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 sender of a request by using the Access Key ID/Access Key Secret symmetric encryption method. The AccessKey ID identifies a user. With the AccessKey Secret, a user can encrypt the signature string and the OSS can verify the access key of the signature string. The AccessKey Secret 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:

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confidential has operation permissions on specified resources for a specified period of time, and the permissions are withdrawn after the period of time expires.

For details, refer to Access Identity Verification in the OSS product documentation

Before sending a request to the OSS as an individual identity, a user needs to generate a signature string for the request according to the format specified by the OSS and then encrypt the signature string using the AccessKey Secret to generate a verification code. After receiving the request, the OSS finds the corresponding Access Key Secret based on the Access Key ID, and obtains the signature string and verification code in the same way. If the obtained verification code is the same as the provided verification code, the request is assumed valid. If not, the OSS rejects the request and returns an HTTP 403 error.

Adding a Signature to a Header

A user can add an authorization header to carry the signature information in an HTTP request, thereby indicating that the message has been authorized.

Calculation of the Authorization Field

```
"Authorization: OSS " + Access Key Id + ":" + Signature

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

- AccessKeySecret indicates the key required for the signature
- VERB indicates the HTTP request method, including PUT, GET, POST, HEAD, DELETE, and so on
- CONTENT-MD5 indicates the value of MD5 in the request content data. For details, see RFC2616
- CONTENT-TYPE indicates the type of the request content
- DATE indicates the operation time, and it must be in the GMT format supported in HTTP1.1
- CanonicalizedOSSHeaders indicates an assembly of HTTP headers whose prefixes are “x-oss-“
- CanonicalizedResource indicates the OSS resource that the user wants to access.

Among them, the values of DATE and CanonicalizedResource cannot be blank. 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 rejects the service and returns an HTTP 403 error.

NOTE

"Authorization: OSS " + Access Key Id + ":" + Signature requires a space after OSS and no space around colon.

Constructing CanonicalizedOSSHeaders

All the HTTP headers whose prefixes are "x-oss-" are called CanonicalizedOSSHeaders, and the method for constructing CanonicalizedOSSHeaders 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. Sort all HTTP request headers obtained in the foregoing step in the lexicographically ascending order.
3. Combine request headers with the same name according to chapter 4.2 in RFC2616 (where two values are separated by a comma (,)). For example, if there are two request headers 'x-oss-meta-name' whose values are 'TaoBao' and 'Alipay', the combined value is 'x-oss-meta-name:TaoBao,Alipay'.
4. Delete any space at either side of a separator between the request header and content. For example, convert 'x-oss-meta-name: TaoBao,Alipay' into: 'x-oss-meta-name:TaoBao,Alipay'.
5. Separate all the headers and content using the '\n' separator to form the final CanonicalizedOSSHeader.

NOTE

1. No need to add "\n" in the end if CanonicalizedOSSHeaders is null.
2. Attention to the "/n" if CanonicalizedOSSHeaders is not null ex. x-oss-meta-a\n for one CanonicalizedOSSHeaders and `x-oss-meta-a:a\nx-oss-meta-b:b\nx-oss-meta-c:c\n' for more than one canon.

Constructing 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 as a blank string ("").
2. Set "/BucketName/ObjectName" to the OSS resource to be accessed (this parameter can be left empty if there is no ObjectName).
3. If the requested resource includes sub-resources, 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, for example, /BucketName/ObjectName?acl &uploadId=UploadId

4. If the user specifies in the query string that the header of the response request needs to be overridden, 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 lexicographical order of the parameters. In this case, CanonicalizedResource is, for example,/BucketName/ObjectName?acl&response-content-type=ContentType & uploadId =UploadId. For details about the override request headers supported by the OSS, refer to Get Object.

The sub-resources currently supported by the OSS include: acl, uploadId, partNumber, uploads, logging, website, location, lifecycle, referer, cors, delete, append, and position, and security-token used in the STS service.

Rules for Calculating a Signature Header

1. The string used for signature must be in UTF-8 format. A signature string containing Chinese characters must undergo UTF-8 encoding, and is then used together with Access Key Secret to calculate the final signature.
2. The signing method adopts the HMAC-SHA1 method defined in RFC 2104, where Key is Access Key Secret.
3. content-type and content-md5 are not mandatory in the request. If the request requires signature verification, the null value can be replaced by the linefeed '\n'.
4. Among all the non-HTTP-standard headers, only the headers starting with "x-oss-" requires signature strings, and other non-HTTP-standard headers (for example, x-oss-magic in an example above) are ignored by the OSS.
5. Headers starting with "x-oss-" must comply with the following specifications before being used for signature verification:
    - The header name is changed to lowercase letters
    - The headers are sorted in the lexicographically ascending order.
    - There must be no space before and after the colon that separates the header name and value.
    - Each header is followed by a linefeed '\n'. If there is no header, CanonicalizedOSSHeaders is set to null.

Example Signature

For example: to add the following signature information:

```
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
```

Assuming that accessId is: “44CF9590006BF252F707” and Access Key Secret “OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV”, the method for adding a signature is as follows:

request	calculation	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”

Example python codes

```
import base64
import hmac
import sha
h = hmac.new("OtxrzxIsfpFjA7SwPzILwy8Bw21TLhquhboDYROV",
"PUT\neB5eJF1ptWaXm4bijSPyxw==\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)
base64.encodestring(h.digest()).strip()
```

A calculated signature is 26NBxoKdsyly4EDv6inkoDft/yA=, which, together with the Authorization header, constitutes the final message that needs 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 there is no incoming AccessID or the incoming AccessID is inactive, error 403 Forbidden is returned, where the error code is InvalidAccessKeyId.
2. If the Authorization value in the user request header is invalid, error 400 Bad Request is returned, where the error code is InvalidArgument.
3. All the requests of the OSS must use the GMT time format stipulated by the HTTP 1.1

protocol. The following three date formats are available:date1 = 2DIGIT SP month SP 4DIGIT; day month year (e.g., 02 Jun 1982)date2 = 2DIGIT "-" month "-" 2DIGIT; day-month-year (e.g., 02-Jun-82)date3 = month SP (2DIGIT | (SP 1DIGIT)); month day (e.g., Jun 2) [Note that there are two spaces before "2"]In the foregoing three date formats, “day” occupies “2 DIGIT”. 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, error 403 Forbidden is returned, where the error code is AccessDenied.
5. The request must be input within 15 minutes based on the current time of the OSS server; otherwise, error 403 Forbidden is returned, where the error code is RequestTimeTooSkewed.
6. If AccessID is active, and the OSS determines that the signature of the user request is incorrect, error 403 Forbidden is returned, and the correct signature string for verification and encryption are 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>
```

Adding a Signature to a URL

In addition to using an Authorization Header, a user can also add signature information to a URL so that the user can forward the URL to a third party, to implement authorized access.

Implementation Method

Example URL that includes a signature:

```
http://oss-example.oss-cn-hangzhou.aliyuncs.com/oss-api.pdf?OSSAccessKeyId=44CF9590006BF252F707&Expires=1141889120&Signature=vjbyPxybdZaNmGa%2ByT272YEAiv4%3D
```

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

- Expires: 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 times out is returned. For example: if the current time is 1141889060 and the developer wants to create a URL that will automatically become invalid 60 seconds later, the value of Expires can be set to 1141889120.
- OSSAccessKeyId: indicates an Access Key ID.
- Signature: indicates the signature information. For all the requests and various Header parameters supported by the OSS, the algorithms for adding a signature to a URL are basically the same as those in Adding a Signature to a Header.

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

The differences 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 Header at the same time.
3. If there are more than one incoming value of each of Signature, Expires, and OSSAccessKeyId, the initial values are used.
4. Whether the request time is later than the Expires time is verified before the signature.

