |
| ExpiredToken | Signature string expired | 403 Forbidden |

5XX Errors

| Error code | Description | HTTP status code |
| --- | --- | --- |
| InternalError | Internal error occurred on the server end | 500 Internal Server |
| NotImplemented | A method in the Header cannot be implemented | 501 Not Implemented |
| ServiceUnavailable | Internal error on the server. Try again | 503 Service Unavailable |
| SlowDown | Please reduce access frequency | 503 Slow Down |

Other Errors

| Error code | Description | HTTP status code |
| --- | --- | --- |
| InvaildAddressingHeader | Anonymous access is required | N/A |

# Overview

Below are Tencent Cloud Object Storage (COS) service related APIs and their descriptions:

## About Service Operation

| API | Description |
| --- | --- |
| Get Service | List all Buckets under this account |

## About Bucket Operations

| API | Description |
| --- | --- |
| Get Bucket | List some or all of the Objects under the specified Bucket |
| Get Bucket ACL | Obtain the ACL table of the Bucket |
| Get Bucket CORS | Obtain the cross-domain access configuration of the Bucket |
| Get Bucket Location | Obtain the region of the Bucket |
| Get Bucket Lifecycle | Read lifecycle management configurations |
| Put Bucket | Create a Bucket under the specified account |
| Put Bucket ACL | Write to the ACL table of the Bucket |
| Put Bucket CORS | Configure the cross-domain access permission of the Bucket |
| Put Bucket Lifecycle | Set the features for lifecycle management |
| Delete Bucket | Delete the Bucket under the specified account |
| Delete Bucket CORS | Delete the cross-domain access configuration of the Bucket |
| Delete Bucket Lifecycle | Delete lifecycle management |
| Head Bucket | Confirm whether a specified Bucket exists under the specified account |
| List Multipart Uploads | Query the ongoing multipart upload |

## About Object Operations

| API | Description |
| --- | --- |
| Append Object | Upload an Object (file/object) to the specified Bucket via multipart upload method |
| Get Object | Download an Object (file/object) to the local computer |
| Get Object ACL | Obtain the ACL table of the Object (file/object) |
| Put Object | Upload an Object (file/object) to the specified Bucket |
| Put Object ACL | Write to the ACL table of the Object (file/object) |
| Delete Object | Delete the specified Object (file/object) in the Bucket |
| Delete Multiple Object | Delete Objects (files/objects) in batch in the Bucket |
| Head Object | Obtain the meta information of the Object |
| Options Object | A preflight request for cross-domain access |
| Initiate Multipart Upload | Initialize the Multipart Upload operation |
| List Multipart Uploads | Multipart upload files |
| List Parts | Query the uploaded parts in a specific multipart upload operation |
| Complete Multipart Upload | Complete the multipart upload of the entire file |
| Abort Multipart Upload | Abort a multipart upload operation and delete the uploaded parts |
| Put Object Copy | Copy a file from the source path to the destination path |

# Service APIs

# Bucket APIs

## Delete Bucket CORS

## Description

Delete Bucket CORS API request is used to delete configuration information of cross-domain access.

## Request

Syntax:

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

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

### Request Line

DELETE /?cors HTTP/1.1

This API allows DELETE request.

### Request Header

Common Header

This request operation is implemented using common request header. For more information, please see Common Request Headers chapter.

Non-common Header

No particular request header information for this request operation.

### Request Body

The request body of this request is null.

## Response

### Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this response.

### Response Body

Null is returned for the response body.

## Practical Case

### Request

DELETE /?cors HTTP/1.1

Host: arlenhuangtestsgnoversion-1251668577.cos.ap-beijing.myqcloud.com
Date: Tue, 23 Oct 2016 21:32:00 GMT
Authorization:
q-sign-algorithm=sha1&q-
ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUj
fGFO&q-sign-time=1484816036;32557712036&q-key-time=1484816036;32557712036&q-header-
list=host&q-url-param-list=cors&q-signature=e92eecbf0022fe7e5fd39b2c500b22da062be50a

## Response

HTTP/1.1 204 No Content
Content-Type: application/xml
Content-Length: 405
Connection: keep-alive
Date: Tue, 23 Oct 2016 21:32:00 GMT
Server: tencent-cos
x-cos-request-id: NTg4MDdlYWNfOTgxZjRlXzZhYTlfZjAz
x-cos-trace-id: OGVmYzZiMmQzYjA2OWNhODk0NTRkMTBiOWVmMDAxODczNTBmNjMwZmQ0MT
ZkMjg0NjlkNTYyNmY4ZTRkZTk0N2M2MTdkZGZlMGNhOWQyYjk3MWNmNWNkYzFhMjQzNzRiZTE1
NjgzNzFhOGI5M2EwZDMyNGM4Y2ZmMzhiNTllMjk=

# Get Bucket ACL

## Description

Get Bucket ACL API is used to obtain ACL (access control list) of Bucket, that is, the access permission control list of Bucket. Only the Bucket owner has permission to use this API.

