Object Storage Service

API Reference

API Reference

OSS API Documentation Overview

The Object Storage Service (OSS) is a cloud storage service provided by AliCloud, featuring massive capacity, security, low cost, and high reliability. Users can upload and download data anytime, anywhere and on any Internet device through a simple REST interface described in this documentation. With the OSS, users can create various multimedia sharing websites, network disks, personal and corporate data backups and enjoy other massive data–based services.

Before using these interfaces, please make sure that you fully understand the OSS product instructions, usage agreement, and billing methods.

API Overview

Service Operations

API	Description
GetService	Get all the buckets belonging to this account

Bucket Operations

API	Description
Put Bucket	Create a bucket
Put Bucket ACL	Set bucket access permissions
Put Bucket Logging	Enable the bucket logging function
Put Bucket Website	Set a bucket to the static website hosting mode
Put Bucket Referer	Set the anti-leech protection rules for a bucket
Put Bucket Lifecycle	Set the lifecycle rules for objects in a bucket
Get Bucket Acl	Acquire permissions to access a bucket

Get Bucket Location	Acquire information about the location of a bucket in the data center
Get Bucket Logging	View the access log configurations of a bucket
Get Bucket Website	View the static website hosting status of a bucket
Get Bucket Referer	View the anti-leech protection rules of a bucket
Get Bucket Lifecycle	View the lifecycle rules of objects in a bucket
Delete Bucket	Delete a bucket
Delete Bucket Logging	Disable the bucket logging function.
Delete Bucket Website	Disable the static website hosting mode of a bucket
Delete Bucket Lifecycle	Delete the lifecycle rules of objects in a bucket
Get Bucket(List Object)	Acquire information of all the objects in a bucket

Object Operations

API	Description
Put Object	Upload an object
Copy Object	Copy an object as another object
Get Object	Acquire an object
Delete Object	Delete an object
Delete Multiple Objects	Delete multiple objects
Head Object	Acquire the meta information of an object
Post Object	Use a post request to upload an object

Multipart Upload Operations

API	Description
Initiate Multipart Upload	Initialize a MultipartUpload event
Upload Part	Upload an object by parts
Upload Part Copy	Upload an object by copying parts of the file
Complete Multipart Upload	Complete a multipart upload of an entire file
Abort Multipart Upload	Abort a multipart upload event

List Multipart Uploads	List all the multipart upload events in execution
List Parts	List all the parts that are successfully uploaded and that belong to a specified upload ID

Cross-Origin Resource Sharing (CORS)

API	Description
Put Bucket cors	Set a CORS rule for a specified bucket
Get Bucket cors	Acquire the present CORS rules of a specified bucket
Delete Bucket cors	Disable the CORS function for a specified bucket and clear all the CORS rules of the bucket
Option Object	Preflight request for cross-origin access

Access Control

User signature authentication

The OSS verifies the identity of a request sender by using the AccessKeyId/AccessKeySecret symmetric encryption method. The AccessKeyId identifies a user. With the AccessKeySecret, you can encrypt the signature string and the OSS to verify the access key of the signature string. The AccessKeySecret must be kept only known to the user and the OSS. The AccessKeys can be categorized into the following types based on the account types:

- Alibaba Cloud account AccessKey: The AccessKey provided by each Alibaba Cloud account has full permissions on its resources.
- RAM account AccessKey: A RAM account is generated under the authorization of an Alibaba Cloud account, and the AccessKey of the RAM account has limited operation permissions on specified resources.
- STS temporary access credential: A temporary credential generated by an Alibaba Cloud account or a RAM account. The AccessKey of the temporary credential has limited operation permissions on specified resources for a specific period of time. The permissions are withdrawn after the period of time expires.

For details, refer to Access control.

Before sending a request to the OSS as an individual identity, you first need to generate a signature string for the request to be sent according to the format specified by the OSS and then encrypt the signature string using the AccessKeySecret to generate a verification code. After receiving the request, the OSS finds the corresponding AccessKeySecret based on the AccessKeyID, and extracts the signature string and verification code in the same way. If the calculated verification code is the same as the provided verification code, the request is deemed as valid. Otherwise, the OSS will reject the request and return an HTTP 403 error.

Add a signature to a header

You can add an authorization header to carry signature information in an HTTP request to indicate that the message has been authorized.

Calculation of the Authorization field

```
Authorization = "OSS " + AccessKeyId + ":" + Signature

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

*The ‘AccessKeySecret’ indicates the key required for the signature

- The ‘VERB’ indicates the HTTP request method, including PUT, GET, POST, HEAD, and DELETE
- The ‘\n’ is a line break.
- The ‘Content-MD5’ is the MD5 value of requested content data. The message content (excluding the header) is calculated to obtain an MD5 value, which is a 128-bit number. This number is encoded with Base64 into a Content-MD5 value. The request header can be used to check the message validity, that is, whether the message content is consistent with the sent content, such as “eB5eJF1ptWaXm4bijSPyxw==”. The request header may be empty. For details, see RFC2616 Content-MD5
- The ‘Content-Type’ indicates the requested content type, such as “application/octet-stream”. It may be empty.
- The ‘Date’ indicates the operation time. It must be in GMT format, such as “Sun, 22 Nov 2015 08:16:38 GMT”.
- The ‘CanonicalizedOSSHeaders’ indicates an assembly of HTTP headers whose prefixes

are “x-oss- “
- The ‘CanonicalizedResource’ indicates the OSS resource that the user wants to access

In specific, the values of Date and CanonicalizedResource cannot be empty. If the difference between the value of Date in the request and the time of the OSS server is greater than 15 minutes, the OSS server will reject the service and return the HTTP 403 error.

Construct CanonicalizedOSSHeaders

All the HTTP headers whose prefixes are **x-oss-** are called CanonicalizedOSSHeaders, and the method for constructing CanonicalizedResource is as follows:

1. Convert the names of all HTTP request headers whose prefixes are **x-oss-** into **lowercase** letters. For example, convert ‘X-OSS-Meta-Name: TaoBao’ into ‘x-oss-meta-name: TaoBao’ .
2. If the request is sent with the AccessKeyId and AccessKeySecret obtained by the STS, you should also add the obtained security-token value to the signature string in the form of ‘x-oss-security-token:security-token’ .
3. Sort all acquired HTTP request headers in a lexicographically ascending order.
4. Delete any space at either side of a separator between the request header and content. For example, convert ‘x-oss-meta-name: TaoBao’ into ‘x-oss-meta-name:TaoBao’ .
5. Separate all the headers and contents with the ‘\n’ separator to form the final CanonicalizedOSSHeaders.

Note:

- CanonicalizedOSSHeaders can be empty, and the ‘\n’ at the end can be dropped.
- If there is only one header, it should be like ‘x-oss-meta-a\n’ . Note the ‘\n’ at the end.
- If there are multiple headers, it should be like ‘x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n’ . Note the ‘\n’ at the end.

Construct CanonicalizedResource

The target OSS resource specified in the request sent by a user is called a CanonicalizedResource, and the method for constructing CanonicalizedResource is as follows:

1. Set CanonicalizedResource into a null character string (“”);
2. Add the OSS resource to be accessed in the following format: ‘/BucketName/ObjectName’ . (**If ObjectName does not exist**, CanonicalizedResource is “/BucketName/ “. If **BucketName does not exist either**, CanonicalizedResource is “/ “.)
3. If the requested resource includes sub-resources (SubResource), sort all the sub-resources in the **lexicographically** ascending order and separate the sub-resources using the separator

'&', to generate a sub-resource string. Add "?" and the sub-resource string to the end of the CanonicalizedResource string. In this case, CanonicalizedResource is like: /BucketName/ObjectName?acl&uploadId=UploadId

4. If the user request specifies the query string (QueryString, also called HTTP Request Parameters), sort these query strings and request values in the **lexicographically** ascending order, separate the query strings and request values using the separator '&', and add them to CanonicalizedResource based on the parameters. In this case, CanonicalizedResource is like:/BucketName/ObjectName?acl&response-content-type=ContentType&uploadId=UploadId。

Note:

- The sub-resources supported by OSS currently include: acl, uploads, location, cors, logging, website, referer, lifecycle, delete, append, tagging,objectMeta, uploadId, partNumber, security-token, position, img, style, styleName,replication, replicationProgress, replicationLocation, cname, bucketInfo, comp, qos,live, status, vod, startTime, endTime, symlink, x-oss-process, response-content-type,response-content-language, response-expires, response-cache-control, response-content-disposition, and response-content-encoding.
- There are three types of sub-resources:
  - Resource identifiers, such as acl, append, uploadId, and symlink sub-resources. For details, see Bucket-related Operations and Object-related Operations.
  - Specify response header fields such as 'response-*'. For details, see the Request Parameters section of Get Object.
  - Object handling methods, such as 'x-oss-process'. It is used as the object handling method, such as Image Processing

Rules for calculating a signature header

1. A signature string must be in the UTF-8 format. A signature string containing Chinese characters must be encoded with UTF-8 first, and then used together with 'AccessKeySecret' to calculate the final signature.
2. The signing method adopted is the HMAC-SHA1 method defined in RFC 2104, where Key is 'AccessKeySecret'.
3. Content-Type and Content-MD5 are not mandatory in a request. If the request requires signature verification, the null value can be replaced with the line break "\n".
4. Among all non-HTTP-standard headers, only the headers starting with "x-oss-" require signature strings, and other non-HTTP-standard headers will be ignored by OSS. (For example, the "x-oss-magic" header in the above example must be added with a signature string.)
5. Headers starting with "x-oss-" must comply with the following specifications before being used for signature verification:

- The header name is changed to lower-case letters.
- The headers are sorted in the lexicographically ascending order.
- There is no space before and after the colon that separates the header name and value.
- Each header is followed by the line break “\n”. If there is no header, CanonicalizedOSSHeaders is set to null.

Example signature

Assume that AccessKeyID is 44CF9590006BF252F707 and AccessKeySecret is OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV.

Request	Signature string calculation formula	Signature string
PUT /nelson HTTP/1.0 Content-MD5: eB5eJF1ptWaXm4bijSPyxw== Content-Type: text/html Date: Thu, 17 Nov 2005 18:49:58 GMT Host: oss-example.oss-cn-hangzhou.aliyuncs.com X-OSS-Meta-Author: foo@bar.com X-OSS-Magic: abracadabra	Signature = base64(hmac-sha1(AccessKeySecret, VERB + “\n” + Content-MD5 + “\n” + Content-Type + “\n” + Date + “\n” + CanonicalizedOSSHeaders + CanonicalizedResource))	“PUT\n eB5eJF1ptWaXm4bijSPyxw==\n text/html\n Thu, 17 Nov 2005 18:49:58 GMT\n x-oss-magic:abracadabra\nx-oss-meta-author:foo@bar.com\n /oss-example/nelson”

The signature calculation method is as follows:

Python sample code:

```
import base64
import hmac
import sha
h = hmac.new("OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
"PUT\nODBGOERFMDMzQTczRUY3NUE3NzA5QzdFNUYzMDQxNEM=\ntext/html\nThu, 17 Nov 2005 18:49:58
GMT\nx-oss-magic:abracadabra\nx-oss-meta-author:foo@bar.com\n/oss-example/nelson", sha)
Signature = base64.b64encode(h.digest())
print("Signature: %s" % Signature)
```

The signature calculation result is 26NBxoKdsyly4EDv6inkoDft/yA=.According to the formula Authorization = “OSS ” + AccessKeyID + “:” + Signature,the value of Authorization is OSS 44CF9590006BF252F707:26NBxoKdsyly4EDv6inkoDft/yA=.The value is added with the authorization header to form the message to be sent:

```
PUT /nelson HTTP/1.0
Authorization:OSS 44CF9590006BF252F707:26NBxoKdsyly4EDv6inkoDft/yA=
Content-Md5: eB5eJF1ptWaXm4bijSPyxw==
```

```
Content-Type: text/html
Date: Thu, 17 Nov 2005 18:49:58 GMT
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
X-OSS-Meta-Author: foo@bar.com
X-OSS-Magic: abracadabra
```

Detail analysis

1. If the input AccessKeyID does not exist or is inactive, the error 403 Forbidden will be returned. Error code: InvalidAccessKeyId.
2. If the authorization value format in the user request header is incorrect, the error 400 Bad Request will be returned. Error code: InvalidArgument.
3. All the requests of the OSS must use the GMT time format stipulated by the HTTP 1.1 protocol. In specific, the date format is:date1 = 2DIGIT SP month SP 4DIGIT; day month year (for example, 02 Jun 1982)In the foregoing date format, “day” occupies “2 digits”. Therefore, “Jun 2”, “2 Jun 1982”, and “2-Jun-82” are all invalid date formats.
4. If Date is not input into the header or the format is incorrect during signature verification, the error 403 Forbidden will be returned. Error code: AccessDenied.
5. The request must be input within 15 minutes based on the current time of the OSS server; otherwise, the error 403 Forbidden will be returned. Error code: RequestTimeTooSkewed.
6. If the AccessKeyID is active but OSS determines that the signature of the user request is incorrect, the error 403 Forbidden will be returned, and the correct signature string for verification and encryption is returned to the user in the response message. The user can check whether the signature string is correct based on the response of the OSS.Return example:

```
<?xml version="1.0" ?>
<Error>
<Code>
SignatureDoesNotMatch
</Code>
<Message>
The request signature we calculated does not match the signature you provided. Check your key and signing method.
</Message>
<StringToSignBytes>
47 45 54 0a 0a 0a 57 65 64 2c 20 31 31 20 4d 61 79 20 32 30 31 31 20 30 37 3a 35 39 3a 32 35 20 47 4d 54 0a 2f 75 73 72 65 61 6c 74 65 73 74 3f 61 63 6c
</StringToSignBytes>
<RequestId>
1E446260FF9B10C2
</RequestId>
<HostId>
oss-cn-hangzhou.aliyuncs.com
</HostId>
<SignatureProvided>
y5H7yzPsA/tP4+0tH1HHvPEwUv8=
</SignatureProvided>
```

```
<StringToSign>
GET
Wed, 11 May 2011 07:59:25 GMT
/oss-example?acl
</StringToSign>
<OSSAccessKeyId>
AKIAIVAKMSMOY7VOMRWQ
</OSSAccessKeyId>
</Error>
```

Note:

- OSS SDK has implemented the signature. You don’ t need to worry about the signature issue during usage of the OSS SDK. If you want to learn more about the signature implementations of specific languages, refer to the OSS SDK code. The files for implementing OSS SDK signature are shown in the table below:

SDK	Signature implementations
Java SDK	OSSRequestSigner.java
Python SDK	auth.py
.Net SDK	OssRequestSigner.cs
PHP SDK	OssClient.php
C SDK	oss_auth.c
JavaScript SDK	client.js
Go SDK	auth.go
Ruby SDK	util.rb
iOS SDK	OSSModel.m
Android SDK	OSSUtils.java

- When you implement the signature on your own, and the access to OSS receives an error of SignatureDoesNotMatch, you can use Visualized Signature Tool to confirm the signature and eliminate the error.

FAQs

Content-MD5 calculation method

Content-MD5 calculation
The message content "123456789" is used as an example. The Content-MD5 value of the string

```
is calculated as follows:
The algorithm defined in related standards can be simplified to the following:
1. Calculate the MD5-encrypted 128-bit binary array.
2. Encode the binary array (instead of the 32-bit string code) with Base64.

Python is used as an example.
The correct calculation code is:
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789")
>>> base64.b64encode(hash.digest())
'eB5eJF1ptWaXm4bijSPyxw=='

Note:
The correct code is: hash.digest(), used to calculate a 128-bit binary array
>>> hash.digest()
'x\x1e^$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'

A common error is encoding the calculated 32-bit string code with Base64.
An incorrect example: hash.hexdigest(), and a visible 32-bit string will be calculated.
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
Result of encoding the incorrect MD5 value with Base64:
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

Add a signature to URL

In addition to using an authorization header, you can also add signature information to a URL so that you can forward the URL to a third party for authorized access.

Implementation

URL signature example:

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-
api.pdf?AccessKeyId=AccessKeyId&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv4%3D
```

The URL signature must include at least the following three parameters: Signature, Expires, and AccessKeyId.

- The Expires parameter indicates the timeout time of the URL. The value of this parameter is UNIX time (which is the number of seconds that have elapsed since 00:00:00 UTC, January 1, 1970. For details, see Wikipedia). If the time when the OSS receives the URL request is later than the value of the Expires parameter included in the signature, an error code indicating

that the request has timed out will be returned. For example, if the current time is 1141889060, to create a URL that will expire in 60 seconds, you can set the value of Expires to 1141889120.
- AccessKeyId refers to the AccessKeyId in the key.
- Signature indicates the signature information. For all requests and header parameters that OSS supports, the algorithm for adding a signature to a URL is basically the same as that for adding a signature to a header.

```
Signature = urlencode(base64(hmac-sha1(AccessKeySecret,
VERB + "\n"
+ CONTENT-MD5 + "\n"
+ CONTENT-TYPE + "\n"
+ EXPIRES + "\n"
+ CanonicalizedOSSHeaders
+ CanonicalizedResource)))
```

Specifically, the differences mainly lie in the following:

1. When a signature is added to a URL, the Expires parameter replaces the Date parameter.
2. Signatures cannot be included in the URL and the Header at the same time.
3. If there are more than one incoming Signature, Expires, or AccessKeyId value, the first of each incoming value is used.
4. Whether the request time is later than the Expires time is verified first before the signature is verified.
5. When you put the signature string into the URL, remember to perform the UrlEncode for the URL.

- When you add a signature to a temporary user URL, the ‘security-token’ should be carried. The format is as follows:

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-
api.pdf?AccessKeyId=AccessKeyId&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv4
%3D&security-token=SecurityToken
```

Sample code

Python sample code used to add a signature to a URL:

```
import base64
import hmac
import sha
import urllib
h = hmac.new("OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
```

```
"GET\n\n\n1141889120\n/oss-example/oss-api.pdf",
sha)
urllib.quote (base64.encodestring(h.digest()).strip())
```

Note:

- The above is the Python sample code
- OSS SDK provides the method for adding a signature into the URL. For usage, see the 'Authorize Access' section in the SDK file.
- For the implementation of adding a signature to the OSS SDK URL, see the table below.

SDK	URL signature method	Implementation file
Java SDK	OSSClient.generatePresignedUrl	OSSClient.java
Python SDK	Bucket.sign_url	api.py
.Net SDK	OssClient.GeneratePresignedUri	OssClient.cs
PHP SDK	OssClient.signUrl	OssClient.php
JavaScript SDK	signatureUrl	object.js
C SDK	oss_gen_signed_url	oss_object.c

Detail analysis

1. If you adopt the approach of adding a signature to a URL, the authorized data will be exposed on the internet before the authorization period expires. Please assess the usage risks in advance.
2. The PUT and GET requests both support adding a signature in a URL.
3. When a signature is added to a URL, the sequence of Signature, Expires, and AccessKeyId can be swapped. If one or more Signature, Expires, or AccessKeyId parameter are missing, the error 403 Forbidden will be returned. Error code: AccessDenied.
4. If the current access time is later than the Expires time set in the request, the error 403 Forbidden will be returned. Error code: AccessDenied.
5. If the format of the Expires time is incorrect, the error 403 Forbidden will be returned. Error code: AccessDenied.
6. If the URL includes one or more Signature, Expires, or AccessKeyId parameter and the header also includes signature information, the error 400 Bad Request will be returned. Error code: InvalidArgument.
7. When the signature string is generated, the Date parameter is replaced by the Expires parameter, but the headers such as content-type and content-md5 defined in the foregoing section are still included. (Though the Date request header still exists in the request, it does not need to be added to the signature string.)

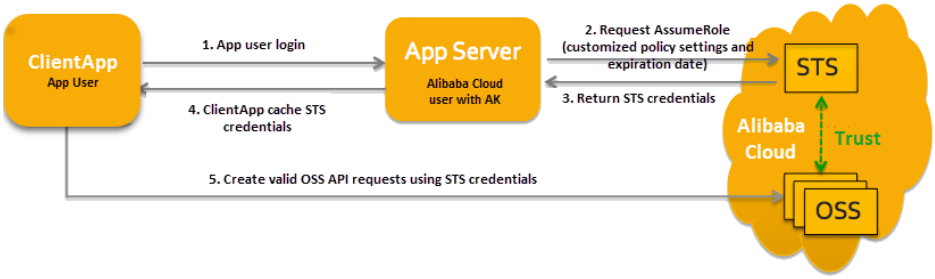
Temporary access credential

Introduction to STS

OSS can temporarily perform authorized access through the Alibaba Cloud STS (Security Token Service). Alibaba Cloud STS is a web service that provides a temporary access token to a cloud computing user. Using STS, you can grant access credentials to a third-party application or federated user (you can manage the user IDs) with customized permissions and validity periods. Third-party applications or federated users can use these access credentials to directly call the Alibaba Cloud product APIs or use the SDKs provided by Alibaba Cloud products to access the cloud product APIs.

- You do not need to expose your long-term key (AccessKey) to a third-party application and only need to generate an access token and send the access token to the third-party application. You can customize the access permission and validity of this token.
- You do not need to care about permission revocation issues. The access credential automatically becomes invalid when it expires.

Using an app as an example, the interaction process is shown below:



The solution is described in detail as follows:

1. Log on as the app user. App user IDs are managed by the customer. Customers can customize the ID management system, or use an external web account or OpenID. For each valid app user, the AppServer can precisely define the minimum access permission.
2. The AppServer requests a security token (SecurityToken) from the STS. Before calling STS, the AppServer needs to determine the minimum access permission (described in policy syntax) of the app user and the expiration time of the authorization.Then the security token is obtained by calling the STS' AssumeRole interface. For details about role management and usage, refer to Role Management in the RAM User Guide.
3. The STS returns a valid access credential to the AppServer, including a security token, a temporary access key (AccessKeyID and AccessKeySecret), and the expiry time.
4. The AppServer returns the access credential to the ClientApp. The ClientApp can cache this credential. When the credential becomes invalid, the ClientApp needs to request a new valid access credential from the AppServer. For example, if the access credential is valid for

one hour, the ClientApp can request the AppServer to update the access credential every 30 minutes.
5. The ClientApp uses the access credential cached locally to request Alibaba Cloud Service APIs. The cloud services will perceive the STS access credential, and rely on STS to verify the credential and correctly respond to the user request.