Example Python Codes

Example python codes for adding 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_plus (base64.encodestring(h.digest()).strip())
```

Detail Analysis

1. If the method of adding a signature to a URL is used, the authorized data will be exposed on the Internet after the authorization period expires. Use the method with caution.
2. The PUT and GET requests both support the manner of adding a signature in a URL.
3. When a signature is added to a URL, the sequence of Signature, Expires, and OSSAccessKeyId can be changed. If one or more parameters of Signature, Expires, and OSSAccessKeyId are missing, error 403 Forbidden is returned, where the error code is AccessDenied.
4. If the current access time is later than the Expires time set in the request, error 403 Forbidden is returned, where the error code is AccessDenied.
5. If the format of the Expires time is incorrect, error 403 Forbidden is returned, where the error code is AccessDenied.
6. If the URL includes one or more parameters of Signature, Expires, and OSSAccessKeyId and the header includes signature information, error 400 Bad Request is returned, where the error code is 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. (The Date request header still exists in the request, and the Date header does not need to be added to the signature string.)

Temporary Access Credential

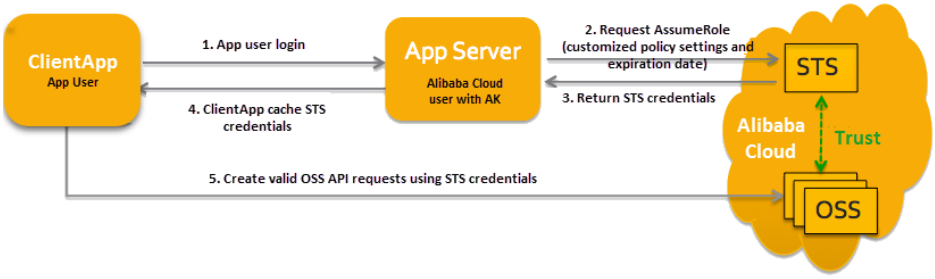
Introduction to STS

Through AliCloud Security Token Service (STS), the OSS can grant temporary authorized access. AliCloud 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 associated users with customized permissions and validity periods (where you can manage the user IDs). Third-party applications or associated users can use these access credentials to directly call the AliCloud product APIs or use the SDKs provided by AliCloud products to access the cloud product APIs.

- You do not need to expose you 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 period 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 in as the app user. App user IDs are managed by the customer. The customer can customize the ID management system and may also 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 app users and the expiry time of the authorization.Then, the security token is obtained by calling the AssumeRole interface of the STS. For details about role management and usage, see Role Management in the RAM User Guide.
3. The STS returns a valid access credential to the AppServer, where the access credential includes 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 caches 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 for AliCloud Service APIs. The ECS is aware of the STS access credential, relies on STS to verify the credential, and correctly responds to the user request.

For details about the STS security token, see Role Management in the RAM User Guide. The key is to obtain a valid access credential by simply calling the STS interface AssumeRole. The method can also

be called by using the STS DSK. Clicking to View Details

Using 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 of creating an authorized access signature is the basically same as the method of Adding a Signature to a Header by using AccessKey that uses a root account. 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 bother manners are selected, the OSS returns an InvalidArgument error.
    - The header includes ‘x-oss-security-token: SecurityToken’. 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

The OSS provides an Access Control List (ACL) for permission control. The OSS ACL provides bucket-level access control. Currently, three access permissions are provided for a bucket: public-read-write, public-read, and private. They are described as follows:

- public-read-write: Any one (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 and Delete Object) on the objects in the bucket. Any one (including anonymous users) can perform read operations (Get Object) on the objects in the bucket.
- private: Only the creator of a bucket can perform read and write operations (including Put, Delete, and Get Object) on the objects in the bucket. Others cannot access the 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	to 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	to 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	Container for saving results of the Get Service request. Type: Container Subnode: 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	to 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	to indicate whether all results have been returned: “true” indicates that not all results are returned; “false” indicates that all results are returned. This node is available only when not all buckets are returned. Type: Enumerated string Valid value: true and false Parent node:ListAllMyBucketsResult
NextMarker	to 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	Container used for saving 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	Container used for saving the information about multiple Buckets. Type: Container Subnode: Bucket Parent node:ListAllMyBucketsResult
Bucket	Container used for saving the bucket information. Type: Container Subnode: Name, CreationDate, Location Parent node:ListAllMyBucketsResult.Buckets
Name	Bucket name. Type: String Parent node:ListAllMyBucketsResult.Buckets.Bucket

CreateDate	Bucket creation time Type: Time (format: yyyy-mm-ddThh:mm:ss.timezone, e.g., 2011-12-01T12:27:13.000Z) Parent node:ListAllMyBucketsResult.Buckets.Bucket
Location	Data center in which a bucket is located 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. The error code is 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: 10.97.188.37
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: close
Server: AliyunOSS
x-oss-request-id: 5374A2880232A65C23002D74

<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
<Owner>
<ID>ut_test_put_bucket</ID>
<DisplayName>ut_test_put_bucket</DisplayName>
</Owner>
<Buckets>
<Bucket>
<Location>oss-cn-hangzhou-a</Location>
```

```
<Name>xz02tphky6fjfiuc0</Name>
<CreationDate>2014-05-15T11:18:32.000Z</CreationDate>
</Bucket>
<Bucket>
<Location>oss-cn-hangzhou-a</Location>
<Name>xz02tphky6fjfiuc1</Name>
<CreationDate>2014-05-15T11:18:32.000Z</CreationDate>
</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: 10.97.188.37
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: close
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>
<Location>oss-cn-hangzhou-a</Location>
<Name>xz02tphky6fjfiuc0</Name>
<CreationDate>2014-05-15T11:18:32.000Z</CreationDate>
</Bucket>
</Buckets>
</ListAllMyBucketsResult>
```