## Request

Syntax:

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

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

### Request Line

GET /?acl HTTP/1.1

This API allows GET request.

### Request Header

Common Header

This request operation is implemented using common request header. For more information, please

see Common Request Headers chapter.

Non-common Header

Required Header
This request operation is implemented using the following required headers:

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| Authorization | Signature string | String | Yes |

### Request Body

The request body of this request is null.

## Response

### Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this response.

### Response Body

application/xml data is returned for the response body, including the complete node data, as show below:

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

```
<ID>qcs::cam::uin/<OwnerUin>:uin/<SubUin></ID>
<DisplayName>qcs::cam::uin/<OwnerUin>:uin/<SubUin></DisplayName>
</Owner>
<AccessControlList>
<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RootAccount">
<ID>qcs::cam::uin/<OwnerUin>:uin/<SubUin></ID>
<DisplayName>qcs::cam::uin/<OwnerUin>:uin/<SubUin></DisplayName>
</Grantee>
<Permission></Permission>
</Grant>
<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RootAccount">
<ID>qcs::cam::uin/<OwnerUin>:uin/<SubUin></ID>
<DisplayName>qcs::cam::uin/<OwnerUin>:uin/<SubUin></DisplayName>
</Grantee>
<Permission></Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

Detailed data content is shown as below:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| AccessControlPolicy | None | Container for saving results of Get Bucket ACL | Container |

Content of Container node AccessControlPolicy:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| Owner | AccessControlPolicy | Information of Bucket owner | Container |
| AccessControlList | AccessControlPolicy | Information of | Container |

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| | | authorized account and permissions | |

Content of Container node Owner:

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| ID | AccessControlPolicy.Owner | | |

# Get Bucket CORS

## Description

Get Bucket CORS is used to read cross-domain access configurations.

## Request

### Request Syntax

GET /?cors HTTP 1.1
Host:<Bucketname>-<UID>.<Region>.myqcloud.com
Date:date
Authorization: Auth

### Request Parameter

No particular request parameters

### Request Header

No particular request headers. Please refer to "Common Request Headers" for other headers

### Request Content

No request content

## Returned Value

### Response Header

No particular response headers. Please refer to "Common Response Headers" for other headers

## Response Content

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| CORSConfiguration | Describe all information regarding cross-domain configurations, may contain up to 100 CORSRule entries | Container | Yes |
| CORSRule | Information of a single configuration<br>Parent node: CORSRule | Container | Yes |
| ID | Name of the rule, optional<br>Parent node: CORSRule | String | No |
| AllowedMethod | Allowed HTTP operations, enumerated values include Get, Put, Head, Post, Delete<br>Parent node: CORSRule | Enum | Yes |
| AllowedOrigin | Allowed access source. Wildcard "*" is supported<br>Parent node: CORSRule | String | Yes |
| AllowedHeader | When sending an OPTIONS request, notify the server end about which custom HTTP request headers are allowed to be used by subsequent requests<br>Parent node: CORSRule | String | No |
| MaxAgeSeconds | Configure the valid period for the results obtained by OPTIONS | Integer | No |

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| | request<br>Parent node: CORSRule | | |
| ExposeHeadr | Configure what kind of custom header information from the server end can be received by the browser<br>Parent node: CORSRule | String | No |

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

## Example

### Request

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

## Response

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 345
Connection: keep-alive
Date: Thu Jan 19 16:52:31 2017
Server: tencent-cos
x-cos-request-id: NTg4MDdlNGZfNDYyMDRlXzM0YWFfZTBh

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

# Object APIs

## Abort Multipart Upload

## Description

Abort Multipart Upload is used to abort a multipart upload operation and delete parts that are already uploaded. When Abort Multipart Upload is called, the Upload Parts returns failure to any request that is using the Upload Parts. "404 NoSuchUpload" is returned if the UploadID does not exist.

> Note:
> It is recommended that you complete multipart upload in time or abort the upload operation for the reason that parts that have been uploaded but not aborted can take up storage, incurring cost.

## Request

Syntax:

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

> Authorization: Auth String (For more information, please see Request Signature chapter)

### Request Line

DELETE /ObjectName?uploadId=UploadId HTTP/1.1

This API allows DELETE request.

## Request Parameters

Example of request line that contains all request parameters.

DELETE /ObjectName?uploadId=UploadId HTTP/1.1

See the details below:

| Parameter Name | Description | Type | Required |
| --- | --- | --- | --- |
| UploadID | Indicate the ID of current multipart upload. You can obtain an uploadid when you use the API "Initiate Multipart Upload" to initiate multipart upload. This ID exclusively identifies this multipart data, and the relative position of this multipart in the entire file | String | Yes |

## Request Header

Common Header

This request operation is implemented using common request header. For more information, please see Common Request Headers chapter.

Non-common Header

No particular request header information for this request operation.

### Request Body

The request body of this request is null.

## Response

### Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this request operation.

### Response Body

The response body of this request is null.

## Practical Case

### Request