For details about the STS security token, refer to Role Management in the RAM User Guide. The key is to call AssumeRole of the STS interface to obtain valid access credentials. You can also directly use STS SDK to call the this method. Click here for details.

Use STS credentials to construct signed requests

After obtaining the STS temporary credential, the client of the user creates a signature using the security token (SecurityToken) and temporary access key (AccessKeyId and AccessKeySecret) in the credential. The method for constructing an authorized access signature is basically the same as using the AccessKey of a root account to add a signature to a header. Pay attention to the following two points:

- The signature key used by the user is the temporary access key (AccessKeyId and AccessKeySecret) provided by The STS.
- The user needs to carry the security token (security token) in the request header or in the URI As a request parameter. These two manners are alternative. If both manners are selected, the OSS will return an InvalidArgument error.
    - The header x-oss-security-token: SecurityToken is carried in a request header. When CanonicalizedOSSHeaders of the signature is calculated, x-oss-security-token is taken into consideration.
    - Parameter security-token=SecurityToken is carried in the URL. When CanonicalizedResource of the signature is calculated, security-token is taken into consideration and considered as a sub-resource.

Bucket permission control

OSS provides an Access Control List (ACL) for bucket-level access control. Currently, three access permissions are available for a bucket: public-read-write, public-read, and private.

- public-read-write: Anyone (including anonymous users) can perform Put, Get, and Delete operations on the objects in the bucket. The expenses incurred by these operations shall be borne by the creator of the bucket. Please use this permission with caution.
- public-read: Only the creator of a bucket can perform write operations (including Put Object and Delete Object) on objects in the bucket. Other users (including anonymous users) can perform read operations (Get Object) on objects in the bucket.
- private: Only the creator of a bucket can perform read and write operations (including Put

Object, Delete Object, and Get Object) on objects in the bucket. Other users cannot access objects in the bucket.

When a user creates a new bucket without designating the bucket permission, the OSS will automatically set the permission to private. For an existing bucket, only the creator of the bucket can change its permissions by using the Put Bucket Acl interface provided by the OSS.

Definitions of Common HTTP Headers

Common Request Headers

Some common request headers are used in the OSS RESTful interfaces. These request headers can be used by all the OSS requests. The following table lists the specific definitions of the request headers:

Name	Description
Authorization	Indicates the verification information used to verify the validity of a request. Type: string Default value: none Application scenario: non-anonymous requests
Content-Length	Indicates the HTTP request content length defined in RFC2616. Type: string Default value: none Application scenario: requests in which data needs to be submitted to the OSS
Content-Type	Indicates the HTTP request content type defined in RFC2616. Type: string Default value: none Application scenario: requests in which data needs to be submitted to the OSS
Date	Indicates the GMT time stipulated in the HTTP 1.1 protocol, for example, Wed, 05 Sep. 2012 23:00:00 GMT Type: string Default value: none
Host	Indicates the host to be accessed. The format of the value is: <bucketname>.oss-cn-hangzhou.aliyuncs.com. Type: string Default value: none

Common Response Headers

Some common response headers are used in the OSS RESTful interfaces. These response headers can be used by all the OSS requests. The following table lists the specific definitions of the response headers:

Name	Description
Content-Length	Indicates the HTTP request content length defined in RFC2616. Type: string Default value: none Application scenario: requests in which data needs to be submitted to the OSS
Connection	Indicates the connection status between the client and the OSS server. Type: Enumeration Valid value: open or close Default value: none
Date	Indicates the GMT time stipulated in the HTTP 1.1 protocol, for example, Wed, 05 Sep. 2012 23:00:00 GMT Type: string Default value: none
ETag	The ETag (entity tag) is created when an object is generated and is used to indicate the content of the object. For an object created by using a Put Object request, the value of ETag is the value of MD5 in the content of the object. For an object created in other manners, the value of ETag is the UUID in the content of the object. The value of ETag can be used to check whether the content of the object is changed. Type: string Default value: none
Server	Indicates the server that generates the response. Type: string Default value: AliyunOSS
x-oss-request-id	Indicates the UUID of the response and is created by AliCloud OSS. If you encounters a problem when using the OSS service, you can contact OSS support personnel by using this field, to rapidly locate the problem. Type: string Default value: none

Service Operations

GetService (ListBucket)

Sending a Get request to the server can return all buckets owned by the requester, and “/ “ represents the root directory.

Request syntax

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

Request parameters

When using GetService(ListBucket), you can prescribe a limit to the list with prefix, marker and max-uploads to return partial results.

Name	Description
prefix	Limit that the returned bucket name must be prefixed accordingly. You can also choose not to set the prefix, which then does not filter the prefix information Data type: string Default value: none
marker	to set the returned results to begin from the first entry after the marker in alphabetical order. You can also choose not to set the marker, which then returns results from the beginning Data type: string Default value: none
max-uploads	Limit the maximum number of buckets returned for one request. If not set, the default value is 100. The max-uploads value cannot exceed 1000 Data type: string Default value: 100

Response elements

Name	Description
ListAllMyBucketsResult	The container that saves results of the Get Service request. Type: container Sub-nodes: Owner, Buckets Parent node: none
Prefix	Prefix of the result of this query. This node is available only when not all buckets are returned Type: string Parent node: ListAllMyBucketsResult
Marker	Mark the origin of this GetService(ListBucket) request. This node is available only when not all buckets are returned Type: string Parent node: ListAllMyBucketsResult
MaxKeys	The maximum number of returned results in response to a request. This node is available only when not all buckets are returned Type: string Parent node: ListAllMyBucketsResult
IsTruncated	Whether all results have been returned. “true” means that not all results are returned this time; “false” means that all results are returned this time. This node is available only when not all buckets are returned. Type: enumerative string Valid values: true, false Parent node: ListAllMyBucketsResult
NextMarker	Indicate that this can be taken as the marker for the next GetService(ListBucket) request to return unreturned results. This node is available only when not all buckets are returned. Type: string Parent node: ListAllMyBucketsResult
Owner	The container that saves the information about the bucket owner. Type: container Parent node: ListAllMyBucketsResult
ID	User ID of the bucket owner. Type: string Parent node: ListAllMyBucketsResult.Owner
DisplayName	Name of the bucket owner (the same as the ID currently). Type: string Parent node: ListAllMyBucketsResult.Owner

Buckets	The container that saves the information about multiple buckets. Type: container Sub-node: Bucket Parent node: ListAllMyBucketsResult
Bucket	The container that saves the bucket information. Type: container Sub-nodes: Name, CreationDate, Location Parent node: ListAllMyBucketsResult.Buckets
Name	Bucket name. Type: string Parent node: ListAllMyBucketsResult.Buckets.Bucket
CreateDate	The create time of the bucket Type: time (format: yyyy-mm-ddThh:mm:ss.timezone, for example, 2011-12-01T12:27:13.000Z) Parent node: ListAllMyBucketsResult.Buckets.Bucket
Location	The region of the data center that the bucket is located in Type: string Parent node: BucketInfo.Bucket
Location	The data center that the bucket is located in Type: string Parent node: ListAllMyBucketsResult.Buckets.Bucket
ExtranetEndpoint	The internet domain name that the bucket accesses Type: string Parent node: ListAllMyBucketsResult.Buckets.Bucket
IntranetEndpoint	The intranet domain name for accessing the bucket from ECS in the same region Type: string Parent node: ListAllMyBucketsResult.Buckets.Bucket

Detail analysis

1. The API of GetService is valid only for those users who have been authenticated.
2. If no information for user authentication is provided in a request (namely an anonymous access), 403 Forbidden is returned. Error code: AccessDenied.
3. When all buckets are returned, the returned xml does not contain the nodes Prefix, Marker, MaxKeys, IsTruncated and NextMarker. If some results are not returned yet, the above nodes are added, in which NextMarker is used to assign the marker for successive query.

Example

Request example I

```
GET / HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: oss-cn-hangzhou.aliyuncs.com
Authorization: OSS nxj7dtl1c24jwhcyl5hpvnhi:COS3OQkfQPnKmYZTEHYv2qUl5jI=
```

Return example I

```
HTTP/1.1 200 OK
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 556
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D74

<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
<Owner>
<ID>51264</ID>
<DisplayName>51264</DisplayName>
</Owner>
<Buckets>
<Bucket>
<CreationDate>2015-12-17T18:12:43.000Z</CreationDate>
<ExtranetEndpoint>oss-cn-shanghai.aliyuncs.com</ExtranetEndpoint>
<IntranetEndpoint>oss-cn-shanghai-internal.aliyuncs.com</IntranetEndpoint>
<Location>oss-cn-shanghai</Location>
<Name>app-base-oss</Name>
</Bucket>
<Bucket>
<CreationDate>2014-12-25T11:21:04.000Z</CreationDate>
<ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
<IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</IntranetEndpoint>
<Location>oss-cn-hangzhou</Location>
<Name>atestleo23</Name>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

Request example II

```
GET /?prefix=xz02tphky6fjfiuc&max-keys=1 HTTP/1.1
Date: Thu, 15 May 2014 11:18:32 GMT
Host: oss-cn-hangzhou.aliyuncs.com
Authorization: OSS nxj7dtl1c24jwhcyl5hpvnhi:COS3OQkfQPnKmYZTEHYv2qUl5jI=
```

Return example II

```
HTTP/1.1 200 OK
Date: Thu, 15 May 2014 11:18:32 GMT
Content-Type: application/xml
Content-Length: 545
Connection: keep-alive
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D75

<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
<Prefix>xz02tphky6fjfiuc</Prefix>
<Marker></Marker>
<MaxKeys>1</MaxKeys>
<IsTruncated>true</IsTruncated>
<NextMarker>xz02tphky6fjfiuc0</NextMarker>
<Owner>
<ID>ut_test_put_bucket</ID>
<DisplayName>ut_test_put_bucket</DisplayName>
</Owner>
<Buckets>
<Bucket>
<CreationDate>2014-05-15T11:18:32.000Z</CreationDate>
<ExtranetEndpoint>oss-cn-hangzhou.aliyuncs.com</ExtranetEndpoint>
<IntranetEndpoint>oss-cn-hangzhou-internal.aliyuncs.com</IntranetEndpoint>
<Location>oss-cn-hangzhou</Location>
<Name>xz02tphky6fjfiuc0</Name>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

Bucket Operations

Put Bucket

Put Bucket is used to create buckets (anonymous access is not supported). The region of the created bucket is consistent with the region of the endpoint that sends the request. Once the data center is determined, all objects in this bucket will be stored in the corresponding region. For details, see Regions and endpoints.

Request syntax

```
PUT / HTTP/1.1
```

```
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
x-oss-acl: Permission
Authorization: SignatureValue
```

Detail analysis

1. You can use the “x-oss-acl” header in a Put request to set access permissions for a bucket. Currently, three bucket access permissions are available: public-read-write, public-read, and private.
2. If the requested bucket already exists and is owned by the requester, 200 OK is returned for success.
3. If the requested bucket already exists but is not owned by the requester, 409 Conflict is returned. Error code: BucketAlreadyExists.
4. If the bucket to be created does not conform to the naming conventions, the message of 400 Bad Request is returned. Error code: InvalidBucketName.
5. If the information for user authentication is not introduced when you initiate a Put Bucket request, the message of 403 Forbidden is returned. Error code: AccessDenied.
6. If the maximum number of bucket creation (**10 by default**) is exceeded when you initiate a PutBucket request, the message of 400 Bad Request is returned. Error code: TooManyBuckets.
7. If no access permission is specified for the created bucket, the “Private” permission is used by default.

Example

Request example:

```
PUT / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 03:15:40 GMT
x-oss-acl: private
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:77Dvh5wQgIjWjwO/KyRt8dOPfo8=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 03:15:40 GMT
Location: /oss-example
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Put Bucket ACL

The Put Bucket ACL interface is used to modify the access permissions for a bucket. Currently, three bucket access permissions are available: public-read-write, public-read, and private. You can use the “x-oss-acl” header in a Put request to set the Put Bucket ACL operation. Only the creator of the bucket has permission to perform this operation. If the operation succeeds, 200 will be returned; otherwise, the corresponding error code and prompt message will be returned.

Request syntax

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

Detail analysis

1. When a bucket already exists and is owned by the request sender and the permission in the request is different from the existing permission, this request does not change bucket content but updates permission.
2. If the information for user authentication is not introduced when you initiate a Put Bucket request, the message of 403 Forbidden will be returned. Error code: AccessDenied.
3. If the “x-oss-acl” header is unavailable in a request and the bucket already exists and belongs to the request sender, the permissions for the original bucket remain the same.

Example

Request example:

```
PUT /?acl HTTP/1.1
x-oss-acl: public-read
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 03:21:12 GMT
Content-Length: 0
```

```
Connection: keep-alive
Server: AliyunOSS
```

If the permission for this setting does not exist, the message of 400 Bad Request is shown:

Returned error example:

```
HTTP/1.1 400 Bad Request
x-oss-request-id: 56594298207FB304438516F9
Date: Fri, 24 Feb 2012 03:55:00 GMT
Content-Length: 309
Content-Type: text/xml; charset=UTF-8
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>InvalidArgument</Code>
<Message>no such bucket access control exists</Message>
<RequestId>56594298207FB304438516F9</RequestId>
<HostId>leo.oss-test.aliyun-inc.com</HostId>
<ArgumentName>x-oss-acl</ArgumentName>
<ArgumentValue>error-acl</ArgumentValue>
</Error>
```

Put Bucket Logging

> The OSS provides bucket access logs for bucket owners to understand and analyze bucket access behaviors in a convenient way. The bucket access logs provided by the OSS do not guarantee that every single access record is logged.

A bucket owner can enable the access logging function for his/her bucket. When this function is enabled, the OSS automatically records detailed information about the requests to this bucket, and follows the user-specified rules to write the access logs as an object into the user-specified bucket hourly.The OSS provides bucket access logs for bucket owners to understand and analyze bucket access behaviors in a convenient way. The bucket access logs provided by the OSS do not guarantee that every single access record is logged.

Request syntax

```
PUT /?logging HTTP/1.1
Date: GMT Date
Content-Length：ContentLength
Content-Type: application/xml
```

```
Authorization: SignatureValue
Host: BucketName.oss-cn-hangzhou.aliyuncs.com

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus>
<LoggingEnabled>
<TargetBucket>TargetBucket</TargetBucket>
<TargetPrefix>TargetPrefix</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

Request elements

Name	Description	Required?
BucketLoggingStatus	The container for storing access log status information Type: container Child element: LoggingEnabled Parent element: none	Yes
LoggingEnabled	The container for storing access log information. This element is required only when server access logging is enabled. Type: container Child element: TargetBucket, TargetPrefix Parent element: BucketLoggingStatus	No
TargetBucket	The bucket for storing access logs. Type: character Child element: none Parent element: BucketLoggingStatus.LoggingEnabled	This element is required when server access logging is enabled
TargetPrefix	The prefix of the names of saved access log files. Type: character Child element: none Parent element: BucketLoggingStatus.LoggingEnabled	No

Naming rules for the objects storing access logs

```
<TargetPrefix><SourceBucket>-YYYY-mm-DD-HH-MM-SS-UniqueString
```

In the naming rules, the TargetPrefix is specified by the user; YYYY, mm, DD, HH, MM and SS give the year, month, day, hour, minutes and seconds of the creation time in Arabic numerals (note the digits); and UniqueString is the string generated by the OSS system. An example for the name of an object actually used to store OSS access logs is given below:

```
MyLog-oss-example-2012-09-10-04-00-00-0000
```

In the above example, "MyLog-" is the Object prefix specified by the user; "oss-example" is the name of the origin bucket; "2012-09-10-04-00-00" is the Object creation time (Beijing time); and "0000" is the string generated by the OSS system.

Log file format

Name	Example	Description
Remote IP	119.140.142.11	IP address from which the request is initiated (the proxy or user firewall may block this field)
Reserved	-	Reserved field
Reserved	-	Reserved field
Time	[02/May/2012:00:00:04 +0800]	Time when the OSS receives the request
Request-URI	"GET /aliyun-logo.png HTTP/1.1"	User-Requested URI (including query-string)
HTTP Status	200	HTTP status code returned by the OSS
SentBytes	5576	Traffic that the user downloads from the OSS
RequestTime (ms)	71	Time spent in completing this request (in ms)
Referer	http://www.alicloud.com/product/oss	HTTP Referer in the request
User-Agent	curl/7.15.5	HTTP User-Agent header
HostName	oss-example.oss-cn-hangzhou.aliyuncs.com	Domain name for access request
Request ID	505B01695037C2AF032593A4	UUID used to uniquely identify this request
LoggingFlag	true	Whether the access logging function is enabled
Requester AliCloud ID	1657136103983691	Alibaba Cloud ID of the requester, "-" for anonymous access

Operation	GetObject	Request type
Bucket	oss-example	Name of the bucket requested for access
Key	/aliyun-logo.png	Key of user request
ObjectSize	5576	Object size
Server Cost Time (ms)	17	Time taken by the OSS server to process this request (in ms)
Error Code	NoSuchBucket	Error code returned by the OSS
Request Length	302	Length of user request (byte)
UserID	1657136103983691	ID of the bucket owner
Delta DataSize	280	Bucket size variation, "-" for no change
Sync Request	-	Whether this is a back-to-source request from CND, "-" for no
Reserved	-	Reserved field

Detail analysis

- The source bucket and target bucket must belong to the same user.
- In the request syntax shown above, "BucketName" refers to the bucket for which access logging is enabled; "TargetBucket" refers to the bucket into which access logs are saved; "TargetPrefix" refers to the name prefix of the object storing access logs and can be null.
- The source bucket and target bucket can be the same or different buckets. You can save logs from multiple source buckets to the same target bucket (in this case, it is recommended that you assign different values to TargetPrefix).
- To disable the access logging function for a bucket, you just need to send an empty BucketLoggingStatus request. For the detailed method, refer to the following request example.
- All PUT Bucket Logging requests must be provided with signatures, meaning that anonymous access is not supported.
- If the initiator of a PUT Bucket Logging request is not the owner of the source bucket (BucketName in the request example), the OSS will return error code 403.
- If the source bucket does not exist, OSS will return the error code: NoSuchBucket.
- If the initiator of a PUT Bucket Logging request is not the owner of the target bucket (indicated by TargetBucket in the request example), OSS will return Error 403. If the target bucket does not exist, OSS will return the error code: InvalidTargetBucketForLogging.
- The source bucket and target bucket must belong to the same data center. Otherwise, Error 400 with the error code: InvalidTargetBucketForLogging will be returned.

- If a PUT Bucket Logging request has invalid XML, the error code: MalformedXML will be returned.
- The source bucket and target bucket can be the same bucket. You can save the logs of different source buckets into the same target bucket (note that you need to set TargetPrefix to different values).
- When the source bucket is deleted, the corresponding logging rules are also deleted.
- The OSS generates a bucket access log file every hour. However, all requests in the hour may not be recorded in the log file, but may be recorded in the previous or next log file.
- In the naming rules for log files generated by the OSS, “UniqueString” is just a UUID that the OSS generates for a file to uniquely identify the file.
- Each time the OSS generates a bucket access log file, this is considered a PUT operation and the occupied space is recorded, but the generated traffic is not recorded. After log files are generated, you can operate these log files as common objects.
- The OSS ignores all query-string parameters prefixed by “x-” but such query-string parameters are recorded in access logs. If you want to mark a special request from massive access logs, you can add a query-string parameter prefixed by “x-” to the URL. For example:
  http://oss-example.oss-cn-hangzhou.aliyuncs.com/aliyun-logo.png
  http://oss-example.oss-cn-hangzhou.aliyuncs.com/aliyun-logo.png?x-user=admin
- When the OSS processes the above two requests, the results are the same. However, you can search access logs with “x-user=admin” to quickly locate the marked request.
- You may see “-” in any field of OSS logs. It indicates that data is unknown or the field is invalid for the current request.
- Certain fields will be added to the end of OSS log files in the future based on the requirements. It is recommended that developers take compatibility issues into consideration when developing log processing tools.
- If you have uploaded the Content-MD5 request header, the OSS will calculate the body’s Content-MD5 and check if the two are the same. If the two are different, the error code: InvalidDigest will be returned.

Example

Example of a request for enabling bucket access logging:

```
PUT /?logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus>
<LoggingEnabled>
<TargetBucket>doc-log</TargetBucket>
<TargetPrefix>MyLog-</TargetPrefix>
```

```
</LoggingEnabled>
</BucketLoggingStatus>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Example of a request for disabling bucket access logging:

```
PUT /?logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Type: application/xml
Content-Length: 86
Date: Fri, 04 May 2012 04:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

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

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 04:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Put Bucket Website

With the Put Bucket Website operation, you can set a bucket to the static website hosting mode.

Request syntax

```
PUT /?website HTTP/1.1
Date: GMT Date
Content-Length：ContentLength
Content-Type: application/xml
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
```

```
Authorization: SignatureValue

<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
<IndexDocument>
<Suffix>index.html</Suffix>
</IndexDocument>
<ErrorDocument>
<Key>errorDocument.html</Key>
</ErrorDocument>
</WebsiteConfiguration>
```

Request elements

Name	Description	Required?
ErrorDocument	The parent element of the child element key Type: container Parent element: WebsiteConfiguration	No
IndexDocument	The parent element of the child element suffix Type: container Parent element: WebsiteConfiguration	Yes
Key	The file name used to return Error 404 Type: string Parent element: WebsiteConfiguration.ErrorDocument This element is required only when ErrorDocument is set	Conditional
Suffix	The index file name added when a directory URL is returned. This element cannot be empty or contain a slash (/). For example, if the index file index.html is configured, oss-cn-hangzhou.aliyuncs.com/mybucket/mydir/ contained in an access request is converted into oss-cn-hangzhou.aliyuncs.com/mybucket/index.html by default. Type: string Parent element: WebsiteConfiguration.IndexDocument	Yes
WebsiteConfiguration	Requested container Type: container Parent element: none	Yes

Detail analysis

1. Static websites are websites where all web pages are composed of static content, including scripts such as JavaScript executed on the client. The OSS does not support content that needs to be processed by the server, such as PHP, JSP, and APS.NET.
2. If you want to use your own domain name to access bucket-based static websites, the CNAME domain name applies. For details about the configuration method, refer to Section 3.4 Custom Domain Name Binding.
3. When you set a bucket to the static website hosting mode, you must specify the index page and the error page is optional.
4. When you set a bucket to the static website hosting mode, the specified index page and error page are an object in the bucket.
5. After a bucket is set to static website hosting mode, the OSS will return the index page for anonymous access to the root domain name of the static website, and return Get Bucket results for signed access to the root domain name of the static website.
6. If you have uploaded the Content-MD5 request header, the OSS will calculate the body’ s Content-MD5 and check if the two are the same. If the two are different, the error code: InvalidDigest will be returned.