```
DELETE
/ObjectName?uploadId=1484727270323
ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e HTTP/1.1
Host: arlenhuangtestsgnoversion-1251668577.cos.ap-beijing.myqcloud.com
```

Date: Tue, 26 Oct 2013 21:22:00 GMT

Authorization:

q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484728626;32557624626&q-key-time=1484728626;32557624626&q-header-list=host&q-url-param-list=uploadId&q-signature=2d3036b57cade4a257b48a3a5dc922779a562b18

## Response

HTTP/1.1 200 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

# Complete Multipart Upload

## Description

Complete Multipart Upload API request is used to complete the entire multipart upload. You must use this API to complete the multipart upload operation of the entire file when you have uploaded all parts using Upload Parts. When using this API, you need to provide the PartNumber and ETag for every part in request Body, to verify the accuracy of parts.
The merging of parts is required and takes several minutes, thus COS returns status code 200 immediately when the merging process begins. During merging, COS may returns blank information periodically to keep the connection active, until the merging process completes, upon which the COS will return the content of the merged parts in Body.
When this API is called, "400 EntityTooSmall" is returned if the uploaded part is smaller than 1 MB.
"400 InvalidPart" is returned if the numbers of uploaded parts are discontinuous.
"400 InvalidPartOrder" is returned if the part information entries in the request Body are not sorted in ascending order according to their numbers.
"404 NoSuchUpload" is returned if the UploadId does not exist when this API is called.

> Note:
> It is recommended that you complete multipart upload in time or abort the upload operation for the reason that parts that have been uploaded but not aborted can take up storage, incurring cost.

## Request

Syntax:

```
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 (For more information, please see Request Signature chapter)

## Request Line

POST /ObjectName?uploadId=UploadId HTTP/1.1

This API allows POST request.

Request parameter

Example of request line that contains all request parameters.

POST /ObjectName?uploadId=UploadId HTTP/1.1

See the details below:

| Parameter Name | Description | Type | Required |
| --- | --- | --- | --- |
| uploadId | Indicate the ID of current multipart upload. You can obtain an uploadid when you use the API "Initiate Multipart Upload" to initiate multipart upload. This ID exclusively identifies this multipart | String | Yes |

| Parameter Name | Description | Type | Required |
| --- | --- | --- | --- |
| | data, and the relative position of this multipart in the entire file | | |

## Request Header

Common Header

This request operation is implemented using common request header. For more information, please see Common Request Headers chapter.

Non-common Header

No particular request header information for this request operation.

## Request Body

The specific nodes of the request body for this API request are:

```
<CompleteMultipartUpload>
  <Part>
  <PartNumber></PartNumber>
  <ETag></ETag>
  </Part>
  ...
</CompleteMultipartUpload>
```

Detailed data content is shown as below:

| Node Name (Keyword) | Parent Node | Description | Type | Required |
| --- | --- | --- | --- | --- |
| CompleteMultipartUpload | None | Used to describe all information of the | Container | Yes |

| Node Name (Keyword) | Parent Node | Description | Type | Required |
|---|---|---|---|---|
| | | current multipart upload operation | | |

Content of Container node CompleteMultipartUpload:

| Node Name (Keyword) | Parent Node | Description | Type | Required |
|---|---|---|---|---|
| Part | CompleteMultipartUpload | Used to describe information of every part in the current multipart upload operation | Container | Yes |

Content of Container node Part:

| Node Name (Keyword) | Parent Node | Description | Type | Required |
|---|---|---|---|---|
| PartNumber | CompleteMultipartUpload.Part | Part number | String | Yes |
| ETag | CompleteMultipartUpload.Part | MD5 algorithm check value for every part file | String | Yes |

# Response

## Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this response.

## Response Body

application/xml data is returned for the response body, including the complete node data, as show below:

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

Detailed data content is shown as below:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| CompleteMultipartUploadResult | None | Indicate all the returned information | Container |

Content of Container node CompleteMultipartUploadResult:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| Location | CompleteMultipartUploadResult | Domain name for public network access of the created Object | URL |
| Bucket | CompleteMultipartUploadResult | The target Bucket for multipart upload | String |
| Key | CompleteMultipartUploadResult | Name of Object | String |
| ETag | CompleteMultipartUploa | MD5 algorithm check | String |

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| | dResult | value for the merged file | |

## Practical Case

### Request

POST
/ObjectName?uploadId=1484728886e63106e87d8207536ae8521c89c42a436fe23bb58854a7bb5e87b7d77d4ddc48 HTTP/1.1
Host: arlenhuangtestsgnoversion-1251668577.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484729794;32557625794&q-key-time=1484729794;32557625794&q-header-list=host&q-url-param-list=uploadId&q-signature=23627c8fddb3823cce4257b33c663fd83f9f820d
Content-Length: 155
Content-Type: application/x-www-form-urlencoded

### Response

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: NTg3ZjJlMjVfNDYyMDRlXzM0YzRfMjc1

```
<CompleteMultipartUploadResult>
  <Location>
arlenhuangtestsgnoversion-1251668577.cos.ap-beijing.myqcloud.com/ObjectName</Location>
  <Bucket>arlenhuangtestsgnoversion</Bucket>
  <Key>ObjectName</Key>
```

<ETag>"3a0f1fd698c235af9cf098cb74aa25bc"</ETag>
</CompleteMultipartUploadResult>

# Get Object ACL

## Description

Get Object ACL API is used to obtain access permission of an Object under a Bucket. Only the Bucket owner is allowed to perform the action.

## Request

Syntax:

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

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

### Request Line

GET /ObjectName?acl HTTP/1.1

This API allows GET request.

### Request Header

Common Header

This request operation is implemented using common request header. For more information, please

see Common Request Headers chapter.

Non-common Header

Required header

This request operation is implemented using the following required headers:

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| Authorization | Signature string | String | Yes |

### Request Body

The request body of this request is null.

## Response

### Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this response.

### Response Body

application/xml data is returned for the response body, including the complete node data, as show below:

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

```
<ID>qcs::cam::uin/<OwnerUin>:uin/<SubUin></ID>
<DisplayName>qcs::cam::uin/<OwnerUin>:uin/<SubUin></DisplayName>
</Owner>
<AccessControlList>
<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RootAccount">
<ID>qcs::cam::uin/<OwnerUin>:uin/<SubUin></ID>
<DisplayName>qcs::cam::uin/<OwnerUin>:uin/<SubUin></DisplayName>
</Grantee>
<Permission></Permission>
</Grant>
<Grant>
<Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RootAccount">
<ID>qcs::cam::uin/<OwnerUin>:uin/<SubUin></ID>
<DisplayName>qcs::cam::uin/<OwnerUin>:uin/<SubUin></DisplayName>
</Grantee>
<Permission></Permission>
</Grant>
</AccessControlList>
</AccessControlPolicy>
```

Detailed data content is shown as below:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| AccessControlPolicy | None | Container for saving results of Get Object ACL | Container |

Content of Container node AccessControlPolicy:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| Owner | AccessControlPolicy | Information of Object owner | Container |
| AccessControlList | AccessControlPolicy | Information of | Container |

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| | | authorized account and permissions | |

Content of Container node Owner:

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| ID | AccessControlPolicy.Owner | | |

# Initiate Multipart Upload

## Description

Initiate Multipart Upload request is used for the initialization of multipart upload. After the execution of this request, UploadId will be returned for the subsequent Upload Part requests.