Example

Request example:

```
PUT /?website HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 209
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration>
<IndexDocument>
<Suffix>index.html</Suffix>
</IndexDocument>
<ErrorDocument>
<Key>error.html</Key>
</ErrorDocument>
</WebsiteConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
```

```
Server: AliyunOSS
```

Put Bucket Referer

With the Put Bucket Referer operation, you can set the referer access white list of a bucket and whether the access request with the referer field being null is allowed. For details about bucket referer, see OSS Anti-Leech.

Request syntax

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

<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
<RefererList>
<Referer> http://www.aliyun.com</Referer>
<Referer> https://www.aliyun.com</Referer>
<Referer> http://www.*.com</Referer>
<Referer> https://www.?.aliyuncs.com</Referer>
</RefererList>
</RefererConfiguration>
```

Request elements

Name	Description	Required?
RefererConfiguration	The container that saves the Referer configuration content Type: container Sub-nodes: AllowEmptyReferer node and RefererList node Parent node: none	Yes
AllowEmptyReferer	Specify whether the access request with the referer field being null is allowed. Type: enumerative string Valid value: true or false Default value: true Parent node:	Yes

	RefererConfiguration	
RefererList	The container that saves the referer access whitelist. Type: container Parent node: RefererConfiguration Sub-node: Referer	Yes
RefererList	Specify a referer access whitelist. Type: string Parent node: RefererList	Optional

Detail analysis

1. Only the bucket owner can initiate a Put Bucket Referer request. Otherwise, the message of 403 Forbidden will be returned. Error code: AccessDenied.
2. The configuration specified in AllowEmptyReferer replaces the previous AllowEmptyReferer configuration. This field is mandatory. By default, AllowEmptyReferer in the system is configured as true.
3. This operation overwrites the previously configured whitelist with the whitelist in the RefererList. When the user-uploaded RefererList is empty (containing no referer request element), this operation overwrites the configured whitelist, that is, the previously configured RefererList is deleted.
4. If you have uploaded the Content-MD5 request header, the OSS will calculate the body’ s Content-MD5 and check if the two are the same. If the two are different, the error code: InvalidDigest will be returned.

Example

Request example:

Example of a request with no referer contained:

```
PUT /?referer HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList />
</RefererConfiguration>
```

Example of a request with referer contained:

```
PUT /?referer HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 247
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
< RefererList>
<Referer> http://www.aliyun.com</Referer>
<Referer> https://www.aliyun.com</Referer>
<Referer> http://www.*.com</Referer>
<Referer> https://www.?.aliyuncs.com</Referer>
</ RefererList>
</RefererConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Put Bucket Lifecycle

The bucket owner can set the lifecycle of a bucket with the Put Bucket Lifecycle request. After Lifecycle is enabled, the OSS automatically deletes the objects matching the lifecycle rules on a regular basis.

Request syntax

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

<?xml version="1.0" encoding="UTF-8"?>
<LifecycleConfiguration>
<Rule>
<ID>RuleID</ID>
<Prefix>Prefix</Prefix>
```

```
<Status>Status</Status>
<Expiration>
<Days>Days</Days>
</Expiration>
<AbortMultipartUpload>
<Days>Days</Days>
</AbortMultipartUpload>
</Rule>
</LifecycleConfiguration>
```

Request elements

Name	Description	Required?
CreatedBeforeDate	Specify the time before which the rules will go into effect. The date must conform to the ISO8601 format and always be UTC 00:00 am. For example: 2002-10-11T00:00:00.000Z Type: string Parent node: Expiration or AbortMultipartUpload	One from the two: Days and CreatedBeforeDate
Days	Specify how many days after the last object modification until the rules take effect. Type: positive integer Parent node: Expiration	One from the two: Days and CreatedBeforeDate
Expiration	Specify the expiration attribute of the object. Type: container Sub-node: Days or CreatedBeforeDate Parent node: Rule	No
AbortMultipartUpload	Specify the expiration attribute of the unfulfilled Part rules. Type: container Sub-node: Days or CreatedBeforeDate Parent node: Rule	No
ID	The unique ID of a rule. An ID is composed of 255 bytes at most. When you fail to specify this value or this value is null, the OSS generates a unique value for you. Type: string Sub-node: none Parent node: Rule	No
LifecycleConfiguration	Container used for storing	Yes

	lifecycle configurations, which can hold a maximum of 1,000 rules. Type: container Sub-node: Rule Parent node: none	
Prefix	Specify the prefix applicable to a rule. Only those objects with the matching prefix can be affected by the rule. It cannot be overlapped. Type: string Sub-node: none Parent node: Rule	Yes
Rule	Express a rule Type: container Sub-nodes: ID, Prefix, Status, Expiration Parent node: LifecycleConfiguration	Yes
Status	If this value is Enabled, the OSS executes this rule regularly. If this value is Disabled, the OSS ignores this rule. Type: string Parent node: Rule Valid value: Enabled，Disabled	Yes

Detail analysis

1. Only the bucket owner can initiate a Put Bucket Lifecycle request. Otherwise, the message of 403 Forbidden will be returned. Error code: AccessDenied.
2. If no lifecycle has been set previously, this operation creates a new lifecycle configuration or overwrites the previous configuration.
3. You can also set an expiration time for the object, or for the Part. Here the Part refers to the unsubmitted parts for multipart upload.

Example

Request example:

```
PUT /?lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Content-Length: 443
Date: Mon, 14 Apr 2014 01:08:38 GMT
```

```
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

<?xml version="1.0" encoding="UTF-8"?>
</LifecycleConfiguration>
<Rule>
<ID>delete objects and parts after one day</ID>
<Prefix>logs/</Prefix>
<Status>Enabled</Status>
<Expiration>
<Days>1</Days>
</Expiration>
<AbortMultipartUpload>
<Days>1</Days>
</AbortMultipartUpload>
</Rule>
<Rule>
<ID>delete created before date</ID>
<Prefix>backup/</Prefix>
<Status>Enabled</Status>
<Expiration>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</Expiration>
<AbortMultipartUpload>
<CreatedBeforeDate>2014-10-11T00:00:00.000Z</CreatedBeforeDate>
</AbortMultipartUpload>
</Rule>
</LifecycleConfiguration>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:10 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Get Bucket (List Object)

The Get Bucket operation can be used to list all of the object information in a bucket.

Request syntax

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

Request parameters

When you initiate a GetBucket (ListObject) request, you can use prefix, marker, delimiter and max-uploads to prescribe a limit to the list to return partial results. Besides, encoding-type can be used to encode the following elements in the returned results: delimiter, marker, prefix, NextMarker, and key.

Name	Description
delimiter	A character used to group object names. All those objects whose names contain the specified prefix and behind which the delimiter occurs for the first time act as a group of elements - CommonPrefixes. Data type: string Default value: none
marker	Set the returned results to begin from the first entry after the marker in alphabetical order. Data type: string Default value: none
max-uploads	Limit the maximum number of objects returned for one request. If not specified, the default value is 100. The max-keys value cannot exceed 1,000. Data type: string Default value: 100
prefix	Limit that the returned object key must be prefixed accordingly. Note that the keys returned from queries using a prefix will still contain the prefix. Data type: string Default value: none
encoding-type	Specify the encoding of the returned content and the encoding type. Parameters delimiter, marker, prefix, NextMarker, and key use UTF-8 characters, but the XML 1.0 Standard does not support parsing certain control characters, such as characters with ASCII values ranging from 0 to 10. If some elements in the returned results contain characters that are not supported by the XML 1.0 Standard, encoding-type can be specified to encode these elements, such as delimiter, marker, prefix, NextMarker, and key. Data type: string Default value: none. Optional value: url

Response elements

Name	Description
Contents	Container used for saving every returned

	object meta. Type: container Parent node: ListBucketResult
CommonPrefixes	If the delimiter parameter is specified in the request, the response returned by the OSS contains the CommonPrefixes element. This element indicates the set of objects which end with a delimiter and have a common prefix. Type: string Parent node: ListBucketResult
Delimiter	A character used to group object names. All those objects whose names contain the specified prefix and behind which the delimiter occurs for the first time act as a group of elements - CommonPrefixes. Type: string Parent node: ListBucketResult
EncodingType	Specify the encoding type for the returned results. If encoding-type is specified in a request, the following elements in the returned results will be encoded: delimiter, marker, prefix, NextMarker, and key. Type: string Parent node: ListBucketResult
DisplayName	Name of the object owner. Type: string Parent node: ListBucketResult.Contents.Owner
ETag	The ETag (entity tag) is created when an object is generated and is used to indicate the content of the object. For an object created for a Put Object request, the value of ETag is the value of MD5 in the content of the object. For an object created in other approaches, the value of ETag is the UUID in the content of the object. The value of ETag can be used to check whether the content of the object is changed. It is not recommended that the ETag be used as the MD5 value of the object content to verify data integrity. Type: string Parent node: ListBucketResult.Contents
ID	User ID of the bucket owner. Type: string Parent node: ListBucketResult.Contents.Owner
IsTruncated	Specify whether all results have been returned; “true” means that not all results are returned this time; “false” means that all results are returned this time. Type: enumerative string Valid values: true, false Parent node: ListBucketResult
Key	The object key.

	Type: string Parent node: ListBucketResult.Contents
LastModified	Time when the object is last modified. Type: time Parent node: ListBucketResult.Contents
ListBucketResult	The container that saves the Get Bucket request results. Type: container Sub-nodes: Name, Prefix, Marker, MaxKeys, Delimiter, IsTruncated, Nextmarker, Contents Parent node: none
Marker	Mark the origin of the current Get Bucket (List Object) request. Type: string Parent node: ListBucketResult
MaxKeys	The maximum number of returned results in response to the request. Type: string Parent node: ListBucketResult
Name	Bucket name Type: string Parent node: ListBucketResult
Owner	Container used for saving the information about the bucket owner. Type: container Sub-nodes: DisplayName, ID Parent node: ListBucketResult
Prefix	Starting prefix for the current results of query. Type: string Parent node: ListBucketResult
Size	Number of bytes of the object. Type: string Parent node: ListBucketResult.Contents
StorageClass	Object storage type. Only the “Standard” type is available currently Type: string Parent node: ListBucketResult.Contents

Detail analysis

1. The user-defined meta in the object is not returned during the GetBucket request.
2. If the bucket to be accessed does not exist, or if you attempt to access a bucket which cannot be created due to non-standard naming, the error 404 Not Found with the error code: NoSuchBucket will be returned.
3. If you try to access a bucket without the required permission, the error 403 Forbidden with the error code: AccessDenied will be returned.
4. If listing cannot be completed at one time because of the max-uploads setting, a

<NextMarker> is appended to the returned result, prompting that this can be taken as a marker for continued listing. The value in NextMarker is still in the list result.

5. During a condition query, even if the marker does not actually exist in the list, what is returned is printed starting from the next to what conforms to the marker letter sorting. If the max-uploads value is less than 0 or greater than 1,000, the error 400 Bad Request is returned. Error code: InvalidArgument.
6. If the prefix, marker or delimiter parameters do not meet the length requirement, 400 Bad Request is returned. Error code: InvalidArgument.
7. The prefix and marker parameters are used to achieve display by pages, and the parameter length must be less than 1,024 bytes.
8. Setting prefix as the name of a folder enumerates the files starting with this prefix, recursively returning all files and sub-folders in this folder. If, in addition, we set the Delimiter as "/ ", the returned values will list the files in the folder and the subfolders will be returned in the CommonPrefixes section. Recursive files and folders in subfolders will not be displayed. For example, a bucket has the following three objects:fun/test.jpg, fun/movie/001.avi, and fun/movie/007.avi.If the prefix is set to "fun/ ", three objects are returned. If the delimiter is set to "/ " additionally, file "fun/test.jpg" and prefix "fun/movie/ " are returned. That is, the folder logic is achieved.

Scenario example

Four objects are available in the bucket "my_oss" and are respectively named as:

- oss.jpg
- fun/test.jpg
- fun/movie/001.avi
- fun/movie/007.avi

Example

Request example:

```
GET / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:BC+oQIXVR2/ZghT7cGa0ykboO4M=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1866
```

```
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Name>oss-example</Name>
<Prefix></Prefix>
<Marker></Marker>
<MaxKeys>100</MaxKeys>
<Delimiter></Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>fun/movie/001.avi</Key>
<LastModified>2012-02-24T08:43:07.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user-example</DisplayName>
</Owner>
</Contents>
<Contents>
<Key>fun/movie/007.avi</Key>
<LastModified>2012-02-24T08:43:27.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user-example</DisplayName>
</Owner>
</Contents>
<Contents>
<Key>fun/test.jpg</Key>
<LastModified>2012-02-24T08:42:32.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user-example</DisplayName>
</Owner>
</Contents>
<Contents>
<Key>oss.jpg</Key>
<LastModified>2012-02-24T06:07:48.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
```

```
<DisplayName>user-example</DisplayName>
</Owner>
</Contents>
</ListBucketResult>
```

Example of a request containing the prefix parameter:

```
GET /?prefix=fun HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:BC+oQIXVR2/ZghT7cGa0ykboO4M=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1464
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Name>oss-example</Name>
<Prefix>fun</Prefix>
<Marker> </Marker>
<MaxKeys>100</MaxKeys>
<Delimiter> </Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>fun/movie/001.avi</Key>
<LastModified>2012-02-24T08:43:07.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user_example</DisplayName>
</Owner>
</Contents>
<Contents>
<Key>fun/movie/007.avi</Key>
<LastModified>2012-02-24T08:43:27.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user_example</DisplayName>
</Owner>
</Contents>
```

```
<Contents>
<Key>fun/test.jpg</Key>
<LastModified>2012-02-24T08:42:32.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user_example</DisplayName>
</Owner>
</Contents>
</ListBucketResult>
```

Example of a request containing parameters prefix and delimiter:

```
GET /?prefix=fun/&delimiter=/ HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 08:43:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:DNrnx7xHk3sgysx7I8U9I9IY1vY=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 712
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<ListBucketResult xmlns=”  http://doc.oss-cn-hangzhou.aliyuncs.com”  >
<Name>oss-example</Name>
<Prefix>fun/</Prefix>
<Marker> </Marker>
<MaxKeys>100</MaxKeys>
<Delimiter>/</Delimiter>
<IsTruncated>false</IsTruncated>
<Contents>
<Key>fun/test.jpg</Key>
<LastModified>2012-02-24T08:42:32.000Z</LastModified>
<ETag>&amp;quot;5B3C1A2E053D763E1B002CC607C5A0FE&amp;quot;</ETag>
<Type>Normal</Type>
<Size>344606</Size>
<StorageClass>Standard</StorageClass>
<Owner>
<ID>00220120222</ID>
<DisplayName>user_example</DisplayName>
</Owner>
</Contents>
<CommonPrefixes>
<Prefix>fun/movie/</Prefix>
</CommonPrefixes>
```

```
</ListBucketResult>
```

Get Bucket ACL

Get Bucket ACL is used to obtain the access permissions for a bucket.

Request syntax

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

Response elements

Name	Description
AccessControlList	Container used for storing the ACL information Type: container Parent node: AccessControlPolicy
AccessControlPolicy	Specify the container that stores the Get Bucket ACL result Type: container Parent node: none
DisplayName	Name of the bucket owner. (Consistent with the ID at present) Type: string Parent node: AccessControlPolicy.Owner
Grant	ACL permissions of the bucket. Type: enumerative string Valid values: private，public-read，public-read-write Parent node: AccessControlPolicy.AccessControlList
ID	User ID of the bucket owner Type: string Parent node: AccessControlPolicy.Owner
Owner	Container used for saving the information about the bucket owner. Type: container Parent node: AccessControlPolicy

Detail analysis

1. Only the bucket owner can use the Get Bucket ACL interface.

Example

Request example:

```
GET /?acl HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 04:11:23 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0FmgbrQ0=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 04:11:23 GMT
Content-Length: 253
Content-Tupe: application/xml
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" ?>
<AccessControlPolicy>
<Owner>
<ID>00220120222</ID>
<DisplayName>user_example</DisplayName>
</Owner>
<AccessControlList>
<Grant>public-read</Grant>
</AccessControlList>
</AccessControlPolicy>
```

Get Bucket Location

Get Bucket Location is used to view the location information about the data center to which a bucket belongs.

Request syntax

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

```
Date: GMT Date
Authorization: SignatureValue
```

Response elements

Name	Description
LocationConstraint	Region where a bucket is located Type: string Values: oss-cn-hangzhou、oss-cn-qingdao、oss-cn-beijing、oss-cn-hongkong、oss-cn-shenzhen、oss-cn-shanghai

Detail analysis

1. Only the owner of a bucket can view the location information of the bucket. If other users attempt to access the location information, the error 403 Forbidden with the error code: AccessDenied will be returned.
2. LocationConstraint has the following valid values: oss-cn-hangzhou, oss-cn-qingdao, oss-cn-beijing, oss-cn-hongkong, oss-cn-shenzhen, oss-cn-shanghai, oss-us-west-1, oss-us-east-1, and oss-ap-southeast-1, which respectively correspond to Hangzhou data center, Qingdao data center, Beijing data center, Hong Kong data center, Shenzhen data center, Shanghai data center, US Silicon Valley data center, US Virginia data center and Asia-Pacific (Singapore) data center.

Example

Request example:

```
Get /?location HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3naA=
```

Response example with logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 15 Mar 2013 05:31:04 GMT
Connection: keep-alive
Content-Length: 90
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<LocationConstraint xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >oss-cn-hangzhou</LocationConstraint
>
```

Get Bucket Logging

Get Bucket Logging is used to view the access log configurations of a bucket.

Request syntax

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

Response elements

Name	Description
BucketLoggingStatus	The container for storing access log status information Type: container Child element: LoggingEnabled Parent element: none
LoggingEnabled	The container for storing access log information. This element is required only when server access logging is enabled. Type: container Child element: TargetBucket, TargetPrefix Parent element: BucketLoggingStatus
TargetBucket	The bucket for storing access logs. Type: character Child element: none Parent element: BucketLoggingStatus.LoggingEnabled
TargetPrefix	The prefix of the names of saved access log files. Type: character Child element: none Parent element: BucketLoggingStatus.LoggingEnabled

Detail analysis

1. If a bucket does not exist, the error “404 no content” will be returned. Error code: NoSuchBucket.

2. Only the owner of a bucket can view the access logging configuration of the bucket. If other users attempt to access the configuration, the error 403 Forbidden with the error code: AccessDenied will be returned.
3. If no logging rules are set for the source bucket, the OSS still returns an XML message body with the element BucketLoggingStatus being null.

Example

Request example:

```
Get /?logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 04 May 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3naA=
```

Response example with logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 210
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<LoggingEnabled>
<TargetBucket>mybucketlogs</TargetBucket>
<TargetPrefix>mybucket-access_log/</TargetPrefix>
</LoggingEnabled>
</BucketLoggingStatus>
```

Response example with no logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 110
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<BucketLoggingStatus xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
</BucketLoggingStatus>
```

Get Bucket Website

The Get Bucket Website operation is used to view the static website hosting status of a bucket.

Request syntax

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

Response elements

Name	Description
ErrorDocument	The parent element of the child element key Type: container Parent element: WebsiteConfiguration
IndexDocument	The parent element of the child element suffix Type: container Parent element: WebsiteConfiguration
Key	The file name used to return Error 404 Type: string Parent element: WebsiteConfiguration.ErrorDocument This element is required when ErrorDocument is set
Suffix	The index file name added when a directory URL is returned. This element cannot be empty or contain a slash (/). For example, if the index file index.html is configured, oss-cn-hangzhou.aliyuncs.com/mybucket/mydir/ contained in an access request is converted into oss-cn-hangzhou.aliyuncs.com/mybucket/index.html by default. Type: string Parent element: WebsiteConfiguration.IndexDocument
WebsiteConfiguration	Requested container Type: container Parent element: none

Detail analysis

1. If a bucket does not exist, the error “404 no content” will be returned. Error code:

NoSuchBucket.

2. Only the owner of a bucket can view the static website hosting status of the bucket. If other users attempt to access the status information, the error 403 Forbidden with the error code: AccessDenied will be returned.
3. If the source bucket is not configured with static website hosting, OSS returns Error 404 with the error code: NoSuchWebsiteConfiguration.

Example

Request example:

```
   Get /?website HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51NNHD39zXw=
```

Response example with logging rules configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<WebsiteConfiguration xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<IndexDocument>
<Suffix>index.html</Suffix>
</IndexDocument>
<ErrorDocument>
<Key>error.html</Key>
</ErrorDocument>
</WebsiteConfiguration>
```

Return example with LOG rules not set

```
HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<Error xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Code>NoSuchWebsiteConfiguration</Code>
<Message>The specified bucket does not have a website configuration.</Message>
<BucketName>oss-example</BucketName>
```

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

Get Bucket Referer

The Get Bucket Referer operation is used to view the referer configuration of a bucket. For details about bucket referer, see OSS Anti-Leech.

Request syntax

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

Response elements

Name	Description
RefererConfiguration	The container that saves the Referer configuration content Type: container Sub-nodes: AllowEmptyReferer node and RefererList node Parent node: none
AllowEmptyReferer	Specify whether the access request with the referer field being null is allowed. Type: enumerative string Valid value: true or false `Default value: true Parent node: RefererConfiguration
RefererList	The container that saves the referer access white list. Type: container Parent node: RefererConfiguration Sub-node: Referer
RefererList	Specify a referer access white list. Type: string Parent node: RefererList

Detail analysis

1. If the bucket does not exist, error 404 is returned. Error code: NoSuchBucket.
2. Only the owner of a bucket can view the referer configuration of the bucket. If other users attempt to access the configuration, the error 403 Forbidden with the error code: AccessDenied will be returned.
3. If no referer configuration has been conducted for the bucket, the OSS returns the default AllowEmptyReferer value and an empty RefererList.

Example

Request example:

```
Get /?referer HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51NNHD39zXw=
```

Response example with a referer rule configured for the bucket:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: keep-alive
Content-Length: 218
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
<RefererList>
<Referer> http://www.aliyun.com</Referer>
<Referer> https://www.aliyun.com</Referer>
<Referer> http://www.*.com</Referer>
<Referer> https://www.?.aliyuncs.com</Referer>
</RefererList>
</RefererConfiguration>
```

Response example with no referer rule configured for the bucket:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: keep-alive
Content-Length: 308
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<RefererConfiguration>
<AllowEmptyReferer>true</AllowEmptyReferer >
```

```
< RefererList />
</RefererConfiguration>
```

Get Bucket Lifecycle

Get Bucket Lifecycle is used to view the lifecycle configuration of a bucket.

Request syntax

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

Detail analysis

1. Only the owner of a bucket can view the lifecycle configuration of the bucket. If other users attempt to access the configuration, the error 403 Forbidden with the error code: AccessDenied will be returned.
2. If the bucket or lifecycle does not exist, the error 404 Not Found with the error code: NoSuchBucket or NoSuchLifecycle will be returned.

Example

Request example:

```
Get /?lifecycle HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Date: Mon, 14 Apr 2014 01:17:29 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3naA=
```

Response example with bucket lifecycle configured:

```
HTTP/1.1 200
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: keep-alive
Content-Length: 255
Server: AliyunOSS

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

```
<LifecycleConfiguration>
<Rule>
<ID>delete after one day</ID>
<Prefix>logs/</Prefix>
<Status>Enabled</Status>
<Expiration>
<Days>1</Days>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

Response example with no bucket lifecycle configured:

```
HTTP/1.1 404
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: keep-alive
Content-Length: 278
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<Error>
<BucketName>oss-example</BucketName>
<Code>NoSuchLifecycle</Code>
<Message>No Row found in Lifecycle Table.</Message>
<RequestId>534B372974E88A4D89060099</RequestId>
<HostId> oss-example.oss.aliyuncs.com</HostId>
</Error>
```

Delete Bucket

The Delete Bucket interface is used to delete a bucket.

Request syntax

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

Detail analysis

1. If a bucket does not exist, the error “404 no content” will be returned. Error code: NoSuchBucket.

2. To prevent accidental deletion, OSS does not allow users to delete a non-empty bucket.
3. If you try to delete a non-empty bucket, the error 409 Conflict with the error code: BucketNotEmpty will be returned.
4. Only the bucket owner has the permission to delete the bucket. If you try to delete a bucket you have no permission for, the error 403 Forbidden will be returned. Error code: AccessDenied.

Example

Request example:

```
DELETE / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:31:04 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:ceOEyZavKY4QcjoUWYSpYbJ3naA=
```

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:31:04 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Delete Bucket Logging

The Delete Bucket Logging interface is used to disable the access logging function of a bucket.

Request syntax

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

Detail analysis

1. If the bucket does not exist, the error 404 No Content with the error code: NoSuchBucket will be returned.

2. Only the bucket owner can disable the access logging function for the bucket. If you try to operate a bucket which does not belong to you, OSS returns the error 403 Forbidden with the error code: AccessDenied.
3. If the access logging function is not enabled for the target bucket, HTTP status code 204 is returned.

Example

Request example

```
DELETE /?logging HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:35:24 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:6ZVHOehYzxoC1yxRydPQs/CnMZU=
```

Return example

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:35:24 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Delete Bucket Website

The Delete Bucket Website interface is used to disable the static website hosting mode of a bucket.

Request syntax

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

Detail analysis

1. If the bucket does not exist, the error 404 No Content with the error code: NoSuchBucket will be returned.
2. Only the bucket owner can disable the bucket’ s static website hosting mode. If you try to

operate a bucket which does not belong to you, OSS returns the error 403 Forbidden with the error code: AccessDenied.

Example

Request example:

```
DELETE /?website HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:LnM4AZ1OeIduZF5vGFWicOMEkVg=
```

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Delete Bucket Lifecycle

The Delete Bucket Lifecycle interface is used to delete the lifecycle configuration of a specified bucket.

Request syntax

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

Detail analysis

1. This operation deletes all lifecycle rules of a specified bucket. After that, no objects are automatically deleted in this bucket.
2. If the bucket or lifecycle does not exist, the error 404 Not Found with the error code: NoSuchBucket or NoSuchLifecycle will be returned.
3. Only the bucket owner can delete the lifecycle configuration of a bucket. If you try to

operate a bucket which does not belong to you, OSS returns the error 403 Forbidden with the error code: AccessDenied.

Example

Request example:

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

Response example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B371674E88A4D8906008B
Date: Mon, 14 Apr 2014 01:17:35 GMT
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
```

Object Operations

Put Object

The Put Object interface is used to upload files.

Request syntax

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

Request header

Name	Description

Cache-Control	Specify the web page caching behavior when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Disposition	Specify the name of the object when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Encoding	Specify the content encoding format when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-MD5	As defined in RFC 1864, the message content (excluding the header) is calculated to obtain an MD5 value, which is a 128-bit number. Then this number is encoded using Base64 into a Content-MD5 value. This request header can be used for checking the validity of a message, that is, whether the message content is consistent with the sent content. Although this request header is optional, the OSS recommends that you use this request header for an end-to-end check. Type: string Default value: none Limits: none
Expires	Specify the expiration time. For details, refer to RFC2616. Type: string Default value: none **Attention**: OSS will impose no limits or verification on this value
x-oss-server-side-encryption	Specify the server-side encryption algorithm when the OSS creates an object. Type: string Valid value: AES256
x-oss-object-acl	Specify the access permission when the OSS creates an object. Type: string Valid values: public-read, private, and public-read-write

Detail analysis

- If you have uploaded the Content-MD5 request header, the OSS will calculate the body’ s Content-MD5 and check if the two are consistent. If the two are different, the error code

InvalidDigest will be returned.
- If the Content-Length value in the request header is smaller than the length of data transmitted in the actual request body, the OSS still creates a file, but the object size is equal to the size defined by Content-Length, and the remaining data will be dropped.
- If a file of the same name with the object to be added already exists, and you are authorized to access this object, the newly-added file will overwrite the existing file, and the system will return the 200 OK message.
- If the PutObject request carries a parameter prefixed with x-oss-meta-,the parameter is treated as user meta, for example, x-oss-meta-location. A single object can have multiple similar parameters, but the total size of all user meta cannot exceed 8 KB.
- If the Content length parameter is not added to the header, the system will return the 411 Length Required error. Error code: MissingContentLength.
- If the length is set, but the message body is not sent, or the size of the sent body is smaller than the specified size, the server will wait until timeout, and then return the 400 Bad Request message. Error code: RequestTimeout.
- If the bucket of the object to be added does not exist, the system will return the 404 Not Found error. Error code: NoSuchBucket.
- If you have no permission to access the bucket of the object to be added, the system will return the 403 Forbidden error. Error code: AccessDenied.
- If the length of the added file exceeds 5 GB, the system will return the 400 Bad Request message. Error code: InvalidArgument.
- If the length of the input object key exceeds 1,023 bytes, the system will return the 400 Bad Request message. Error code: InvalidObjectName.
- When you put an object, the OSS supports the following five header fields defined in RFC2616: Cache-Control, Expires, Content-Encoding, Content-Disposition and Content-Type. If these headers are set when you upload an object, the corresponding header values will be automatically set to the uploaded values next time when this object is downloaded.
- If the x-oss-server-side-encryption header is specified when you upload an object, the value of this header must be set to AES256. Otherwise, the system will return the 400 error and the error code: InvalidEncryptionAlgorithmError. After this header is specified, the response header will also contain this header, and the OSS stores the encryption algorithm of the uploaded object. When this object is downloaded, the response header will contain x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this object.

FAQs

Content-MD5 calculation method error

The uploaded content "123456789" is used as an example. The Content-MD5 value of the string should be calculated

as follows:
The algorithm defined in related standards can be simplified to the following:

1. Calculate the MD5-encrypted 128-bit binary array.
2. Encode the binary array (instead of the 32-bit string code) with Base64.

Python is used as an example.
The correct calculation code is:
```
>>> import base64,hashlib
>>> hash = hashlib.md5()
>>> hash.update("0123456789")
>>> base64.b64encode(hash.digest())
'eB5eJF1ptWaXm4bijSPyxw=='
```

Note:
The correct code is: hash.digest(), used to calculate a 128-bit binary array
```
>>> hash.digest()
'x\x1e^$]i\xb5f\x97\x9b\x86\xe2\x8d#\xf2\xc7'
```

A common error is encoding the calculated 32-bit string code with Base64.
An incorrect example: hash.hexdigest(), and a visible 32-bit string will be calculated.
```
>>> hash.hexdigest()
'781e5e245d69b566979b86e28d23f2c7'
```
Result of encoding the incorrect MD5 value with Base64:
```
>>> base64.b64encode(hash.hexdigest())
'NzgxZTVlMjQ1ZDY5YjU2Njk3OWI4NmUyOGQyM2YyYzc='
```

Example

Request example:

```
PUT /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Cache-control: no-cache
Expires: Fri, 28 Feb 2012 05:38:42 GMT
Content-Encoding: utf-8
Content-Disposition: attachment;filename=oss_download.jpg
Date: Fri, 24 Feb 2012 06:03:28 GMT
Content-Type: image/jpg
Content-Length: 344606
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2PRrk=

[344606 bytes of object data]
```

Response example:

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

Copy Object

The Copy Object operation is used to copy an existing object in the OSS into another object. To copy an existing object in the OSS into another object, you can send a PUT request to the OSS, and add the element “x-oss-copy-source” to the PUT request header to specify the copy source. The OSS automatically determines that this is a copy operation, and directly performs this operation on the server side. If the copy operation is successful, the system will return the information of the new object to you.

This operation is applicable to files smaller than 1 GB. To copy a file greater than 1 GB, you must use the Multipart Upload operation. For details, see Upload Part Copy.

Request syntax

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

Request header

Name	Description
x-oss-copy-source	Specify the copy source address (the requester must have the permission to read the source object) Type: string Default value: none
x-oss-copy-source-if-match	If the source object’ s ETag value is the same as the ETag value provided by the user, a copy operation will executed, and the code 200 will be returned. Otherwise, the system will return the HTTP error code 412 (indicating preprocessing failure). Type: string Default value: none
x-oss-copy-source-if-match	If the source object’ s ETag value is not the same as the ETag value provided by the user, a copy operation will be executed, and the code 200 will be returned. Otherwise, the system will return the HTTP error code 304 (indicating preprocessing failure). Type: string Default value: none

x-oss-copy-source-if-unmodified-since	If the time specified by the input parameter is the same as or later than the actual modification time of the file, the system will transfer the file normally, and return the 200 OK message; otherwise, the system will return the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-modified-since	If the source object has been modified after the time specified by the user, the system will perform a copy operation. Otherwise, the system will return the 304 HTTP error code (indicating preprocessing failure). Type: string Default value: none
x-oss-metadata-directive	Valid values include COPY and REPLACE. If this parameter is set to COPY, the system will copy meta for the new object from the source object. If this parameter is set to REPLACE, the system will ignore all meta values of the source object, and use the meta value specified in this request. If this parameter is set to a value other than COPY and REPLACE, the system will return the 400 HTTP error code. Note that when the value is COPY, the source object' s x-oss-server-side-encryption meta value will not be copied. Type: string Default value: COPY Valid value: COPY and REPLACE
x-oss-server-side-encryption	Specify the server-side encryption algorithm when the OSS creates the target object. Type: string Valid value: AES256
x-oss-object-acl	Specify the access permission when the OSS creates an object. Type: string Valid values: public-read, private, and public-read-write

Response elements

Name	Description
CopyObjectResult	Result of Copy Object. Type: string Default value: none
ETag	ETag value of the new object. Type: string Parent element: CopyObjectResult
LastModified	Last update time of the new object.

	Type: string Parent element: CopyObjectResult

Detail analysis

- You can use the copy operation to modify the meta information of an existing object.
- If the source object address is the same as the target object address in the copy operation, the system will directly replace the meta information in the source object regardless of the value of x-oss-metadata-directive.
- The OSS allows the copy operation to contain any number of the four pre-judgment headers. For details about the logic, see Detail Analysis of Get Object.
- To complete a copy operation, the requester must have the permission to read the source object.
- The source object and the target object must belong to the same data center; otherwise, the system will return the error code 403 AccessDenied. The error message is: Target object does not reside in the same data center as source object.
- In the billing statistics of the copy operation, the number of Get requests will increase by 1 in the bucket of the source object, the number of Put requests will increase by 1 in the bucket of the target object, and a storage space will be added accordingly.
- In the copy operation, all relevant request headers start from “x-oss-”, and therefore must be added to the signature string.
- If the x-oss-server-side-encryption header is specified in the copy request, and its value (AES256) is valid, the target object will be encrypted on the server side after the copy operation is performed no matter whether the source object has been encrypted on the server side or not. In addition, the copy operation response header will contain x-oss-server-side-encryption, the value of which is set to the encryption algorithm of the target object. When this target object is downloaded, the response header will also contain x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this target object. If the x-oss-server-side-encryption request header is not specified in the copy operation, the target object will be the data that is not encrypted on the server side no matter whether the source object has been encrypted on the server side or not.
- When the x-oss-metadata-directive header in the copy request is set to COPY (default value), the system will not copy the x-oss-server-side-encryption value of the source object. That is, the target object will be encrypted on the server side only when x-oss-server-side-encryption request header is specified accordingly in the copy request.
- When the x-oss-server-side-encryption request header is specified in the copy operation, and the request value is not AES256, the system will return the error code 400 and the error message: InvalidEncryptionAlgorithmError.
- If the size of the file to be copied is greater than 1 GB, the system will return the error code 400 and the error message: EntityTooLarge.
- This operation cannot copy objects that are generated through the appended upload method.
- If the file type is **symbolic link**, only the symbolic links will be copied.

Example

Request example:

```
PUT /copy_oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 07:18:48 GMT
x-oss-copy-source: /oss-example/oss.jpg
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:gmnwPKuu20LQEjd+iPkL259A+n0=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Content-Type: application/xml
Content-Length: 193
Connection: keep-alive
Date: Fri, 24 Feb 2012 07:18:48 GMT
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<CopyObjectResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<LastModified>Fri, 24 Feb 2012 07:18:48 GMT</LastModified>
<ETag>"5B3C1A2E053D763E1B002CC607C5A0FE"</ETag>
</CopyObjectResult>
```

Get Object

The Get Object operation is used to obtain an object. This operation requires that you must have the read permission to the object.

Request syntax

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

Request parameters

OSS supports customizing some headers in the OSS response request when you send a GET request, on the premise that the GET request you send must carry a signature. These headers include:

Name	Description
response-content-type	Specify the content-type header in the OSS response request Type: string Default value: none
response-content-language	Specify the content-language header in the OSS response request Type: string Default value: none
response-expires	Specify the expires header in the OSS response request Type: string Default value: none
response-cache-control	Specify the cache-control header in the OSS response request Type: string Default value: none
response-content-disposition	Specify the content-disposition header in the OSS response request Type: string Default value: none
response-content-encoding	Specify the content-encoding header in the OSS response request Type: string Default value: none

Request header

Name	Description
Range	Specify the range of file transfer. For example, if Range is set to bytes=0-9, the system will transfer byte 0 to byte 9. Type: string Default value: none
If-Modified-Since	If the specified time is earlier than the actual modification time, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 304 Not Modified message. Type: string Default value: none Time format: GMT. For example, Fri, 13 Nov 2015 14:47:53 GMT
If-Unmodified-Since	If the time specified by the input parameter is the same with or later than the actual

	modification time of the file, the system will transfer the file normally, and return the 200 OK message; otherwise, the system will return the 412 Precondition Failed error Type: string Default value: none Time format: GMT. For example, Fri, 13 Nov 2015 14:47:53 GMT
If-Match	If the input expected ETag matches the ETag of the object, the system will transfer the file normally, and return the 200 OK message; otherwise, the system will return the 412 Precondition Failed error Type: string Default value: none
If-None-Match	If the input ETag does not match the ETag of the object, the system will transfer the file normally, and return the 200 OK message; otherwise, the system will return the 304 Not Modified message. Type: string Default value: none

Detail analysis

- The Range parameter in the Get Object request can be set to support resumable data transfer from breakpoints. This feature is recommended if the object size is large.
- If the Range parameter is specified in the request header, the returned message will contain the length of the entire file and the range returned this time. For example, Content-Range: bytes 0-9/44 indicates that the length of the entire file is 44, and the range returned this time is 0–9. If the range requirement is not met, the system will transfer the entire file and does not include Content-Range in the result.
- If the time specified in the "If-Modified-Since" element does not match the specification, the system will directly return the file, as well as the 200 OK message.
- If-Modified-Since can coexist with If-Unmodified-Since. If-Match can also coexist with If-None-Match.
- If the request contains If-Unmodified-Since and If-Unmodified-Since does not match the actual modification time, or the request contains If-Match and If-Match does not match the Etag of the object, the system will return the 412 Precondition Failed message.
- If the request contains If-Modified-Since and If-Modified-Since does not match the actual modification time, or the request contains If-None-Match and If-None-Match does not match the Etag of the object, the system will return the 304 Not Modified message.
- If the file does not exist, the system will return the 404 Not Found error. Error code: NoSuchKey.
- The OSS does not allow you to customize the headers in the OSS response request by using request parameters in the GET request during an anonymous access.

- When you customize some headers in the OSS response request, the OSS will set these headers to the values specified in parameters in the GET Object request only when the request is successfully processed, that is, when the system returns the 200 OK message.
- If this object is entropy encrypted on the server side, the system will automatically return the decrypted object on receiving the GET Object request, and return x-oss-server-side-encryption in the response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm of the object.
- If you need to compress and transfer the returned content using GZIP, you need to add Accept-Encoding:gzip to the display mode in the request header. The OSS will determine whether to return the data compressed by GZIP to you based on the Content-Type and size of the file. If the content is compressed using GZIP, the content will not contain the Etag. Currently, the OSS supports GZIP compression for the following Content-Types: HTML, JavaScript, CSS, XML, RSS, and JSON, and the file size must be at least 1 KB.
- If the file type is **symbolic link**, the content of the target file will be returned. In addition, the Content-Length, ETag and Content-Md5 parameters in the response header will be the meta information of the target file; the Last-Modified parameter will be the maximum value of the target file and symbolic link; all the other parameters will be the meta information of the symbolic link.
- If the file type is **symbolic link**, and the target file does not exist, the system will return the 404 Not Found error. Error code: SymlinkTargetNotExist.
- If the file type is **symbolic link**, and the target file type is symbolic link, the system will return the 400 Bad Request error. Error code: InvalidTargetType.

Example

Request example:

```
GET /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:38:30 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJCZkcde6OhZ9Jfe8=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 3a89276f-2e2d-7965-3ff9-51c875b99c41
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 06:38:30 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Content-Type: image/jpg
Content-Length: 344606
Server: AliyunOSS

[344606 bytes of object data]
```

Request example with range specified:

```
GET //oss.jpg HTTP/1.1
Host:oss-example. oss-cn-hangzhou.aliyuncs.com
Date: Fri, 28 Feb 2012 05:38:42 GMT
Range: bytes=100-900
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:qZzjF3DUtd+yK16BdhGtFcCVknM=
```

Response example:

```
HTTP/1.1 206 Partial Content
x-oss-request-id: 28f6508f-15ea-8224-234e-c0ce40734b89
x-oss-object-type: Normal
Date: Fri, 28 Feb 2012 05:38:42 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Accept-Ranges: bytes
Content-Range: bytes 100-900/344606
Content-Type: image/jpg
Content-Length: 801
Server: AliyunOSS

[801 bytes of object data]
```

Request example with a customized returned message header:

```
GET /oss.jpg?response-expires=Thu%2C%2001%20Feb%202012%2017%3A00%3A00%20GMT& response-content-type=text&response-cache-control=No-cache&response-content-disposition=attachment%253B%2520filename%253Dtesting.txt&response-content-encoding=utf-8&response-content-language=%E4%B8%AD%E6%96%87 HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com:
Date: Fri, 24 Feb 2012 06:09:48 GMT
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 06:09:48 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE "
Content-Length: 344606
Connection: keep-alive
Content-disposition: attachment; filename:testing.txt
Content-language: Chinese
Content-encoding: utf-8
Content-type: text
Cache-control: no-cache
Expires: Fri, 24 Feb 2012 17:00:00 GMT
Server: AliyunOSS
```

```
[344606 bytes of object data]
```

Symbolic link request example:

```
GET /link-to-oss.jpg HTTP/1.1
Accept-Encoding: identity
Date: Tue, 08 Nov 2016 03:17:58 GMT
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:qZzjF3DUtd+yK16BdhGtFcCVknM=
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 08 Nov 2016 03:17:58 GMT
Content-Type: application/octet-stream
Content-Length: 20
Connection: keep-alive
x-oss-request-id: 582143E6D3436A212ADCC87D
Accept-Ranges: bytes
ETag: "8086265EFC0211ED1F9A2F09BF462227"
Last-Modified: Tue, 08 Nov 2016 03:17:58 GMT
x-oss-object-type: Symlink
Content-MD5: gIYmXvwCEe0fmi8Jv0YiJw==
```

Append Object

The Append Object operation is used to upload files in the appending write mode. The type of the objects created with the Append Object operation is Appendable Object, while the type of the objects uploaded with the Put Object operation is Normal Object.

Request syntax

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

Request header

Name	Description

Cache-Control	Specify the web page caching behavior when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Disposition	Specify the name of the object when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Encoding	Specify the content encoding format when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-MD5	As defined in RFC 1864, the message content (excluding the header) is calculated to obtain an MD5 value, which is a 128-bit number. Then this number is encoded using Base64 into a Content-MD5 value. This request header can be used for checking the validity of a message, that is, whether the message content is consistent with the sent content. Although this request header is optional, the OSS recommends that you use this request header for an end-to-end check. Type: string Default value: none Limits: none
Expires	Specify the expiration time. For details, refer to RFC2616. Type: integer Default value: none
x-oss-server-side-encryption	Specify the server-side encryption algorithm when the OSS creates an object. Type: string Valid value: AES256
x-oss-object-acl	Specify the access permission when the OSS creates an object. Type: string Valid values: public-read, private, and public-read-write

Response header

Name	Description
x-oss-next-append-position	Specify the position that should be provided in the next request. It is in fact the current object length. This header is contained when a successful message is returned for Append

	Object, or when a 409 error occurs due to mismatching of the position and the object length. Type: 64-bit integer
x-oss-hash-crc64ecma	Specify the 64-bit CRC value of the object. This 64-bit CRC is calculated according to ECMA-182. Type: 64-bit integer

Association with other operations

- Append Object is not applicable to a non-appendable object. For example, if a normal object with the same name already exists and the Append Object operation is still performed, the system will return the 409 error and the error code is: ObjectNotAppendable.
- If you perform the Put Object operation on an existing appendable object, this appendable object will be overwritten by the new object, and the type of this object will be changed to Normal Object.
- After the Head Object operation is performed, the system will return x-oss-object-type, which indicates the type of the object. If the object is an appendable object, the value of x-oss-object-type is Appendable. For an appendable object, after the Head Object operation is performed, the system will also return x-oss-next-append-position and x-oss-hash-crc64ecma.
- In the response XML of the Get Bucket (List Objects) request, the type of an appendable object will be set to Appendable.
- You cannot use Copy Object to copy an appendable object or change the encryption attribute of this object on the server. You can use Copy Object to modify the custom metadata.

Detail analysis:

- The two URL parameters, append and position, are both CanonicalizedResource, and must be contained in the signature.
- URL parameters must also contain append which specifies that the operation is an Append Object operation.
- URL query parameters must contain position which specifies the position where the appending starts. The value of position in the first Append Object operation must be 0, and the value of position in the subsequent operation is the current object length. For example, if the value of position specified in the first Append Object request is 0, and the value of content-length is 65536, the value of position specified in the second Append Object request must be set to 65536. After each operation succeeds, x-oss-next-append-position in the response header will also specify the position of the next Appendix Object request.
- If the position value is different from the current object length, the OSS will return the 409 error and the error code is: PositionNotEqualToLength. If such an error occurs, you can

obtain the position for the next Append Object request from x-oss-next-append-position in the response header, and send an Append Object request again.
- If the position value is 0 and an appendable object with the same name does not exist, or if the length of an appendable object with the same name is 0, the Append Object operation will be successful; otherwise, the system regards that the position and object length are mismatched.
- If the position value is 0 and an object with the same name does not exist, request headers (such as x-oss-server-side-encryption) can be set in the Append Object request as in the Put Object request. This is the same as in the case of Initiate Multipart Upload. If the position value is 0, and the correct x-oss-server-side-encryption header is added to the request, the header of the response to the subsequent Append Object request will also contain x-oss-server-side-encryption, which indicates the encryption algorithm. If meta needs to be modified later, you can use the Copy Object request.
- Due to the concurrency, even if you set the value of position to x-oss-next-append-position, this request may still fail due to PositionNotEqualToLength.
- The length limit of an object generated by Append Object is the same as that of an object generated by Put Object.
- After each Append Object operation, the last modification time of this object will be updated.
- If the position value is correct and the content with the length of 0 is appended to an existing appendable object, this operation will not change the status of the object.

CRC64 calculation method

The CRC of an appendable object is calculated according to ECMA-182. Its calculation method is the same as that of XZ. CRC64 can be calculated as follows using the Boost CRC module:

```
typedef boost::crc_optimal<64, 0x42F0E1EBA9EA3693ULL, 0xffffffffffffffffULL, 0xffffffffffffffffULL, true, true>
boost_ecma;

uint64_t do_boost_crc(const char* buffer, int length)
{
boost_ecma crc;
crc.process_bytes(buffer, length);
return crc.checksum();
}
```

Alternatively, CRC64 can be calculated as follows using the Python crcmod:

```
do_crc64 = crcmod.mkCrcFun(0x142F0E1EBA9EA3693L, initCrc=0L, xorOut=0xffffffffffffffffL, rev=True)

print do_crc64( “123456789” )
```

Example

Request example:

```
POST /oss.jpg?append&position=0 HTTP/1.1
Host: oss-example.oss.aliyuncs.com
Cache-control: no-cache
Expires: Wed, 08 Jul 2015 16:57:01 GMT
Content-Encoding: utf-8
Content-Disposition: attachment;filename=oss_download.jpg
Date: Wed, 08 Jul 2015 06:57:01 GMT
Content-Type: image/jpg
Content-Length: 1717
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2PRrk=

[1717 bytes of object data]
```

Response example:

```
HTTP/1.1 200 OK
Date: Wed, 08 Jul 2015 06:57:01 GMT
ETag: "0F7230CAA4BE94CCBDC99C5500000000"
Connection: keep-alive
Content-Length: 0
Server: AliyunOSS
x-oss-hash-crc64ecma: 14741617095266562575
x-oss-next-append-position: 1717
x-oss-request-id: 559CC9BDC755F95A64485981
```

Delete Object

The Delete Object operation is used to delete an object.

Request syntax

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

Detail analysis

- To delete an object with Delete Object, you must have the write permission to this object.

- If the object to be deleted does not exist, the OSS will return the 204 No Content status code.
- If the bucket of the object does not exist, the system will return 404 Not Found.
- If the file type is **symbolic link**, only the symbolic links will be deleted.

Example

Request example:

```
DELETE /copy_oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 07:45:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:zUglwRPGkbByZxm1+y4eyu+NIUs=
```

Response example:

```
HTTP/1.1 204 NoContent
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Fri, 24 Feb 2012 07:45:28 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Delete Multiple Objects

The Delete Multiple Objects operation allows you to delete multiple objects in the same bucket with one HTTP request. You can perform the Delete Multiple Objects operation to delete up to 1,000 objects with one request. Two response modes are available: the Verbose mode and the Quiet mode.

- Verbose mode: The message body returned by the OSS contains the result of each deleted object.
- Quiet mode: The message body returned by the OSS only contains the results for objects which encountered an error in the DELETE process. If all objects are deleted successfully, there will be no message body.

Request syntax

```
POST /?delete HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: ContentLength
Content-MD5: MD5Value
```

```
Authorization: SignatureValue

<?xml version="1.0" encoding="UTF-8"?>
<Delete>
<Quiet>true</Quiet>
<Object>
<Key>key</Key>
</Object>
...
</Delete>
```

Request parameters

During the Delete Multiple Objects operation, you can use encoding-type to encode the Key in the returned result.

Name	Description
encoding-type	Specify the encoding type of the Key in the returned result. Currently, the URL encoding is supported. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Data type: string Default value: none. Optional value: url

Request elements

Name	Description
Delete	Specify the container that saves the Delete Multiple Objects request. Type: container Sub-nodes: one or more object elements and the optional quiet element Parent node: none.
Key	Specify the name of the deleted object. Type: string Parent node: Object
Object	Specify the container that saves the information about the object. Type: container Sub-node: key Parent node: Delete.
Quiet	Enables the “Quiet” response mode. Type: enumerative string Valid values: true, false

	Default value: false Parent node: Delete

Response elements

Name	Description
Deleted	Specify the container that saves the successfully deleted objects. Type: container Sub-node: key Parent node: DeleteResult
DeleteResult	Specify the container that saves the result of the Delete Multiple Objects request. Type: container Sub-node: Deleted Parent node: none
Key	Specify the name of the object on which the OSS performs the Delete operation. Type: string Parent node: Deleted
EncodingType	Specify the encoding type for the returned results. If encoding-type is specified in the request, the Key will be encoded in the returned result. Type: string Parent node: Container

Detail analysis

- The Content-Length and Content-MD5 fields must be specified in the Delete Multiple Objects request. The OSS will verify that the received message body is correct based on the two fields before performing the Delete operation.
- Method to generate the Content-MD5 field: Encrypt the MD5 value of the Delete Multiple Objects request to obtain a 128-byte array, and encode the array using Base64. The final string obtained will be the content of the Content-MD5 field.
- The return mode of the Delete Multiple Objects request is Verbose by default.
- If the Delete Multiple Objects request is used to delete a non-existing object, the operation is still regarded as successful.
- The Delete Multiple Objects request can contain a message body of up to 2 MB. If the size of the message body exceeds 2 MB, the system will return the MalformedXML error code.
- The Delete Multiple Objects request can be used to delete up to 1,000 objects at a time. If the number of objects to be deleted at a time exceeds 1,000, the system will return the MalformedXML error code.
- If you have uploaded the Content-MD5 request header, the OSS will calculate the body’ s

Content-MD5 and check if the two are consistent. If the two are different, the error code InvalidDigest will be returned.

Example

Request example I:

```
POST /?delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length:151
Content-MD5: ohhnqLBJFiKkPSBO1eNaUA==
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:+z3gBfnFAxBcBDgx27Y/jEfbfu8=

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

<Delete>
<Quiet>false</Quiet>
<Object>
<Key>multipart.data</Key>
</Object>
<Object>
<Key>test.jpg</Key>
</Object>
<Object>
<Key>demo.jpg</Key>
</Object>
</Delete>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 78320852-7eee-b697-75e1-b6db0f4849e7
Date: Wed, 29 Feb 2012 12:26:16 GMT
Content-Length: 244
Content-Type: application/xml
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<DeleteResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Deleted>
<Key>multipart.data</Key>
</Deleted>
<Deleted>
<Key>test.jpg</Key>
</Deleted>
<Deleted>
<Key>demo.jpg</Key>
</Deleted>
</DeleteResult>
```

Request example Ⅱ:

```
POST /?delete HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length:151
Content-MD5: ohhnqLBJFiKkPSBO1eNaUA==
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:WuV0Jks8RyGSNQrBca64kEExJDs=

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

<Delete>
<Quiet>true</Quiet>
<Object>
<Key>multipart.data</Key>
</Object>
<Object>
<Key>test.jpg</Key>
</Object>
<Object>
<Key>demo.jpg</Key>
</Object>
</Delete>
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Head Object

The Head Object operation is used to return the meta information of a certain object without returning the file content.

Request syntax

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

Request header

Name	Description
If-Modified-Since	If the specified time is earlier than the actual modification time, the system returns the 200 OK message and the object meta; otherwise, the system returns the 304 Not Modified message. Type: string Default value: none
If-Unmodified-Since	If the time specified by the received parameter is the same as or later than the actual modification time of the file, the system returns the 200 OK message and the object meta; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
If-Match	If the received ETag matches the ETag of the object, the system returns the 200 OK message and the object meta; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
If-None-Match	If the received ETag does not match the ETag of the object, the system returns the 200 OK message and the object meta; otherwise, the system returns the 304 Not Modified message. Type: string Default value: none

Detail analysis

- After the Head Object request is sent, no message body is returned no matter whether the system returns the 200 OK message or an error message.
- The If-Modified-Since, If-Unmodified-Since, If-Match, and If-None-Match query conditions can be set in the header of the Head Object request. For the detailed setting rules, refer to the related fields in the Get Object request. If no modification is made, the system will return the 304 Not Modified message.
- If you upload the user meta prefixed with x-oss-meta- when sending a Put Object request, for example, x-oss-meta-location, the user meta will be returned.
- If the file does not exist, the system will return the 404 Not Found error.
- If this object is entropy encrypted on the server side, the system returns x-oss-server-side-encryption in the header of the response to the Head Object request. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm of the object.

- If the file type is **Symbolic Link**, the Content-Length, ETag and Content-Md5 parameters in the response header will be the meta information of the target file; the Last-Modified parameter will be the maximum value of the target file and symbolic link; all the other parameters will be the meta information of the symbolic link.
- If the file type is **symbolic link**, and the target file does not exist, the system will return the 404 Not Found error. Error code: SymlinkTargetNotExist.
- If the file type is **symbolic link**, and the target file type is symbolic link, the system will return the 400 Bad Request error. Error code: InvalidTargetType.

Example

Request example:

```
HEAD /oss.jpg HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 07:32:52 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JbzF2LxZUtanlJ5dLA092wpDC/E=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
x-oss-object-type: Normal
Date: Fri, 24 Feb 2012 07:32:52 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 344606
Content-Type: image/jpg
Connection: keep-alive
Server: AliyunOSS
```

Get Object Meta

Get Object Meta is used to obtain the basic meta information of an object in a bucket, but it does not return the content. The meta information includes the Etag, Size (the file size), and LastModified.

Request syntax

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

Detail analysis

- After the Get Object Meta request is sent, no message body is returned no matter whether the system returns the OK message or an error message.
- Get Object Meta needs to contain the parameters of the ObjectMeta request; otherwise, it indicates a Get Object request.
- If the file does not exist, the system will return the 404 Not Found error.
- Get Object Meta is more lightweight than Header Object. Only some basic meta information of an object in a bucket is returned. The meta information includes the Etag, Size (the file size), and LastModified. The Size is measured with the value of the Content-Length header.
- If the file type is **symbolic link**, only the information of the symbolic link itself will be returned.

Example

Request example:

```
GET /oss.jpg?objectMeta HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0FmgbrQ0=
```

Return example

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
ETag: "5B3C1A2E053D763E1B002CC607C5A0FE"
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 344606
Connection: keep-alive
Server: AliyunOSS
```

Put Object ACL

The Put Object ACL interface is used to modify the access permission of an object. Currently, an object may have three types of access permissions: private, public-read, and public-read-write. You can use the “x-oss-object-acl” header in the Put Object ACL request to set the access permission. Only the bucket owner has the permission to perform this operation. If the operation succeeds, 200 will be returned; otherwise, the corresponding error code and prompt message will be returned.

Request syntax:

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

Definition of Object ACL

Name	Description
private	This ACL indicates that an object is a private resource. Only the owner of this object has the permission to read or write this object.
public-read	This ACL indicates that an object is a resource that can be read by the public. Only the owner of this object has the permission to read and write this object. Other users only have the permission to read this object.
public-read-write	This ACL indicates that an object is a resource that can be read and written by the public. All users have the permission to read and write this object.
default	This ACL indicates an object is a resource inheriting the read-write permissions of the bucket. That is, the bucket and the object have the same permissions.

Detail analysis:

- Read operations to an object include: the read operations to the source object in GetObject, HeadObject, CopyObject and UploadPartCopy. Write operations to an object include: the write operations to a new object in PutObject, PostObject, AppendObject, DeleteObject, DeleteMultipleObjects, CompleteMultipartUpload and CopyObject.
- The x-oss-object-acl must be set to one of the preceding three permissions. Otherwise, the OSS will return the 400 Bad Request message and the error code is: InvalidArgument.
- You can use PutObject ACL to set the ACL of an object. In addition, when writing an object, you can include x-oss-object-acl in the request header to set the ACL of the object. The effect is equivalent to PutObject ACL. For example, if the header of the PutObject request carries x-oss-object-acl, you can set the ACL of an object while uploading the object.
- When a user who has no permission to read an object reads the object, OSS will return the 403 Forbidden message and the error code is: AccessDenied. The message displayed is: You do not have read permission on this object.

- When a user who has no permission to write an object writes the object, OSS will return the 403 Forbidden message and the error code is: AccessDenied. The message displayed is: You do not have write permission on this object.
- Only the owner of a bucket has the permission to call the PutObject ACL to modify the ACL for an object in this bucket. When a non-bucket owner calls the PutObject ACL, OSS will return the 403 Forbidden message and the error code is: AccessDenied. The message displayed is: You do not have write acl permission on this object.
- The object ACL takes precedence over the bucket ACL. For example, if the bucket ACL is private and the object ACL is public-read-write, the system first checks the ACL of the object when a user accesses the object. As a result, all users can access this object even if the bucket is a private bucket. If the ACL of an object has never been set, the ACL of this object is the same as that of the bucket where the object is located.

Example

Request example:

```
PUT /test-object?acl HTTP/1.1
x-oss-object-acl: public-read
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 0
Connection: keep-alive
Server: AliyunOSS
```

Get Object ACL

The Get Object ACL operation is used to obtain the permission to access an object in a bucket.

Request syntax

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

```
Authorization: SignatureValue
```

Response elements

Name	Description
AccessControlList	Container used for storing the ACL information Type: container Parent node: AccessControlPolicy
AccessControlPolicy	Specify the container that stores the Get Object ACL result. Type: container Parent node: none
DisplayName	Name of the bucket owner. (Consistent with the ID at present) Type: string Parent node: AccessControlPolicy.Owner
Grant	Specify the ACL permission of an object. Type: enumerative string Valid values: private，public-read，public-read-write Parent node: AccessControlPolicy.AccessControlList
ID	User ID of the bucket owner Type: string Parent node: AccessControlPolicy.Owner
Owner	Container used for saving the information about the bucket owner. Type: container Parent node: AccessControlPolicy

Detail analysis

- Only the bucket owner can use Get Object ACL to obtain the ACL of an object in the bucket. If you are not the bucket owner and send a Get Object ACL request, the system will return the 403 Forbidden message. Error code: AccessDenied. The message displayed is: You do not have read acl permission on this object.
- If a Get Object ACL request is sent but the ACL has never been set for the object, ObjectACL returned by the OSS is default, indicating that the ACL of this object is the same as the bucket ACL. That is, if the access permission of the bucket is private, the access permission of this object is also private; if the access permission of the bucket is public-read-write, the access permission of this object is also public-read-write.

Example

Request example:

```
GET /test-object?acl HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0FmgbrQ0=
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 559CC9BDC755F95A64485981
Date: Wed, 29 Apr 2015 05:21:12 GMT
Content-Length: 253
Content-Tupe: application/xml
Connection: keep-alive
Server: AliyunOSS

<?xml version="1.0" ?>
<AccessControlPolicy>
<Owner>
<ID>00220120222</ID>
<DisplayName>00220120222</DisplayName>
</Owner>
<AccessControlList>
<Grant>public-read </Grant>
</AccessControlList>
</AccessControlPolicy>
```

Post Object

The Post Object operation is used to upload a file to a specified bucket using the HTML form. As a substitute of Put Object, Post Object makes it possible to upload files to a bucket based on the browser. The message body of Post Object is encoded using multipart/form-data. In the Put Object operation, parameters are transferred through the HTTP request header. In the Post Object operation, parameters are transferred as the form fields in the message body.

Post Object

Request syntax

```
POST / HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
User-Agent: browser_data
Content-Length：ContentLength
```

```
Content-Type: multipart/form-data; boundary=9431149156168

--9431149156168
Content-Disposition: form-data; name="key"

key
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

success_redirect
--9431149156168
Content-Disposition: form-data; name="Content-Disposition"

attachment;filename=oss_download.jpg
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-uuid"

myuuid
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-tag"

mytag
--9431149156168
Content-Disposition: form-data; name="OSSAccessKeyId"

access-key-id
--9431149156168
Content-Disposition: form-data; name="policy"

encoded_policy
--9431149156168
Content-Disposition: form-data; name="Signature"

signature
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg

file_content
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to OSS
--9431149156168--
```

Form fields

Name	Description	Required or Optional
OSSAccessKeyId	Specify the access key ID of the bucket owner. Type: string Default value: none Restriction: This form field is required when the bucket does not allow public-read-	Conditional

	write, or when the Policy (or Signature) form field is provided.	
policy	Specify validity of the form fields in the request. A request that does not contain the Policy form field is treated as an anonymous request, and can only access buckets that allow public-read-write. For details, refer to 5.7.4.1 Post Policy. Type: string Default value: none Restriction: This form field is required when the bucket does not allow public-read-write, or when the OSSAccessKeyId (or Signature) form field is provided.	Conditional
Signature	Specify the signature information that is computed based on the Access Key Secret and Policy. The OSS checks the signature information to verify validity of the Post Object request. For details, refer to 5.7.4.2 Post Signature. Type: string Default value: none Restriction: This form field is required when the bucket does not allow public-read-write, or when the OSSAccessKeyId (or Policy) form field is provided.	Conditional
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	REST request headers. For details, refer to the related descriptions in Put Object. Type: string Default value: none	Optional
file	Specify the file or text content. It must be the last field in the form. The browser automatically sets Content-Type based on the file type, and overwrites the user setting. The OSS can only upload one file at a time. Type: string Default value: none	Required
key	Specify the object name of	Required

	the uploaded file. If the name of the uploaded file needs to be used as the object name, use the ${filename} variable. For example, if the user uploads the b.jpg file and the Key field is set to /user/a/${filename}, the final object name is /user/a/b.jpg. If the file name contains the path, remove the path from the file name. For example, if the user uploads the a/b/c/b.jpg file, use the file name b.jpg. If the Key field is set to /user/a/${filename}, the final object name is /user/a/b.jpg. Type: string Default value: none	
success_action_redirect	Specify the URL to which the client is redirected after successful upload. If this form field is not specified, the returned result is specified by success_action_status. If upload fails, the OSS will return an error code, and the client will not be redirected to any URL. Type: string Default value: none	Optional
success_action_status	Specify the status code returned to the client after the previous successful upload if success_action_redirect is not specified. Valid values include 200, 201, and 204 (default). If this field is set to 200 or 204, the OSS returns an empty file and a corresponding status code. If this field is set to 201, the OSS returns an XML file and the 201 status code. If this field is not specified or set to an invalid value, the OSS returns an empty file and the 204 status code. Type: string Default value: none	Optional
x-oss-meta-*	Specify the user meta value set by the user. The OSS	Optional

	does not check or use this value. Type: string Default value: none	
x-oss-server-side-encryption	Specify the server-side encryption algorithm when the OSS creates an object. Type: string Valid value: AES256	Optional
x-oss-object-acl	Specify the access permission when the OSS creates an object. Type: string Valid values: public-read, private, and public-read-write	Optional
x-oss-security-token	If STS temporary authorization is used for this access, you need to specify the item to be the SecurityToken value. At the same time, OSSAccessKeyId should use a paired temporary AccessKeyId. The signature calculation is consistent with the general AccessKeyId signature. Type: string Default value: none	Optional

Response header

Name	Description
x-oss-server-side-encryption	If x-oss-server-side-encryption is specified in the request, the response contains this header, which indicates the encryption algorithm used. Type: string

Response elements

Name	Description
PostResponse	Specify the container that saves the result of the Post Object request. Type: container Sub-nodes: Bucket, ETag, Key and Location
Bucket	Specify the bucket name. Type: string Parent node: PostResponse
ETag	Specify the entity tag (ETag) that is created

	when an object is generated. For an object created by Post Object, the ETag value is the UUID of the object, and can be used to check whether the content of the object has changed. Type: string Parent node: PostResponse
Location	Specify the URL of the newly created object. Type: string Parent node: PostResponse

Detail analysis

- To perform the Post Object operation, you must have the permission to write the bucket. If the bucket allows public-read-write, you can choose not to upload the signature information; otherwise, signature verification must be performed on the Post Object operation. Unlike Put Object, Post Object uses AccessKeySecret to compute the signature for the policy. The computed signature string is used as the value of the Signature form field. The OSS checks this value to verify validity of the signature.
- No matter whether the bucket allows public-read-write, once any one of the OSSAccessKeyId, Policy, and Signature form fields is uploaded, the remaining two form fields are required. If the remaining two form fields are missing, the OSS will return the error code: InvalidArgument.
- Form encoding submitted by the Post Object operation must be “multipart/form-data”. That is, Content-Type in the header must be in the multipart/form-data; boundary=xxxxxx format, where boundary is the boundary string.
- The URL of the submitted form can be the domain name of the bucket. It is not necessary to specify the object in the URL. That is, the request line is POST / HTTP/1.1, and cannot be written as POST /ObjectName HTTP/1.1.
- The policy specifies the valid values of form fields in the Post Object request. The OSS checks validity of the request based on the policy. If the request is invalid, the OSS will return the error code: AccessDenied. When checking validity of the policy, the OSS does not check irrelevant form fields in the policy.
- The form and policy must be encoded with UTF-8. The policy is a JSON text encoded with UTF-8 and Base64.
- The Post Object request can contain extra form fields. The OSS checks validity of these form fields based on the policy.
- If you have uploaded the Content-MD5 request header, the OSS will calculate the body’s Content-MD5 and check if the two are consistent. If the two are different, the error code InvalidDigest will be returned.
- If the Post Object request contains the Header signature or URL signature, the OSS does not check these signatures.
- If the Put Object request carries a form field prefixed with x-oss-meta-,the form field is treated as the user meta, for example, x-oss-meta-location. A single object can have multiple

similar parameters, but the total size of all user meta cannot exceed 8 KB.
- The total length of the body in the Post Object request cannot exceed 5 GB. When the file length is too large, the system will return the error code: EntityTooLarge.
- If the x-oss-server-side-encryption header is specified when you upload an object, the value of this header must be set to AES256. Otherwise, the system will return 400 and the error code: InvalidEncryptionAlgorithmError. After this header is specified, the response header will also contain this header, and the OSS stores the encryption algorithm of the uploaded object. When this object is downloaded, the response header will contain x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this object.
- Form fields are not case-sensitive, but their values are case-sensitive.