## Request

Syntax:

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

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

### Request Line

POST /Object?uploads HTTP/1.1

This API allows POST request.

### Request Header

Common Header

This request operation is implemented using common request header. For more information, please

see Common Request Headers chapter.

Non-common Header

Recommended Header

This request operation is implemented using the following recommended request headers:

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| Cache-Control | The caching policy defined in RFC 2616, which will be saved as Object metadata. | String | No |
| Content-Disposition | The file name defined in RFC 2616, which will be saved as Object metadata. | String | No |
| Content-Encoding | The encoding format defined in RFC 2616, which will be saved as Object metadata. | String | No |
| Content-Type | The content type defined in RFC 2616, which will be saved as Object metadata. | String | No |
| Expires | The file name defined in RFC 2616, which will be saved as Object metadata. | String | No |
| x-cos-meta-* | The header information allowed to be defined by users, which will be returned as Object metadata. The size is limited to 2K. | String | No |
| | | | |

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| X-cos-storage-class | Set the storage class of Object. Enumerated values: Standard, Standard_IA, Nearline. The default is Standard (this is only supported for South China region) | String | No |

Permission-related headers

This request operation is implemented using header x-cos-acl in request PUT to set the access permission of Object. Object supports three access permissions: public-read-write, public-read and private. The default permission is private if not set. Users can also be clearly granted with permission of read, write or read-write separately. See the details below:

For more information on ACL, please see Put Bucket ACL.

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| x-cos-acl | Define the ACL attribute of Object. Valid values: private, public-read-write, public-read. Default value: private | String | No |
| x-cos-grant-read | Grant read permission to the authorized user. Format: x-cos-grant-read: id=" ", id=" "; when you need to authorize a sub-account, id="qcs::cam::uin/<OwnerUin>:uin/<SubUin>"; when you need to authorize the root | String | No |

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| | account, id="qcs::cam::uin/<OwnerUin>:uin/<OwnerUin>" | | |
| x-cos-grant-write | Grant write permission to the authorized user. Format: x-cos-grant-write: id=" ", id=" "; when you need to authorize a sub-account, id="qcs::cam::uin/<OwnerUin>:uin/<SubUin>"; when you need to authorize the root account, id="qcs::cam::uin/<OwnerUin>:uin/<OwnerUin>" | String | No |
| x-cos-grant-full-control | Grant read-write permission to the authorized user. Format: x-cos-grant-full-control: id=" ", id=" "; when you need to authorize a sub-account, id="qcs::cam::uin/<OwnerUin>:uin/<SubUin>"; when you need to authorize the root account, id="qcs::cam::uin/<OwnerUin>:uin/<OwnerUin>" | String | No |

## Request Body

The request body of this request is null.

# Response

## Response Header

### Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

### Specific Response Header

No particular response header for this response.

## Response Body

application/xml data is returned for the response body, including the complete node data, as show below:

```
<InitiateMultipartUploadResult>
  <Bucket></Bucket>
  <Key></Key>
  <UploadId></UploadId>
</InitiateMultipartUploadResult>
```

Detailed data content is shown as below:

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| InitiateMultipartUploadResult | None | Indicate all the returned information | Container |

Content of Container node InitiateMultipartUploadResult:

| Node Name (Keyword) | Parent Node | Description | Type |
| --- | --- | --- | --- |
| Bucket | InitiateMultipartUploadResult | The target Bucket of multipart upload | Container |
| Key | InitiateMultipartUploadResult | Name of Object | Container |
| UploadId | InitiateMultipartUploadResult | ID used in subsequent uploads | Container |

## Practical Case

### Request

POST /ObjectName?uploads HTTP/1.1
Host: arlenhuangtestsgnoversion-1251668577.cos.ap-beijing.myqcloud.com
Date: Fri, 10 Mar 2016 09:45:46 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484727259;32557623259&q-key-time=1484727259;32557623259&q-header-list=host&q-url-param-list=uploads&q-signature=b5f46c47379aeaee74be7578380b193c01b28045