Example

Request example:

```
POST / HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 344606
Content-Type: multipart/form-data; boundary=9431149156168

--9431149156168
Content-Disposition: form-data; name="key"

/user/a/${filename}
--9431149156168
Content-Disposition: form-data; name="success_action_status"

200
--9431149156168
Content-Disposition: form-data; name="Content-Disposition"

content_disposition
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-uuid"

uuid
--9431149156168
Content-Disposition: form-data; name="x-oss-meta-tag"

metadata
--9431149156168
Content-Disposition: form-data; name="OSSAccessKeyId"

44CF9590006BF252F707
--9431149156168
Content-Disposition: form-data; name="policy"

eyJleHBpcmF0aW9uIjoiMjAxMy0xMi0wMVQxMjowMDowMFoiLCJjb25kaXRpb25zIjpbWyJjb250ZW50LWxlbmd0aC1
yYW5nZSIsIDAsIDEwNDg1NzYwXSx7ImJ1Y2tldCI6ImFoYWhhIn0sIHsiQSI6ICJhIn0seyJrZXkiOiAiQUJDIn1dfQ==
--9431149156168
Content-Disposition: form-data; name="Signature"
```

```
kZoYNv66bsmc10+dcGKw5x2PRrk=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.txt"
Content-Type: text/plain

abcdefg
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to OSS
--9431149156168--
```

Response example:

```
HTTP/1.1 200 OK
x-oss-request-id: 61d2042d-1b68-6708-5906-33d81921362e
Date: Fri, 24 Feb 2014 06:03:28 GMT
ETag: 5B3C1A2E053D763E1B002CC607C5A0FE
Connection: keep-alive
Content-Length: 0
Server AliyunOSS
```

Post Policy

The policy form field requested by POST is used to verify the validity of the request. The policy is a JSON text encoded with UTF-8 and Base64. It states the conditions that a Post Object request must meet. Although the post form field is optional for uploading public-read-write buckets, we strongly suggest using this field to limit POST requests.

Policy example

```
{ "expiration": "2014-12-01T12:00:00.000Z",
"conditions": [
{"bucket": "johnsmith" },
["starts-with", "$key", "user/eric/"]
]
}
```

In the Post Object request, the policy must contain expiration and conditions.

Expiration

Expiration specifies the expiration time of the policy, which is expressed in ISO8601 GMT. For example, “2014-12-01T12:00:00.000Z” means that the Post Object request must be sent before 12:00 on December 1, 2014.

Conditions

Conditions is a list that specifies the valid values of form fields in the Post Object request.Note: The value of a form field is extended after the OSS checks the policy. Therefore, the valid value of the form field set in the policy is equivalent to the value of the form field before extension. For example, if the key form field is set to user/user1/${filename} and the file name of the user is a.txt, the policy form field in the Post Object request must be set to [“eq” ,” $key” ,” user/user1/\${filename}”], instead of [“eq” ,” $key” ,” $key” ,” user/user1/a.txt].The following table lists the conditions supported by the policy:

Name	Description
content-length-range	Specify the acceptable maximum and minimum sizes of the uploaded file. This condition supports the content-length-range match mode.
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	HTTP request headers. This condition supports the exact match and starts-with match modes.
key	Specify the object name of the uploaded file. This condition supports the exact match and starts-with match modes.
success_action_redirect	Specify the URL to which the client is redirected after successful upload. This condition supports the exact match and starts-with match modes.
success_action_status	Specify the status code returned after successful upload if success_action_redirect is not specified. This condition supports the exact match and starts-with match modes.
x-oss-meta-*	Specify the user meta set by the user. This condition supports the exact match and starts-with match modes.

If the Post Object request contains other form fields, these extra form fields can be added to Conditions of the policy. The OSS does not check validity of the form fields that are not contained in the conditions.

Condition match modes

Conditions match modes	Description
Exact match	The value of a form field must be exactly the same as the value declared in the conditions. For example, if the value of the key form field must be a, the conditions must be: { “key” : “a” }, or: [“eq” , “$key” , “a”]
Starts With	The value of a form field must start with the

	specified value. For example, if the value of key must start with /user/user1, the conditions must be: [“starts-with”, “$key”, “/user/user1”]
Specified file size	Specify the maximum and minimum sizes of the files that can be uploaded. For example, if the acceptable file size is 1–10 bytes, the conditions must be: [“content-length-range”, 1, 10]

Escape characters

In the policy form field of the Post Object request, $ is used to indicate a variable. Therefore, to describe $, the escape character must be used. In addition, some characters in JSON strings are escaped. The following chart describes characters in the JSON string of the policy form field of a Post Object request.

Escape characters	Description
\/	Slash
\	Backslash
\”	Double quotation marks
\$	Dollar sign
\b	Space
\f	Form feed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\uxxxx	Unicode character

Post Signature

For a verified Post Object request, the HTML form must contain policy and signature. Policy specifies which values are acceptable in the request. The procedure for computing signature is as follows:

1. Create a UTF-8 encoded policy.
2. Encode the policy with Base64. The encoding result is the value of the policy form field, and this value is used as the string to be signed.
3. Use AccessKeySecret to sign the string. The signing method is the same as the computing method of the signature in the Header, that is, replacing the string to be signed with the policy form field.

Demo sample

- Demo of passing parameters from the web form field to the OSS: Click here

Callback

To perform a callback, you simply need to attach the relevant callback parameters to the request sent to OSS. APIs that currently support callbacks include PutObject, PostObject, and CompleteMultipartUpload.

Construct the callback parameter

The callback parameter is composed of a JSON string encoded in Base64. It is critical that you specify the request callback server URL (callbackUrl) and callback content (callbackBody). Detailed JSON fields are as follow:

Field	Meaning	
callbackUrl	- After a file is uploaded successfully, the OSS will send a callback request to this URL. The request method is POST and the body is the content specified for callbackBody. Under normal circumstances, if this URL needs to respond to "HTTP/1.1 200 OK", the response body must be in the JSON format and the response header Content-Length must be a valid value and not exceed 3 MB. - This function allows users to set up to 5 URLs, separated by ";". The OSS will send requests one by one until the first successful response is returned. - If no URL is configured or the value is null, it is regarded that there is no callback configuration. - HTTPS addresses are supported. - To ensure that Chinese characters is correctly processed, the callbackUrl needs to be encoded. For example,	Required

	http://example.com/Chinese.php?key=value&ChineseName=Chinese Value needs to be encoded into http://example.com/%E4%B8%AD%E6%96%87.php?key=value&%E4%B8%AD%E6%96%87%E5%90%8D%E7%A7%B0=%E4%B8%AD%E6%96%87%E5%80%BC.	
callbackHost	- The host header value for initiating callback requests. It is valid only when the callbackUrl is set. - If no callbackHost is set, the URL in callbackUrl will be resolved and the host generated after resolving will be entered in callbackHost	Optional
callbackBody	- The value of the request body when a callback is initiated, for example, key=$(key)&etag=$(etag)&my_var=$(x:my_var). - It supports OSS system variables, custom variables, and constants. The supported system variables are described in the table below. Custom variables are supported by transmission through callback-var in PutObject and CompleteMultipart. In Post Object operations, each variable is transmitted through a form field.	Required
callbackBodyType	- The Content-Type of the callback requests initiatiated. It supports application/x-www-form-urlencoded and application/json, and the former is the default value. - If the Content-Type is set to application/x-www-form-urlencoded, the variables in callbackBody will be replaced by URL encoded values. If the Content-Type is set to application/json, these variables will be replaced according to the JSON format.	Optional

JSON string examples are as follows:

```
{
"callbackUrl":"121.101.166.30/test.php",
"callbackHost":"oss-cn-hangzhou.aliyuncs.com",
"callbackBody":"{\"mimeType\":${mimeType},\"size\":${size}}",
"callbackBodyType":"application/json"
}
```

```
{
"callbackUrl":"121.43.113.8:23456/index.html",
"callbackBody":"bucket=${bucket}&object=${object}&etag=${etag}&size=${size}&mimeType=${mimeType}&imageInfo.height=${imageInfo.height}&imageInfo.width=${imageInfo.width}&imageInfo.format=${imageInfo.format}&my_var=${x:my_var}"
}
```

Here, the system variables that can be set for callbackBody include the following. In specific, the imageInfo is for the image format. It should be left empty for the non-image format:

System Variable	Meaning
bucket	bucket
object	object
etag	The file’ s etag, that is, the etag field returned to the user
size	The object size. During the CompleteMultipartUpload operation, this is the size of the whole object
mimeType	The resource type. For jpeg images, the resource type is image/jpeg
imageInfo.height	The image height
imageInfo.width	The image width
imageInfo.format	The image format, such as jpg and png

Custom parameters

You can use the callback-var parameter to configure custom parameters.

Custom parameters are a map of key-values. You can configure the required parameters to the map. When initiating a POST callback request, the OSS puts these parameters and the system parameters described in the above section in the body of the POST request, so that these parameters can be easily obtained by the callback recipient.

You can construct custom parameters in the same way as constructing the callback parameter. The custom parameters can also be transmitted in the JSON format. The JSON string is a map containing

key-values of all custom parameters. **It should be particularly noted that, the keys of the custom parameters must start with x: and be in the lower case. Otherwise, the OSS will return an error.** Assume that you need to set two custom parameters x:var1 and x:var2, and the values of the two parameters are value1 and value2 respectively, the JSON format constructed will be as follows:

```
{
"x:var1":"value1",
"x:var2":"Value 2"
}
```

Construct callback requests

After the callback and callback-var parameters are constructed, you can transmit the parameters to the OSS with three methods. The callback parameter is required, and the callback-var parameter is optional. If you configure no custom parameter, the callback-var field does not need to be added. The said three methods are as follows:

- Including parameters in the URL.
- Including parameters in the header.
- Using form fields to include parameters in the body of a POST request. **You can only use this method to specify the callback parameter when using POST to upload an object.**

The three methods are alternative; otherwise, the OSS will return an InvalidArgument error.

To include a parameter in the OSS request, first you need to use Base64 to encode the JSON string constructed above, and include the string in the OSS request using the methods described below:

- To include parameters in the URL, use ‘callback=[CallBack]’ or ‘callback-var=[CallBackVar]’ as a URL parameter to send it with the request. When CanonicalizedResource of the signature is calculated, callback or callback-var is taken into consideration as a sub-resource.
- To include parameters in the header, use ‘x-oss-callback=[CallBack]’ or ‘x-oss-callback-var=[CallBackVar]’ as a head to send it with the request. When CanonicalizedOSSHeaders of the signature is calculated, x-oss-callback-var and x-oss-callback are taken into consideration. An example is provided below:

```
PUT /test.txt HTTP/1.1
Host: callback-test.oss-test.aliyun-inc.com
Accept-ncoding: identity
Content-Length: 5
x-oss-callback-var: eyJ4Om15X3ZhciI6ImZvci1jYWxsYmFjay10ZXN0In0=
User-Agent: aliyun-sdk-python/0.4.0 (Linux/2.6.32-220.23.2.ali1089.el5.x86_64/x86_64;2.5.4)
x-oss-callback:
eyJjYWxsYmFja1VybCI6IjEyMS40My4xMTMuODoyMzQ1Ni9pbmRleC5odG1sIiwgICJjYWxsYmFja0JvZHkiOiJidWNrZXQ9JHtidWNrZXR9Jm9iamVjdD0ke29iamVjdH0mZXRhZz0ke2V0YWd9JnNpemU9JHtzaXplfSZtaW1lVHlwZT0ke21pbWVUeXBlfSZpbWFnZUluZm8uaGVpZ2h0PSR7aW1hZ2VJbmZvLmhlaWdodH0maW1hZ2VJbmZvLndpZHRoPSR7aW1hZ2VJbmZvLndpZHRofSZpbWFnZUluZm8uZm9ybWF0PSR7aW1hZ2VJbmZvLmZvcm1hdH0mbXlfdmFyPSR7eDpteV
```

```
92YXJ9In0=
Host: callback-test.oss-test.aliyun-inc.com
Expect: 100-Continue
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: text/plain
Authorization: OSS mlepou3zr4u7b14:5a74vhd4UXpmyuudV14Kaen5cY4=

Test
```

- It is slightly complicated to include the callback parameter when POST is used to upload an object, because the callback parameter needs to be included using an independent form field. See the example below:

```
--9431149156168
Content-Disposition: form-data; name="callback"

eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5waHAiLCJjYWxsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9JHt4OnRhYmxlfSIsImNhbGxiYWNrQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVybGVuY29kZWQifQ==
```

If there are custom parameters, **you cannot directly include the callback-var parameter in the form field**. Each custom parameter needs to be included using an independent form field. For example, if the JSON of a custom parameter is:

```
{
"x:var1":"value1",
"x:var2":"value2"
}
```

The form field of the POST request must be as follows:

```
--9431149156168
Content-Disposition: form-data; name="callback"

eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5waHAiLCJjYWxsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkmdGFibGU9JHt4OnRhYmxlfSIsImNhbGxiYWNrQm9keVR5cGUiOiJhcHBsaWNhdGlvbi94LXd3dy1mb3JtLXVybGVuY29kZWQifQ==

--9431149156168
Content-Disposition: form-data; name="x:var1"

value1

--9431149156168
Content-Disposition: form-data; name="x:var2"

value2
```

At the same time, you can add callback conditions in the policy (if callback is not added, upload verification will not be performed on this parameter). For example:

```
{ "expiration": "2014-12-01T12:00:00.000Z",
"conditions": [
{"bucket": "johnsmith" },
{"callback":
"eyJjYWxsYmFja1VybCI6IjEwLjEwMS4xNjYuMzA6ODA4My9jYWxsYmFjay5waHAiLCJjYWxsYmFja0hvc3QiOiIxMC4xMDEuMTY2LjMwIiwiY2FsbGJhY2tCb2R5IjoiZmlsZW5hbWU9JChmaWxlbmFtZSkiLCJjYWxsYmFja0JvZHlUeXBlIjoiYXBwbGljYXRpb24veC13d3ctZm9ybS11cmxlbmNvZGVkIn0="},
["starts-with", "$key", "user/eric/"],
]
}
```

Initiate callback requests

If the file is uploaded successfully, the OSS will use the POST method to send the specific content to the application server based on the callback parameter and the custom parameters (the callback-var parameter) in the user’ s request.

```
POST /index.html HTTP/1.0
Host: 121.43.113.8
Connection: close
Content-Length: 181
Content-Type: application/x-www-form-urlencoded
User-Agent: ehttp-client/0.0.1

bucket=callback-
test&object=test.txt&etag=D8E8FCA2DC0F896FD7CB4CB0031BA249&size=5&mimeType=text%2Fplain&imageInf
o.height=&imageInfo.width=&imageInfo.format=&x:var1=for-callback-test
```

Return callback results

For example, the application server will return the following request for response:

```
HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Mon, 14 Sep 2015 12:37:27 GMT
Content-Type: application/json
Content-Length: 9

{"a":"b"}
```

Return upload results

The following content will be sent to the client:

```
HTTP/1.1 200 OK
Date: Mon, 14 Sep 2015 12:37:27 GMT
```

```
Content-Type: application/json
Content-Length: 9
Connection: keep-alive
ETag: "D8E8FCA2DC0F896FD7CB4CB0031BA249"
Server: AliyunOSS
x-oss-bucket-version: 1442231779
x-oss-request-id: 55F6BF87207FB30F2640C548

{"a":"b"}
```

It must be noted that, in the case of requests such as CompleteMultipartUpload, there will be content in the returned request body (for example, information in XMl format). After using the upload callback function, the original body content will be overwritten, such as ‘“a”:”b”‘. Please take this into consideration for judgment and processing.

Callback signature

When the callback parameter is set, the OSS will send the POST callback request to the user’s application server based on the callbackUrl set by the user. After receiving the callback request, if you expect the application server to check whether the callback request is initiated by OSS, you can include a signature in the callback request to verify the OSS identity.

Generate signatures

The signature occurs at the OSS side, and is signed using the RSA Asymmetric Encryption. You can encrypt the signature using a private key as follows:

```
authorization = base64_encode(rsa_sign(private_key, url_decode(path) + query_string + ‘\n’ + body, md5))
```

Instructions: The private_key indicates a private key which is only known to the OSS. The path indicates the resource path of the callback request. The query_string indicates a query string. The body indicates the message body of the callback. The signature thus consists of the following steps:

- Obtain the string to be signed: The resource path URL is decoded, added by the initial query string, a carriage return and the callback message body.
- RSA signature: Use a private key to sign the desired string. The hashing function for signature is MD5.
- Use Base64 to encode the signed result to get the final signature. Put the signature in the authorization header of the callback request.

An example is provided below:

```
POST /index.php?id=1&index=2 HTTP/1.0
Host: 121.43.113.8
Connection: close
Content-Length: 18
```

```
authorization:
kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==
Content-Type: application/x-www-form-urlencoded
User-Agent: ehttp-client/0.0.1
x-oss-pub-key-url: aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9jYWxsYmFja19wdWJfa2V5X3YxLnBlbQ==

bucket=yonghu-test
```

The path is /index.php, the query_string is ?id=1&index=2, the body is bucket=yonghu-test, and the final signature result is kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA==.

Verify signature

Signature verification is an inverse process of signature. The signature is verified by the application server, and the process is as follows:

```
Result = rsa_verify(public_key, md5(url_decode(path) + query_string + '\n' + body),
base64_decode(authorization))
```

The fields have the same meanings as described during the signature process. The public_key indicates a public key. The authorization indicates the signature in the callback header. The signature verification consists of the following steps:

- The x-oss-pub-key-url header of the callback request stores the Base64-encoded URL of the public key. The header must be decoded with Base64 to obtain the public key as follows:

```
public_key = urlopen(base64_decode(Value of the x-oss-pub-key-url header))
```

It should be noted that, the value of the x-oss-pub-key-url header must start with http://gosspublic.alicdn.com/ or https://gosspublic.alicdn.com/, so as to ensure that the public key is provided by OSS.