### 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>arlenhuangtestsgnoversion</Bucket>
  <Key>ObjectName</Key>
  <UploadId>
```

148472727
0323ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e</UploadId>
</InitiateMultipartUploadResult>

# List Parts

## Description

List Parts is used to query the uploaded parts when uploading particular parts, which lists all the uploaded parts under a specified UploadId.

## Request

Syntax:

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

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

### Request Line

GET /ObjectName?uploadId=UploadId HTTP/1.1

This API allows GET request.

### Request Parameters

Example of request line that contains all request parameters.

GET

/ObjectName?uploadId=UploadId&encoding-type=EncodingType&max-parts=MaxParts&part-number-marker=PartNumberMarker HTTP/1.1

See the details below:

| Parameter Name | Description | Type | Required |
|---|---|---|---|
| uploadId | Indicate the ID of current multipart upload. You can obtain an uploadid when you use the API "Initiate Multipart Upload" to initiate multipart upload. This ID exclusively identifies this multipart data, and the relative position of this multipart in the entire file. | String | Yes |
| encoding-type | Indicate the encoding method of the returned value | String | No |
| max-parts | Maximum number of entries returned each time. Default is 1,000 | String | No |
| part-number-marker | Entries are listed in UTF-8 binary order by default, starting from marker | String | No |

## Request Header

Common Header

This request operation is implemented using common request header. For more information, please

see Common Request Headers chapter.

Non-common Header

No particular request header information for this request operation.

### Request Body

The request body of this request is null.

## Response

### Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this response.

### Response Body

application/xml data is returned for the response body, including the complete node data, as show below:

```
<ListPartsResult>
  <Bucket></Bucket>
  <Encoding-type></Encoding-type>
  <Key></Key>
  <UploadId></UploadId>
  <Owner>
```

```
<ID></ID>
<DisplayName></DisplayName>
</Owner>
<PartNumberMarker></PartNumberMarker>
<Initiator>
<ID></ID>
<DisplayName></DisplayName>
</Initiator>
<StorageClass></StorageClass>

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

Detailed data content is shown as below:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| ListPartsResult | None | Indicate information of the current multipart upload operation | Container |

Content of Container node ListPartsResult:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| Bucket | ListPartsResult | The target Bucket for | String |

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| | | multipart upload | |
| Encoding-type | ListPartsResult | Indicate the encoding method of the returned value | String |
| Key | ListPartsResult | Name of Object | String |
| UploadId | ListPartsResult | Indicate the ID of current multipart upload | String |
| Initiator | ListPartsResult | Indicate the information of the initiator of current upload | Container |
| Owner | ListPartsResult | Indicate the information of the owner of these parts | Container |
| StorageClass | ListPartsResult | Indicate the storage class of uploaded parts; enumerated values include Standard, Standard_IA, nearline | String |
| PartNumberMarker | ListPartsResult | Entries are listed using UTF-8 binary order by default, starting from marker | String |
| NextPartNumberMarker | ListPartsResult | If the returned entry is truncated, the returned NextMarker indicates the beginning of the next entry | String |
| MaxParts | ListPartsResult | Maximum number of entries returned at a time | String |
| IsTruncated | ListPartsResult | Indicate whether the returned entry is truncated. Boolean: | Boolean |

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| | | TRUE, FALSE | |
| Part | ListPartsResult | Indicate the information of each part | Container |

Content of Container node Initiator:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| ID | ListPartsResult.Initiator | Unique ID of the creator | String |
| DisplayName | ListPartsResult.Initiator | Name of creator | String |

Content of Container node Owner:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| ID | ListPartsResult.Owner | Unique ID of the user | String |
| DisplayName | ListPartsResult.Owner | Name of User | String |

Content of Container node Part:

| Node Name (Keyword) | Parent Node | Description | Type |
|---|---|---|---|
| PartNumber | ListPartsResult.Part | Part number | String |
| LastModified | ListPartsResult.Part | The last modification time of part | Date |
| ETag | ListPartsResult.Part | MD-5 algorithm check value of Object | String |
| Size | ListPartsResult.Part | Party size (in bytes) | String |

## Practical Case

### Request

GET
/coss3/test10M_2?uploadId=14846420620
b1f381e5d7b057692e131dd8d72dfa28f2633cfbbe4d0a9e8bd0719933545b0&max-parts=1 HTTP/1.1

Host:burning-1251668577.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization:q-sign-algorithm=sha1&q-ak=AKIDDNMEycgLRPI2axw9xa2Hhx87wZ3MqQCn&q-sign-time=1484643123;1484646723&q-key-time=1484643123;1484646723&q-header-list=host&q-url-param-list=max-parts;uploadId&q-signature=b8b4055724e64c9ad848190a2f7625fd3f9d3e87

## 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>burning</Bucket>
<Encoding-type/>
<Key>test10M_2</Key>
<UploadId>
148464206
20b1f381e5d7b057692e131dd8d72dfa28f2633cfbbe4d0a9e8bd0719933545b0</UploadId>
<Initiator>
<ID>123456789</ID>
<DisplyName>123456789</DisplyName>
</Initiator>
<Owner>
<ID>qcs::cam::uin/156545789:uin/156545789</ID>
<DisplyName>156545789</DisplyName>
</Owner>
<PartNumberMarker>0</PartNumberMarker>
<Part>
<PartNumber>1</PartNumber>

<LastModified>Tue Jan 17 16:43:37 2017</LastModified>
<ETag>"a1f8e5e4d63ac6970a0062a6277e191fe09a1382"</ETag>
<Size>5242880</Size>
</Part>
<NextPartNumberMarker>1</NextPartNumberMarker>
<StorageClass>Standard</StorageClass>
<MaxParts>1</MaxParts>
<IsTruncated>true</IsTruncated>
</ListPartsResult>

# Upload Part

## Description

Upload Part request is used to implement the multipart upload after initialization. The allowed number of parts is limited to 10,000, and the size of part should be between 1 MB and 5 GB.
You can obtain an uploadid when you use the API "Initiate Multipart Upload" to initiate multipart upload. This ID exclusively identifies this multipart data, and the relative position of this multipart in the entire file. Upload Part should be used with partNumber and uploadId. partNumber is the part No. and supports out-of-order upload.
If the uploadId and partNumber are the same, the parts uploaded later will overwrite the parts uploaded earlier. A 404 error "NoSuchUpload" will be returned if the uploadId does not exist.

## Request

Syntax:

```
PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: <BucketName>-<APPID>.cos.<Region>.myqcloud.com
Date: GMT Date
Content-Length: Size
Authorization: Auth String
```

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

### Request Line

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

This API allows PUT request.

## Request Parameters

Example of request line that contains all request parameters.

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

See the details below:

| Parameter Name | Description | Type | Required |
| --- | --- | --- | --- |
| partNumber | Indicate the No. of current multipart upload | String | Yes |
| uploadId | Indicate the ID of current multipart upload. You can obtain an uploadid when you use the API "Initiate Multipart Upload" to initiate multipart upload. This ID exclusively identifies this multipart data, and the relative position of this multipart in the entire file | String | Yes |

## Request Header

Common Header

This request operation is implemented using common request header. For more information, please see Common Request Headers chapter.

Non-common Header

Required Header

The following required request header is needed for the implementation of request operation. Details are shown below:

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| Content-Length | HTTP request content length defined in RFC 2616 (in bytes) | String | Yes |

Recommended Header

The following recommended request headers are recommended for implementation of this request operation. Details are shown below:

| Name | Description | Type | Required |
| --- | --- | --- | --- |
| Expect | HTTP request content length defined in RFC 2616 (in bytes) | String | Yes |
| Content-MD5 | 128-bit content MD5 check value encoded using Base64, defined in RFC 1864. This header is used to check whether the file content has changed | String | No |

### Request Body

The request body of this request is null.

## Response

### Response Header

Common Response Header

This response uses common response header. For more information, please see Common Response Headers chapter.

Specific Response Header

No particular response header for this response.

### Response Body

The response body of this response is null.

## Practical Case

### Request

```
PUT
/ObjectName?partNumber=1&uploadId=1484727270323ddb949d528c629235314a9ead80f0ba5d993a3d76b460e6a9cceb9633b08e HTTP/1.1
Host: arlenhuangtestsgnoversion-1251668577.cos.ap-beijing.myqcloud.com
Date: Wed, 18 Jan 2017 16:17:03 GMT
Authorization: q-sign-algorithm=sha1&q-ak=AKIDWtTCBYjM5OwLB9CAwA1Qb2ThTSUjfGFO&q-sign-time=1484727403;32557623403&q-key-time=1484727403;32557623403&q-header-list=host&q-url-param-list=partNumber;uploadId&q-signature=bfc54518ca8fc31b3ea287f1ed2a0dd8c8e88a1d
Content-Length: 10485760

[Object]
```

### Response

```
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: NTg3ZjI0NzlfOWIxZjRlXzZmNGJfMWYy