- Obtain the Base64-decoded signature

```
signature = base64_decode(Value of the authorization header)
```

- Obtain the string to be signed in the same way as that described in the signature process.

```
sign_str = url_decode(path) + query_string + '\n' + body
```

- Verify the signature

```

```

```
result = rsa_verify(public_key, md5(sign_str), signature)
```

The above sample is used as an example:

- Obtain the URL of the public key, that is, decoding the aHR0cDovL2dvc3NwdWJsaWMuYWxpY2RuLmNvbS9jYWxsYmFja19wdWJfa2V5X3YxLnBlbQ== with Base64http://gosspublic.alicdn.com/callback_pub_key_v1.pem`
- The signature header kKQeGTRccDKyHB3H9vF+xYMSrmhMZjzzl2/kdD1ktNVgbWEfYTQG0G2SU/RaHBovRCE8OkQDjC3uG33esH2txA== is decoded with Base64 (The decoded result cannot be displayed because it is a nonprintable string).
- Obtain the string to be singed, that is, url_decode (“index.php”) +” ?id=1&index=2” + “\n” + “bucket=yonghu-test” . Then perform the MD5 check.
- Verify the signature

Application server example

Python is used as an example to demonstrate how an application server verifies a signature. In this example, the M2Crypto library needs to be installed.

```
import httplib
import base64
import md5
import urllib2
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
from M2Crypto import RSA
from M2Crypto import BIO

def get_local_ip():
try:
csock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
csock.connect(('8.8.8.8', 80))
(addr, port) = csock.getsockname()
csock.close()
return addr
except socket.error:
return ""

class MyHTTPRequestHandler(BaseHTTPRequestHandler):
'''
def log_message(self, format, *args):
return
'''

def do_POST(self):
#get public key
pub_key_url = ''
try:
pub_key_url_base64 = self.headers['x-oss-pub-key-url']
pub_key_url = pub_key_url_base64.decode('base64')
if not pub_key_url.startswith("http://gosspublic.alicdn.com/") and not
```

```
pub_key_url.startswith("https://gosspublic.alicdn.com/"):
self.send_response(400)
self.end_headers()
return

url_reader = urllib2.urlopen(pub_key_url)
#you can cache it
pub_key = url_reader.read()
except:
print 'pub_key_url : ' + pub_key_url
print 'Get pub key failed!'
self.send_response(400)
self.end_headers()
return

#get authorization
authorization_base64 = self.headers['authorization']
authorization = authorization_base64.decode('base64')

#get callback body
content_length = self.headers['content-length']
callback_body = self.rfile.read(int(content_length))

#compose authorization string
auth_str = ''
pos = self.path.find('?')
if -1 == pos:
auth_str = urllib2.unquote(self.path) + '\n' + callback_body
else:
auth_str = urllib2.unquote(self.path[0:pos]) + self.path[pos:] + '\n' + callback_body
print auth_str

#verify authorization
auth_md5 = md5.new(auth_str).digest()
bio = BIO.MemoryBuffer(pub_key)
rsa_pub = RSA.load_pub_key_bio(bio)
try:
result = rsa_pub.verify(auth_md5, authorization, 'md5')
except:
result = False

if not result:
print 'Authorization verify failed!'
print 'Public key : %s' % (pub_key)
print 'Auth string : %s' % (auth_str)
self.send_response(400)
self.end_headers()
return

#do something accoding to callback_body

#response to OSS
resp_body = '{"Status":"OK"}'
self.send_response(200)
self.send_header('Content-Type', 'application/json')
self.send_header('Content-Length', str(len(resp_body)))
```

```
self.end_headers()
self.wfile.write(resp_body)

class MyHTTPServer(HTTPServer):
def __init__(self, host, port):
HTTPServer.__init__(self, (host, port), MyHTTPRequestHandler)


if '__main__' == __name__:
server_ip = get_local_ip()
server_port = 23451
server = MyHTTPServer(server_ip, server_port)
server.serve_forever()
```

Application servers implemented in other languages are as follows:

Java version:

- Download address: Click here
- Running method: Extract the package and run java -jar oss-callback-server-demo.jar 9000 (9000 is the port number and can be designated as needed)

PHP version:

- Download address: Click here
- Running method: Deploy the program to an Apache environment. The characteristics of the PHP language determine that the environment is depended on to retrieve some headers. You may refer to the example to make modifications to your own environment.

Python version:

- Download address: Click here
- Running method: Extract the package and directly run python callback_app_server.py. You need to install RSA dependencies to run this program.

Ruby version:

- Download address: Click here
- Running method: ruby aliyun_oss_callback_server.rb

Special instructions

If the input callback parameter or callback-var parameter is invalid, a 400 error will be returned, with the error code of “InvalidArgument”. Invalid situations include the following:

- In the PutObject() and CompleteMultipartUpload() interfaces, the callback(x-oss-callback) or callback-var(x-oss-callback-var) parameters are input at the same time to the URL and header fields

- The callback or callback-var parameter is too long (over 5KB). PostObject() is not subject to this restriction because there is no callback-var parameter, and this is true for the following as well.
- Callback or callback-var is not Base64 encoded
- After Base64 decoding, the callback or callback-var parameter is not in a valid JSON format
- After callback parameter resolution, the callbackUrl field contains more than 5 URLs, or the input port in the URL is invalid, such as{"callbackUrl":"10.101.166.30:test", "callbackBody":"test"}
- After callback parameter resolution, the callbackBody field is blank
- After callback parameter resolution, the callbackBodyType field value is not “application/x-www-form-urlencoded” or “application/json”
- After callback parameter resolution, the callbackBody field contains invalid formats of variables. The valid format is ${var}
- After callback-var parameter resolution, the format is not the expected JSON format. The expected format is:{"x:var1":"value1","x:var2":"value2"...}

If a callback fails, the system will return a 203 error, with the error code “CallbackFailed”. A callback failure only indicates that the OSS did not receive the expected callback response (for example, the response from the application server was not in the JSON format), not that the application server did not receive the callback request. In addition, by this time, the file has been successfully uploaded to the OSS.

The response returned by the application server to the OSS must contain the Content-Length header, and the size of the body cannot exceed 1 MB.

Put Symlink

The Put Symlink operation is used to create a symbolic link.

Request syntax

```
PUT /ObjectName?symlink HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: SignatureValue
x-oss-symlink-target: TargetObjectName
```

Request header

Name	Description

x-oss-symlink-target	The target file that the symbolic link points to. Type: string Valid value: The naming rule is the same as the object.

Detail analysis

- Like the ObjectName, the TargetObjectName needs an encoded URL.
- The target file type of the symbolic link cannot be a symbolic link.
- When a symbolic link is created,
  - The target file availability is not checked
  - The target file type validity is not checked
  - Whether the target file has the access permission is not checked

  The above checks will be postponed to the APIs that require access to the target file such as the GetObject operation.
- If the file to be added already exists, and you are authorized to access this object, the newly-added file will overwrite the existing file, and the system will return the 200 OK message.
- If the PutSymlink request carries a parameter prefixed with x-oss-meta-,the parameter is treated as user meta, for example, x-oss-meta-location. A single object can have multiple similar parameters, but the total size of all user meta cannot exceed 8 KB.

Example

Request example:

```
PUT /link-to-oss?symlink HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Cache-control: no-cache
Content-Disposition: attachment;filename=oss_download.jpg
Date: Tue, 08 Nov 2016 02:00:25 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:kZoYNv66bsmc10+dcGKw5x2PRrk=
x-oss-symlink-target: oss.jpg
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Tue, 08 Nov 2016 02:00:25 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 582131B9109F4EE66CDE56A5
ETag: "0A477B89B4602AA8DECB8E19BFD447B6"
```

Get Symlink

The Get Symlink operation is used to obtain a symbolic link which you must have the permission to read.

Request syntax

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

Response header

Name	Description
x-oss-symlink-target	The target file that the symbolic link points to. Type: string

Detail analysis

1. If the symbolic link does not exist, the system returns the 404 Not Found error. Error code: NoSuchKey.

Example

Request example:

```
GET /link-to-oss.jpg?symlink HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 06:38:30 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:UNQDb7GapEgJCZkcde6OhZ9Jfe8=
```

Response example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Date: Fri, 24 Feb 2012 06:38:30 GMT
Last-Modified: Fri, 24 Feb 2012 06:07:48 GMT
Content-Length: 0
Connection: keep-alive
x-oss-request-id: 5650BD72207FB30443962F9A
x-oss-symlink-target: oss.jpg
```

```
ETag: "A797938C31D59EDD08D86188F6D5B872"
```

Multipart Upload Operations

Introduction

In addition to the PUT Object interface, the OSS also provides the Multipart Upload mode for you to upload files. You can apply the Multipart Upload mode in the following scenarios (but not limited to the following):

- Breakpoint upload need to be supported.
- The files to be uploaded are larger than 100 MB.
- The network conditions are poor, and the connection with the OSS server is frequently broken.
- Before a file is uploaded, the size of the file cannot be determined.

Initiate Multipart Upload

Before transmitting data in the Multipart Upload mode, you must call the Initiate Multipart Upload interface to notify the OSS to initiate a Multipart Upload event. The Initiate Multipart Upload interface returns a globally unique Upload ID created by the OSS server to identify this Multipart Upload event. You can initiate operations based on this ID, such as aborting Multipart Upload and querying Multipart Upload.

Request syntax

```
POST /ObjectName?uploads HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT date
Authorization: SignatureValue
```

Request parameters

During the Initiate Multipart Upload operation, you can use encoding-type to encode the Key in the

returned result.

Name	Description
encoding-type	Specify the encoding type of the Key in the returned result. Currently, the URL encoding is supported. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Data type: string Default value: none. Optional value: url

Request header

Name	Description
Cache-Control	Specify the web page caching behavior when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Disposition	Specify the name of the object when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Encoding	Specify the content encoding format when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Expires	Specify the expiration time in milliseconds. For details, refer to RFC2616. Type: integer Default value: none
x-oss-server-side-encryption	Specify the server-side encryption algorithm used to upload each part of this object. The OSS stores each uploaded part based on server-side encryption. Type: string Valid value: AES256

Response elements

Name	Description

Bucket	Name of a bucket for which a Multipart Upload event is initiated. Type: string Parent node: InitiateMultipartUploadResult
InitiateMultipartUploadResult	The container that saves the result of the Initiate Multipart Upload request. Type: container Sub-nodes: Bucket, Key, UploadId Parent node: none
Key	Name of an object for which a Multipart Upload event is initiated. Type: string Parent node: InitiateMultipartUploadResult
UploadId	Unique ID of a Multipart Upload event. Type: string Parent node: InitiateMultipartUploadResult
EncodingType	Specify the encoding type for the returned results. If encoding-type is specified in the request, the Key will be encoded in the returned result. Type: string Parent node: Container

Detail analysis

- When using this operation to calculate the authentication signature, you need to add "?uploads" to "CanonicalizedResource".
- The Initiate Multipart Upload request supports the following standard HTTP request headers: Cache-Control, Content-Disposition, Content-Encoding, Content-Type, Expires, and custom headers starting with x-oss-meta-. For the specific meanings of these headers, refer to the PUT Object interface.
- The Initiate Multipart Upload request does not affect an existing object with the same name.
- When receiving an Initiate Multipart Upload request, the server returns a request body in XML format. The request body has three elements: Bucket, Key, and UploadID. Please record the UploadID for subsequent Multipart operations.
- If the x-oss-server-side-encryption header is set in the Initiate Multipart Upload request, the server will return this header in the response header. During the upload of each part, the server will automatically store the part based on entropy encryption. Currently, the OSS server only supports the 256-bit advanced encryption standard (AES256). If values of other standards are specified, the OSS server returns the error code 400 and the error message InvalidEncryptionAlgorithmError. When uploading each part, you do not need to add the x-oss-server-side-encryption request header. If this request header is specified, the OSS returns the error code 400 and the error message InvalidArgument.

Example

Request example:

```
POST /multipart.data?uploads HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:/cluRFtRwMTZpC2hTj4F67AGdM4=
```

Response example:

```
HTTP/1.1 200 OK
Content-Length: 230
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 42c25703-7503-fbd8-670a-bda01eaec618
Date: Wed, 22 Feb 2012 08:32:21 GMT
Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8"?>
<InitiateMultipartUploadResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Bucket> multipart_upload</Bucket>
<Key>multipart.data</Key>
<UploadId>0004B9894A22E5B1888A1E29F8236E2D</UploadId>
</InitiateMultipartUploadResult>
```

Upload Part

After initiating a Multipart Upload event, you can upload data in parts based on the specified object name and Upload ID. Each uploaded part has a part number ranging from 1 to 10,000. For the same Upload ID, this part number identifies not only this part of data but also the location of this part in the entire file. If you upload new data using the same part number, the OSS will overwrite the existing data identified by this part number. Except the last part, the minimum size of other parts is 100 KB. There are no restrictions on the size of the last part.

Request syntax

```
PUT /ObjectName?partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
```

Detail analysis

- Before calling the Initiate Multipart Upload interface to upload a part of data, you must call this interface to obtain an Upload ID issued by the OSS server.
- In the Multipart Upload mode, except the last part, all other parts must be larger than 100 KB. However, the Upload Part interface does not immediately verify the size of the uploaded part (because it does not know whether the part is the last one). It verifies the size of the uploaded part only when Multipart Upload is completed.
- OSS will put the MD5 value of the part data received by the server in the ETag header and return it to the user.
- The part number ranges from 1 to 10,000. If the part number exceeds this range, the OSS will return the InvalidArgument error code.
- If the x-oss-server-side-encryption request header is specified when the Initiate Multipart Upload interface is called, the OSS will encrypt the uploaded part and return the x-oss-server-side-encryption header in the Upload Part response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm used for this part. For details, refer to the Initiate Multipart Upload interface.
- In order to ensure that the data transmitted over the network is free from errors, the user includes Content-MD5 in the request. The OSS will calculate the MD5 value for the uploaded data and compare it with the MD5 value uploaded by the user. If they are inconsistent, the OSS will return the InvalidDigest error code.

Example

Request example:

```
PUT /multipart.data?partNumber=1&uploadId=0004B9895DBBB6EC98E36  HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length：6291456
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:J/lICfXEvPmmSW86bBAfMmUmWjI=

[6291456 bytes data]
```

Return example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
ETag: 7265F4D211B56873A381D321F586E4A9
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0c21a
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

Upload Part Copy

Using Upload Part Copy, you can copy data from an existing object to upload a part of the data. You can add an x-oss-copy-source header in the Upload Part request to call the Upload Part Copy interface. When copying a file larger than 1 GB, you must use the Upload Part Copy method. If you want to copy a file smaller than 1 GB, you can refer to Copy Object.

Request syntax

```
PUT /ObjectName? partNumber=PartNumber&uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: SignatureValue
x-oss-copy-source: /SourceBucketName/SourceObjectName
x-oss-copy-source-range:bytes=first-last
```

Request header

Except the common request header, other headers as below in the Upload Part Copy request are used to specify the address of the copied source object and copying range.

Name	Description
x-oss-copy-source	Specify the copy source address (the requester must have the permission to read the source object) Type: string Default value: none
x-oss-copy-source-range	Copying range of the copied source object. For example, if Range is set to bytes=0-9, the system will transfer byte 0 to byte 9. This request header is not required when the entire source object is copied. Type: integer Default value: none

The following request header is used for the source objects specified by x-oss-copy-source.

Name	Description
x-oss-copy-source-if-match	If the ETAG value of the source object is equal to the ETAG value provided by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none

x-oss-copy-source-if-none-match	If the source object has not been modified since the time specified by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-unmodified-since	If the time specified by the input parameter is the same as or later than the actual modification time of the file, the system will transfer the file normally, and return the 200 OK message; otherwise, the system will return the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-modified-since	If the source object has been modified after the time specified by the user, the system will perform a copy operation. Otherwise, the system will return the 412 HTTP error code (indicating preprocessing failure). Type: string Default value: none

Response elements

Name	Description
x-oss-copy-source-if-match	If the ETAG value of the source object is equal to the ETAG value provided by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-none-match	If the source object has not been modified since the time specified by the user, the system performs the Copy Object operation; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-unmodified-since	If the time specified by the input parameter is the same as or later than the actual modification time of the file, the system will transfer the file normally, and return the 200 OK message; otherwise, the system will return the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-modified-since	If the source object has been modified after the time specified by the user, the system will perform a copy operation. Otherwise, the

	system will return the 412 HTTP error code (indicating preprocessing failure). Type: string Default value: none

Detail analysis

- Before calling the Initiate Multipart Upload interface to upload a part of data, you must call this interface to obtain an Upload ID issued by the OSS server.
- In the Multipart Upload mode, except the last part, all other parts must be larger than 100 KB. However, the Upload Part interface does not immediately verify the size of the uploaded part (because it does not know whether the part is the last one). It verifies the size of the uploaded part only when Multipart Upload is completed.
- If the x-oss-copy-source-range request header is not specified, the entire source object is copied. If this request header is specified, the returned message will contain the length of the entire file and the copying range. For example: Content-Range: bytes 0-9/44 indicates that the length of the entire file is 44 and the copying range is 0-9. If the specified range does not conform to the range rules, the OSS copies the entire file and does not contain Content-Range in the result.
- If the x-oss-server-side-encryption request header is specified when the Initiate Multipart Upload interface is called, the OSS will encrypt the uploaded part and return the x-oss-server-side-encryption header in the Upload Part response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm used for this part. For details, refer to the Initiate Multipart Upload interface.
- This operation cannot copy objects that are generated through the appended upload method.

Example

Request example:

```
PUT /multipart.data?partNumber=1&uploadId=0004B9895DBBB6EC98E36  HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length：6291456
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:J/lICfXEvPmmSW86bBAfMmUmWjI=
x-oss-copy-source: /oss-example/ src-object
x-oss-copy-source-range:bytes=100-6291756
```

Return example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Connection: keep-alive
```

```
x-oss-request-id: 3e6aba62-1eae-d246-6118-8ff42cd0c21a
Date: Thu, 17 Jul 2014 06:27:54 GMT'

<?xml version="1.0" encoding="UTF-8"?>
<CopyPartResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<LastModified>2014-07-17T06:27:54.000Z </LastModified>
<ETag>"5B3C1A2E053D763E1B002CC607C5A0FE"</ETag>
</CopyPartResult>
```

Complete Multipart Upload

After uploading all data parts, you must call the Complete Multipart Upload API to complete Multipart Upload for the entire file. During this operation, you must provide the list (including the part number and ETags) of all valid data parts. After receiving the part list you have submitted, the OSS will verify the validity of each data part individually. After all the data parts have been verified, the OSS will combine these parts into a complete object.

Request syntax

```
POST /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Content-Length: Size
Authorization: Signature

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

Request parameters

During the Complete Multipart Upload operation, you can use encoding-type to encode the Key in the returned result.

Name	Description
encoding-type	Specify the encoding type of the Key in the returned result. Currently, the URL encoding is supported. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In

	case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Data type: string Default value: none. Optional value: url

Request elements

Name	Description
CompleteMultipartUpload	Container used for storing the content of the Complete Multipart Upload request. Type: container Sub-node: one or more part elements Parent node: none
ETag	ETag value returned by the OSS after data parts are successfully uploaded. Type: string Parent node: Part
Part	The container that saves uploaded data parts. Type: container Sub-nodes: ETag, PartNumber Parent node: InitiateMultipartUploadResult
PartNumber	Number of parts. Type: integer Parent node: Part

Response elements

Name	Description
Bucket	Specify the bucket name. Type: string Parent node: CompleteMultipartUploadResult
CompleteMultipartUploadResult	The container that stores the result of the Complete Multipart Upload request. Type: container Sub-nodes: Bucket, Key, ETag, Location Parent node: none
ETag	The ETag (entity tag) is created when an object is generated and is used to indicate the content of the object. Objects created based on the Complete Multipart Upload request. The value of ETag is the UUID of the object content. The value of ETag can be used to check whether the content of the object is changed. Type: string Parent node: CompleteMultipartUploadResult

Location	Specify the URL of the newly created object. Type: string Parent node: CompleteMultipartUploadResult
Key	Name of the newly created object. Type: string Parent node: CompleteMultipartUploadResult
EncodingType	Specify the encoding type for the returned results. If encoding-type is specified in the request, the Key will be encoded in the returned result. Type: string Parent node: Container

Detail analysis

- When receiving a Complete Multipart Upload request, the OSS verifies that all parts except the last part are larger than 100 KB and check each part number and ETag in the part list submitted by the user. Therefore, when uploading data parts, the client needs to record not only the part number but also the ETag value returned by the OSS each time a part is uploaded successfully.
- It takes some time for the OSS to process the Complete Multipart Upload request. During this time, if the client is disconnected from the OSS, the OSS will continue to complete the request.
- The part numbers in the part list submitted by a user can be non-consecutive. For example, the first part number is 1 and the second part number is 5.
- After the OSS successfully processes the Complete Multipart Upload request, the corresponding Upload ID will become invalid.
- The same object may have different Upload IDs. When an Upload ID is completed, other Upload IDs of this object are not affected.
- If the x-oss-server-side-encryption request header is specified when the Initiate Multipart Upload interface is called, the OSS will return the x-oss-server-side-encryption header in the Complete Multipart Upload response header. The value of x-oss-server-side-encryption indicates the server-side encryption algorithm used for this object.
- If you have uploaded the Content-MD5 request header, the OSS will calculate the body’ s Content-MD5 and check if the two are consistent. If the two are different, the error code InvalidDigest will be returned.

Example

Request example:

```
POST /multipart.data? uploadId=0004B9B2D2F7815C432C9057C03134D4  HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 1056
```

```
Date: Fri, 24 Feb 2012 10:19:18 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:8VwFhFUWmVecK6jQlHlXMK/zMT0=

<CompleteMultipartUpload>
<Part>
<PartNumber>1</PartNumber>
<ETag>"3349DC700140D7F86A078484278075A9"</ETag>
</Part>
<Part>
<PartNumber>5</PartNumber>
<ETag>"8EFDA8BE206636A695359836FE0A0E0A"</ETag>
</Part>
<Part>
<PartNumber>8</PartNumber>
<ETag>"8C3150651671324441774 11FDA149B92"</ETag>
</Part>
</CompleteMultipartUpload>
```

Return example:

```
HTTP/1.1 200 OK
Server: AliyunOSS
Content-Length: 329
Content-Type: Application/xml
Connection: keep-alive
x-oss-request-id: 594f0751-3b1e-168f-4501-4ac71d217d6e
Date: Fri, 24 Feb 2012 10:19:18 GMT

<?xml version="1.0" encoding="UTF-8"?>
<CompleteMultipartUploadResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Location>http://oss-example.oss-cn-hangzhou.aliyuncs.com /multipart.data</Location>
<Bucket>oss-example</Bucket>
<Key>multipart.data</Key>
<ETag>B864DB6A936D376F9F8D3ED3BBE540DD-3</ETag>
</CompleteMultipartUploadResult>
```

Abort Multipart Upload

This interface can be used to abort a Multipart Upload event based on the Upload ID you provide. When a Multipart Upload event is aborted, you cannot use this Upload ID to perform any operations and the uploaded parts of data will be deleted.

Request syntax

```
DELETE /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
```

```
Authorization: Signature
```

Detail analysis

- When you abort a Multipart Upload event, parts still being uploaded will not be deleted. Therefore, if concurrent accesses exist, you need to call the Abort Multipart Upload interface several times to completely release the space of the OSS.
- If the entered Upload ID does not exist, the OSS returns an error 404 with the error code: NoSuchUpload.

Example

Request example:

```
Delete /multipart.data?&uploadId=0004B9895DBBB6EC98E  HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Wed, 22 Feb 2012 08:32:21 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:J/lICfXEvPmmSW86bBAfMmUmWjI=
```

Return example:

```
HTTP/1.1 204
Server: AliyunOSS
Connection: keep-alive
x-oss-request-id: 059a22ba-6ba9-daed-5f3a-e48027df344d
Date: Wed, 22 Feb 2012 08:32:21 GMT
```

List Multipart Uploads

The List Multipart Uploads interface can be used to list all Multipart Upload events in execution, that is, Multipart Upload events that have been initiated but not completed or aborted. The listing result returned by the OSS contains a maximum of 1000 Multipart Upload messages. If you want to specify the number of Multipart Upload messages in the listing result returned by the OSS, you can add the max-uploads parameter to the request. In addition, the IsTruncated element in the listing result returned by the OSS indicates whether there are other Multipart Upload messages.

Request syntax

```
Get /?uploads HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
```

```
Date: GMT Date
Authorization: Signature
```

Request parameters

Name	Description
delimiter	A character used to group object names. All those objects whose names contain the specified prefix and behind which the delimiter occurs for the first time act as a group of elements - CommonPrefixes. Type: string
max-uploads	Specify the maximum number of multipart upload tasks returned for one request. If this parameter is not specified, the default value 1,000 is used. The max-uploads value cannot exceed 1,000. Type: string
key-marker	Used together with the upload-id-marker parameter to specify the starting position of the returned result. l If the upload-id-marker parameter is not set, the query result includes Multipart tasks in which the lexicographic orders of all object names are greater than the value of the key-marker parameter. l If the upload-id-marker parameter is set, the query result includes Multipart tasks in which the lexicographic orders of all object names are greater than the value of the key-marker parameter and all Multipart Upload tasks in which the object names are the same as the value of the key-marker parameter but the Upload IDs are greater than the value of the upload-id-marker parameter. Type: string
prefix	Limit that the returned object key must be prefixed accordingly. Note that the keys returned from queries using a prefix will still contain the prefix. Type: string
upload-id-marker	Used together with the key-marker parameter to specify the starting position of the returned result. If the key-marker parameter is not set, the OSS ignores the upload-id-marker parameter. l If the key-marker parameter is set, the query result includes Multipart tasks in which the lexicographic orders of all object names are greater than the value of the key-marker parameter and all Multipart Upload tasks in which the object names are the same as the value of the key-marker parameter but the Upload IDs are greater than the value of the upload-id-marker parameter.

	Type: string
encoding-type	Specify the encoding of the returned content and the encoding type. Delimiter, KeyMarker, Prefix, NextKeyMarker, and Key use UTF-8 characters, but the XML 1.0 Standard does not support parsing certain control characters, such as characters with ASCII values ranging from 0 to 10. If some elements in the returned results contain control characters that are not supported by the XML 1.0 Standard, encoding-type can be specified to encode these elements, such as Delimiter, KeyMarker, Prefix, NextMarker, and Key. Data type: string Default value: none

Response elements

Name	Description
ListMultipartUploadsResult	The container that saves the result of the List Multipart Upload request. Type: container Sub-nodes: Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MasUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated, Upload Parent node: none
Bucket	Specify the bucket name. Type: string Parent node: ListMultipartUploadsResult
EncodingType	Specify the encoding type for the returned results. If encoding-type is specified in the request, those elements including Delimiter, KeyMarker, Prefix, NextKeyMarker and Key will be encoded in the returned result. Type: string Parent node: ListMultipartUploadsResult
KeyMarker	Position of the starting object in the list. Type: string Parent node: ListMultipartUploadsResult
UploadIdMarker	Position of the starting Upload ID in the list. Type: string Parent node: ListMultipartUploadsResult
NextKeyMarker	If not all results are returned this time, the response request will include the NextKeyMarker element to indicate the value of KeyMarker in the next request. Type: string Parent node: ListMultipartUploadsResult
NextUploadMarker	If not all results are returned this time, the

	response request will include the NextUploadMarker element to indicate the value of UploadMarker in the next request. Type: string Parent node: ListMultipartUploadsResult
MaxUploads	The maximum upload number returned by the OSS. Type: integer Parent node: ListMultipartUploadsResult
IsTruncated	Specify whether the returned Multipart Upload result list is truncated. The “true” indicates that not all results are returned; “false” indicates that all results are returned. Type: enumerative string Valid values: false, true Default value: false Parent node: ListMultipartUploadsResult
Upload	The container that saves the information about the Multipart Upload event. Type: container Sub-nodes: Key, UploadId, Initiated Parent node: ListMultipartUploadsResult
Key	Name of an object for which a Multipart Upload event is initiated. Type: string Parent node: Upload
UploadId	ID of a Multipart Upload event. Type: string Parent node: Upload
Initiated	Time when a Multipart Upload event is initiated. Type: date Parent node: Upload

Detail analysis

- The maximum value of the “max-uploads” parameter is 1,000.
- The results returned by the OSS are listed in ascending order based on the lexicographic orders of object names; for the same object, the results are listed in ascending time order.
- Using the prefix parameter, you can flexibly manage objects in a bucket in groups (similar to the folder function).
- The List Multipart Uploads request supports five request parameters: prefix, marker, delimiter, upload-id-marker, and max-keys. Based on the combinations of these parameters, you can set rules for querying Multipart Uploads events to obtain the desired query results.

Example

Request example:

```
Get /?uploads  HTTP/1.1
Host:oss-example. oss-cn-hangzhou.aliyuncs.com
Date: Thu, 23 Feb 2012 06:14:27 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:JX75CtQqsmBBz+dcivn7kwBMvOY=
```

Response example:

```
HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1839
Content-type: application/xml
x-oss-request-id: 58a41847-3d93-1905-20db-ba6f561ce67a
Date: Thu, 23 Feb 2012 06:14:27 GMT

<?xml version="1.0" encoding="UTF-8"?>
<ListMultipartUploadsResult xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Bucket>oss-example</Bucket>
<KeyMarker></KeyMarker>
<UploadIdMarker></UploadIdMarker>
<NextKeyMarker>oss.avi</NextKeyMarker>
<NextUploadIdMarker>0004B99B8E707874FC2D692FA5D77D3F</NextUploadIdMarker>
<Delimiter></Delimiter>
<Prefix></Prefix>
<MaxUploads>1000</MaxUploads>
<IsTruncated>false</IsTruncated>
<Upload>
<Key>multipart.data</Key>
<UploadId>0004B999EF518A1FE585B0C9360DC4C8</UploadId>
<Initiated>2012-02-23T04:18:23.000Z</Initiated>
</Upload>
<Upload>
<Key>multipart.data</Key>
<UploadId>0004B999EF5A239BB9138C6227D69F95</UploadId>
<Initiated>2012-02-23T04:18:23.000Z</Initiated>
</Upload>
<Upload>
<Key>oss.avi</Key>
<UploadId>0004B99B8E707874FC2D692FA5D77D3F</UploadId>
<Initiated>2012-02-23T06:14:27.000Z</Initiated>
</Upload>
</ListMultipartUploadsResult>
```

List Parts

The ListParts command can be used to list all successfully uploaded parts mapped to a specific upload ID.

Request syntax

```
Get  /ObjectName?uploadId=UploadId HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Date: GMT Date
Authorization: Signature
```

Request parameters

Name	Description
uploadId	ID of a Multipart Upload event. Type: string Default value: none
max-parts	The maximum part number in the response of the OSS. Type: integer Default value: 1,000
part-number-marker	Starting position of a specific list. A part is listed only when the part number is greater than the value of this parameter. Type: integer Default value: none
encoding-type	Specify the encoding of the returned content and the encoding type. The Key adopts UTF-8 encoding, but the XML 1.0 Standard does not support parsing certain control characters, such as the characters with ASCII values from 0 to 10. In case that the Key contains control characters not supported by the XML 1.0 Standard, you can specify the encoding-type to encode the returned Key. Data type: string Default value: none. Optional value: url

Response elements

Name	Description
ListPartsResult	The container that saves the result of the List Parts request. Type: container Sub-nodes: Bucket, Key, UploadId, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part Parent node: none
Bucket	Specify the bucket name. Type: string Parent node: ListPartsResult

EncodingType	Specify the encoding type for the returned result. If the encoding type is specified in the request, the Key will be encoded in the returned result. Type: string Parent node: ListPartsResult
Key	Object name. Type: string Parent node: ListPartsResult
UploadId	ID of an Upload event. Type: string Parent node: ListPartsResult
PartNumberMarker	Starting position of the part numbers in the listing result. Type: integer Parent node: ListPartsResult
NextPartNumberMarker	If not all results are returned this time, the response request will include the NextPartNumberMarker element to indicate the value of PartNumberMarker in the next request. Type: integer Parent node: ListPartsResult
MaxParts	The maximum part number in the returned request. Type: integer Parent node: ListPartsResult
IsTruncated	Whether the returned result list for List Parts is truncated. The “true” indicates that not all results are returned; “false” indicates that all results are returned. Type: enumerative string Valid values: true, false Parent node: ListPartsResult
Part	The container that saves part information. Type: string Sub-nodes: PartNumber, LastModified, ETag, Size Parent node: ListPartsResult
PartNumber	Part number. Type: integer Parent node: ListPartsResult.Part
LastModified	Time when a part is uploaded. Type: date Parent node: ListPartsResult.part
ETag	ETag value in the content of the uploaded part. Type: string Parent node: ListPartsResult.Part
Size	Size of the uploaded part.

	Type: integer Parent node: ListPartsResult.Part

Detail analysis

- ListParts supports two request parameters: max-parts and part-number-marker.
- The maximum value of the max-parts parameter is 1,000; its default value is also 1,000.
- The results returned by the OSS are listed in ascending order based on the part numbers.
- Because errors may occur in network transmission, it is not recommended that you use the result (part number and ETag value) of List Parts to generate the final part list of Complete Multipart.

Example

Request example:

```
Get  /multipart.data?uploadId=0004B999EF5A239BB9138C6227D69F95  HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 23 Feb 2012 07:13:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:4qOnUMc9UQWqkz8wDqD3lIsa9P8=
```

Response example:

```
HTTP/1.1 200
Server: AliyunOSS
Connection: keep-alive
Content-length: 1221
Content-type: application/xml
x-oss-request-id: 106452c8-10ff-812d-736e-c865294afc1c
Date: Thu, 23 Feb 2012 07:13:28 GMT

<?xml version="1.0" encoding="UTF-8"?>
<ListPartsResult xmlns=”  http://doc.oss-cn-hangzhou.aliyuncs.com”  >
<Bucket>multipart_upload</Bucket>
<Key>multipart.data</Key>
<UploadId>0004B999EF5A239BB9138C6227D69F95</UploadId>
<NextPartNumberMarker>5</NextPartNumberMarker>
<MaxParts>1000</MaxParts>
<IsTruncated>false</IsTruncated>
<Part>
<PartNumber>1</PartNumber>
<LastModified>2012-02-23T07:01:34.000Z</LastModified>
<ETag>&amp;quot;3349DC700140D7F86A078484278075A9&amp;quot;</ETag>
<Size>6291456</Size>
</Part>
<Part>
<PartNumber>2</PartNumber>
<LastModified>2012-02-23T07:01:12.000Z</LastModified>
```

```
<ETag>&amp;quot;3349DC700140D7F86A078484278075A9&amp;quot;</ETag>
<Size>6291456</Size>
</Part>
<Part>
<PartNumber>5</PartNumber>
<LastModified>2012-02-23T07:02:03.000Z</LastModified>
<ETag>&amp;quot;7265F4D211B56873A381D321F586E4A9&amp;quot;</ETag>
<Size>1024</Size>
</Part>
</ListPartsResult>
```

Cross-Origin Resource Sharing

Introduction

Cross-Origin Resource Sharing (CORS) allows web applications to access resources in other domains. With the CORS support, the OSS allows users to develop more flexible web applications. The OSS provides an interface for developers to easily control various permissions for cross-domain access.

Put Bucket cors

With the Put Bucket cors operation, you can set a CORS rule for a specified bucket. If an original rule exists, it will be overwritten.

Request Syntax

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

<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
<CORSRule>
<AllowedOrigin>the origin you want allow CORS request from</AllowedOrigin>
<AllowedOrigin>...</AllowedOrigin>
<AllowedMethod>HTTP method</AllowedMethod>
```

```
<AllowedMethod>...</AllowedMethod>
<AllowedHeader> headers that allowed browser to send</AllowedHeader>
<AllowedHeader>...</AllowedHeader>
<ExposeHeader> headers in response that can access from client app</ExposeHeader>
<ExposeHeader>...</ExposeHeader>
<MaxAgeSeconds>time to cache pre-fight response</MaxAgeSeconds>
</CORSRule>
<CORSRule>
...
</CORSRule>
...
</CORSConfiguration >
```

Request Elements

Name	Description	Essential or Not
CORSRule	CORS rule container. Each bucket allows up to 10 rules Type: Container Parent node: CORSConfiguration	Yes
AllowedOrigin	Indicates the origins allowed for cross-domain requests. Multiple elements can be used to specify multiple allowed origins. Each rule allows up to one wildcard “* ”. If “* ” is specified, cross-domain requests of all origins are allowed. Type: String Parent node: CORSRule	Yes
AllowedMethod	Specifies the allowed methods for cross-domain requests. Type: Enumeration (GET, PUT, DELETE, POST, HEAD) Parent node: CORSRule	Yes
AllowedHeader	Controls whether the headers specified by Access-Control-Request-Headers in the	No

	OPTIONS prefetch command are allowed. Each header specified by Access-Control-Request-Headers must match a value in AllowedHeader. Each rule allows up to one wildcard “*” Type:String Parent node: CORSRule		
ExposeHeader	Specifies the response headers allowing users to access from an application (for example, a Javascript XMLHttpRequest object).	The wildcard “*” is not allowed. Type: String Parent node: CORSRule	No
MaxAgeSeconds	Specifies the cache time for the returned result of a browser prefetch (OPTIONS) request to a specific resource. The unit is seconds. One CORSRule allows up to one such parameter. Type: Integer Parent node: CORSRule	No	
CORSConfiguration	CORS rule container of a bucket Type:Container Parent node: None	Yes	

Detail Analysis

1. CORS is disabled for buckets by default. The origins of all cross-domain requests are forbidden.
2. To use CORS in applications, for example, accessing the OSS from www.a.com through the XMLHttpRequest function of the browser, you need to manually upload a CORS rule through this interface to enable CORS. This rule is described in an XML document.
3. The CORS setting for each bucket is specified by multiple CORS rules. Each bucket allows a

maximum of 10 rules. The uploaded XML document cannot be larger than 16 KB.

4. When the OSS receives a cross-domain request (or an OPTIONS request), it reads the bucket’ s CORS rules and then checks the relevant permissions. The OSS will check each rule sequentially and uses the first rule that matches to approve the request and return the corresponding header. If none of the rules match, the OSS will not attach any CORS header.
5. Successful CORS rule matching must satisfy three conditions. First, the request Origin must match the AllowedOrigin. Second, the request method (e.g. GET, PUT) or the method corresponding to the Access-Control-Request-Method header in an OPTIONS request must match the AllowedMethod. Third, each header contained in the Access-Control-Request-Headers in an OPTIONS request must match the AllowedHeader.
6. If you have uploaded the Content-MD5 request header, the OSS calculates the body’ s Content-MD5 and checks if the two are the same. If the two are different, the error code InvalidDigest is returned.

Example

Example of adding a bucket CORS rule:

```
PUT /?cors HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Content-Length: 186
Date: Fri, 04 May 2012 03:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=

<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
<CORSRule>
<AllowedOrigin>*</AllowedOrigin>
<AllowedMethod>PUT</AllowedMethod>
<AllowedMethod>GET</AllowedMethod>
<AllowedHeader>Authorization</AllowedHeader>
</CORSRule>
<CORSRule>
<AllowedOrigin>http://www.a.com</AllowedOrigin>
<AllowedOrigin>http://www.b.com</AllowedOrigin>
<AllowedMethod>GET</AllowedMethod>
<AllowedHeader> Authorization</AllowedHeader>
<ExposeHeader>x-oss-test</ExposeHeader>
<ExposeHeader>x-oss-test1</ExposeHeader>
<MaxAgeSeconds>100</MaxAgeSeconds>
</CORSRule>
</CORSConfiguration >
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 50519080C4689A033D00235F
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
```

```
Connection: close
Server: AliyunOSS
```

Get Bucket cors

The Get Bucket cors operation is used to obtain the current CORS rules of a specified bucket.

Request Syntax

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

Response Elements

Name	Description
CORSRule	CORS rule container. Each bucket allows up to 10 rules Type:Container Parent node:CORSConfiguration
AllowedOrigin	Indicates the origins allowed for cross-domain requests. Multiple elements can be used to specify multiple allowed origins. Each rule allows up to one wildcard "*". If "*" is specified, cross-domain requests of all origins are allowed. Type: String Parent node: CORSRule
AllowedMethod	Specifies the allowed methods for cross-domain requests. Type: Enumeration (GET, PUT, DELETE, POST, HEAD) Parent node:CORSRule
AllowedHeader	Controls whether the headers specified by Access-Control-Request-Headers in the OPTIONS prefetch command are allowed. Each header specified by Access-

	Control-Request-Headers must match a value in AllowedHeader.Each rule allows up to one wildcard “* ” Type:String Parent node: CORSRule	
ExposeHeader	Specifies the response headers allowing users to access from an application (for example, a Javascript XMLHttpRequest object).	The wildcard “* ” is not allowed. Type: String Parent node: CORSRule
MaxAgeSeconds	Specifies the cache time for the returned result of a browser prefetch (OPTIONS) request to a specific resource. The unit is seconds. One CORSRule allows up to one such parameter. Type: Integer Parent node:CORSRule	
CORSConfiguration	CORS rule container of a bucket Type:Container Parent node: None	

Detail Analysis

1. If a bucket does not exist, error “404 no content” is returned. The error code is: NoSuchBucket.
2. Only the bucket owner can obtain CORS rules. Otherwise, error 403 Forbidden is returned with the error code: AccessDenied.
3. If CORS rules do not exist, the OSS will return the “404 Not Found” error with the error code NoSuchCORSConfiguration.

Example

Request example:

```
Get /?cors HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Thu, 13 Sep 2012 07:51:28 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc: BuG4rRK+zNhH1AcF51NNHD39zXw=
```

Return example with CORS rules already set:

```
HTTP/1.1 200
```

```
x-oss-request-id: 50519080C4689A033D00235F
Date: Thu, 13 Sep 2012 07:51:28 GMT
Connection: close
Content-Length: 218
Server: AliyunOSS

<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration>
<CORSRule>
<AllowedOrigin>*</AllowedOrigin>
<AllowedMethod>GET</AllowedMethod>
<AllowedHeader>*</AllowedHeader>
<ExposeHeader>x-oss-test</ExposeHeader>
<MaxAgeSeconds>100</MaxAgeSeconds>
</CORSRule>
</CORSConfiguration>
```

Delete Bucket cors

Delete Bucket cors is used to disable the CORS function for a specified bucket and clear all the rules.

Request Syntax

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

Detail Analysis

1. If a bucket does not exist, error “404 no content” is returned with the error code: NoSuchBucket.
2. Only the bucket owner can delete the CORS rules of this bucket.If you try to operate a bucket which does not belong to you, the OSS returns error 403 Forbidden with the error code: AccessDenied.

Example

Request example:

```
DELETE /?cors HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
```

```
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:LnM4AZ1OeIduZF5vGFWicOMEkVg=
```

Return example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 5051845BC4689A033D0022BC
Date: Fri, 24 Feb 2012 05:45:34 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

Option Object

Before sending a cross-domain request, the browser sends a preflight request (OPTIONS) containing a specific origin, HTTP method, and header information to the OSS to determine whether to send a real request.The OSS can enable CORS for a bucket through the Put Bucket cors interface. After CORS is enabled, the OSS will assess whether to allow the preflight request of the browser based on the specified rules. If the OSS does not allow this request or CORS is disabled, error 403 Forbidden is returned.

Request Syntax

```
OPTIONS /ObjectName HTTP/1.1
Host: BucketName.oss-cn-hangzhou.aliyuncs.com
Origin:Origin
Access-Control-Request-Method:HTTP method
Access-Control-Request-Headers:Request Headers
```

Request Header

Name	Description
Origin	Origin of a request, used to identify a cross-domain request. Type: string Default value: none
Access-Control-Request-Method	Methods to be used in an actual request. Type: string Default value: none
Access-Control-Request-Headers	Headers, except simple headers, to be used in an actual request. Type: string Default value: none

Response Header

Name	Description
Access-Control-Allow-Origin	Origin contained in a request. This header will not be contained if this request is not allowed. Type: String
Access-Control-Allow-Methods	HTTP method used by a request. This header will not be contained if this request is not allowed. Type: String
Access-Control-Allow-Headers	Header list carried in a request. If the request contains forbidden headers, this header will not be contained and the request will be rejected. Type: String
Access-Control-Expose-Headers	Header list that can be accessed by the client’ s JavaScript application. Type: String
Access-Control-Max-Age	Time duration when the browser can buffer the preflight results. The unit is seconds Type: Integer

Example

Request example:

```
OPTIONS /testobject HTTP/1.1
Host: oss-example.oss-cn-hangzhou.aliyuncs.com
Date: Fri, 24 Feb 2012 05:45:34 GMT
Origin:http://www.example.com
Access-Control-Request-Method:PUT
Access-Control-Request-Headers:x-oss-test
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 5051845BC4689A033D0022BC
Date: Fri, 24 Feb 2012 05:45:34 GMT
Access-Control-Allow-Origin: http://www.example.com
Access-Control-Allow-Methods: PUT
Access-Control-Expose-Headers: x-oss-test
Connection: close
Content-Length: 0
Server: AliyunOSS
```