Bucket Operations

Put Bucket

PutBucket is used to create buckets (anonymous access is not supported). By default, the created buckets are located in the default data center: oss-cn-hangzhou. In the request body, you can specify the data center in which a bucket is located, so as to optimize delay, minimize costs or meet regulatory requirements. When the data center in which a bucket is located is determined, all objects in this bucket will be stored in the corresponding region until these objects are explicitly moved to other data centers. For more information, refer to Bucket and Data Center.

Request Syntax

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

<?xml version="1.0" encoding="UTF-8"?>
< CreateBucketConfiguration >
< LocationConstraint >BucketRegion</ LocationConstraint >
</CreateBucketConfiguration >
```

Request Elements

- Asia Pacific (Singapore) data center:oss-ap-southeast-1

Name	Description
CreateBucketConfiguration	Container used for saving bucket settings. Type: Container Subnode: LocationConstraint Parent node:none
LocationConstraint	to specify the data center in which a bucket is located. For more information about data centers and terminal domain names, refer to Access Domain Names and Data Centers. Type: Enumeration Valid value: oss-cn-hangzhou, oss-cn-qingdao, oss-cn-beijing, oss-cn-hongkong, oss-cn-shenzhen, oss-cn-shanghai, oss-us-west-1, oss-ap-southeast-1 Default value: oss-cn-hangzhou Parent node:CreateBucketConfiguration

Detail Analysis

1. You can use the “x-oss-acl” header in a Put request to set access permissions for a bucket. Currently, a bucket is enabled with three types of access permissions: public-read-write, public-read, and private.
2. When using the default domain name of the OSS to create a bucket, you do not have to specify a data center for the bucket (null request body), and instead, the OSS will specify the default data center “oss-cn-hangzhou” for the bucket. At this time, you can also set the data center in which the bucket is located to any one of the valid data center values.
3. If you use the terminal domain name of a data center to create a bucket, you must specify the data center in which the bucket is to be located and the data center must match the terminal domain name.
4. If the specified data center value is invalid when you create a bucket, 400 is returned with the error code: InvalidLocationConstraint.
5. If the specified data center is different from the requested terminal domain name, 400 is returned with the error code: IllegalLocationConstraintException.
6. Data centers for the existing buckets cannot be modified. Otherwise, 409 Conflict is returned with the error information: Bucket already exists can’t modify location.
7. If the requested bucket already exists and is owned by the requester, 200 OK is returned for success.
8. If the requested bucket already exists but is not owned by the requester, 409 Conflict is returned. The error code is BucketAlreadyExists.
9. If the bucket to be created does not conform to the naming conventions, the message of 400 Bad Request is returned. The error code is InvalidBucketName.
10. If the information for user authentication is not introduced when you initiate a PUT Bucket request, the message of 403 Forbidden is returned. The error code is AccessDenied.
11. If the maximum number of bucket creation (10) is exceeded when you initiate a PutBucket request, the message of 400 Bad Request is returned. The error code is TooManyBuckets.
12. 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=

<?xml version="1.0" encoding="UTF-8"?>
<CreateBucketConfiguration >
<LocationConstraint >oss-cn-hangzhou</LocationConstraint >
```

```
</CreateBucketConfiguration >
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 7c9e8b71-3c6a-1b7d-2361-093f1af5f5e9
Date: Fri, 24 Feb 2012 03:15:40 GMT
Location: /oss-example
Content-Length: 0
Connection: close
Server: AliyunOSS
```

Put Bucket Acl

The Put Bucket ACL interface is used to modify the access permissions for a bucket. Currently, a bucket is enabled with three types of access permissions: 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 is returned; otherwise, the corresponding error code and prompt message are 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 is returned. The error code is 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=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 248c6483-2a95-622e-3022-ebe65d8aad5f
Date: Fri, 24 Feb 2012 03:21:12 GMT
Content-Length: 0
Connection: close
Server: AliyunOSS
```

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

Error return example:

```
HTTP/1.1 400 Bad Request
x-oss-request-id: 4e63c87a-71dc-87f7-11b5-583a600e0038
Date: Fri, 24 Feb 2012 03:55:00 GMT
Content-Length: 309
Content-Type: text/xml; charset=UTF-8
Connection: close
Server: AliyunOSS

<?xml version="1.0" ?>
<Error xmlns=” http://doc.oss-cn-hangzhou.aliyuncs.com” >
<Code>
InvalidArgument
</Code>
<Message>
</Message>
<ArgumentValue>
error-acl
</ArgumentValue>
<ArgumentName>
x-oss-acl
</ArgumentName>
<RequestId>
4e63c87a-71dc-87f7-11b5-583a600e0038
</RequestId>
<HostId>
oss-cn-hangzhou.aliyuncs.com
</HostId>
</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	Container for logging status information. Type: Container Children: LoggingEnabled Ancestry: None	Yes
LoggingEnabled	Container for logging information. This element is present when you are enabling logging (and not present when you are disabling logging). Type: Container	No

	Children: TargetBucket, TargetPrefix Ancestry: BucketLoggingStatus	
TargetBucket	Specifies the bucket where you want Aliyun OSS to store server access logs. Type: String Children: None Ancestry: BucketLoggingStatus.LoggingEnabled	Yes,When BucketLoggingStatus.LoggingEnabled
TargetPrefix	This element lets you specify a prefix for the objects that the log files will be stored. Type: String Children: None Ancestry: 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
Reserved	-	Reserved field
Requester AliCloud ID	1657136103983691	AliCloud 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 request simultaneously generated by a two-node cluster, “-” for no
Reserved	-	Reserved field

Detail Analysis

1. The source bucket and target bucket must belong to the same user.
2. 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.
3. The source bucket and target bucket can be the same bucket or different buckets. You can save logs from multiple source buckets to the same target bucket (it is recommended that you set TargetPrefix to different values).
4. 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.
5. All PUT Bucket Logging requests must be provided with signatures, meaning that anonymous access is not supported.
6. If the initiator of a PUT Bucket Logging request is not the owner of the source bucket (BucketName in the request example), the OSS returns error code 403.
7. If the source bucket does not exist, the OSS returns error code: NoSuchBucket.
8. If the initiator of a PUT Bucket Logging request is not the owner of the target bucket (TargetBucket in the request example), the OSS returns error code 403. If the target bucket does not exist, the OSS returns error code: InvalidTargetBucketForLogging.
9. The source bucket and target bucket must belong to the same data center. Otherwise, error 400 is returned with error code: InvalidTargetBucketForLogging.
10. If the XML in a PUT Bucket Logging request is invalid, the error code: MalformedXML is returned.
11. 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).
12. When the source bucket is deleted, the corresponding logging rules are also deleted.
13. The OSS generates a bucket access log file every hour. However, all requests in an hour may not be recorded in the log file of the hour and may be recorded in the previous or next log file.
14. In the naming rules for log files generated by the OSS, “UniqueString” is just a UUID that the OSS generates for an object to uniquely identify the file.
15. Each time the OSS generates a bucket access log file, a PUT operation is counted 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.
16. 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.pnghttp://oss-example.oss-cn-hangzhou.aliyuncs.com/aliyun-logo.png?x-user=adminWhen 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.

17. You may see “-“ in any field of OSS logs. It indicates that data is unknown or the field is invalid for the current request.
18. 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.
19. 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 the request for enabling bucket access logs:

```
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>
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 19a86d66-3492-0465-12af-7bec0938d0f9
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: close
Server: AliyunOSS
```

Example of the request for disabling bucket access logs:

```
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>
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 5ef71389-094b-0d94-284d-6c4e49c37409
Date: Fri, 04 May 2012 04:21:12 GMT
Content-Length: 0
Connection: close
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	Container for Key element No Type: Container Ancestors:WebsiteConfiguration	No
IndexDocument	Container for the Suffix element. Type: Container Ancestors:WebsiteConfiguration	Yes

Key	The object key name to use when a 4XX class error occurs Type: String Ancestors:WebsiteConfiguration.ErrorDocument Condition: Required when ErrorDocument is specified	Conditional
Suffix	A suffix that is appended to a request that is for a directory on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html) The suffix must not be empty and must not include a slash character. Type: String Ancestors:WebsiteConfiguration.IndexDocument	Yes
WebsiteConfiguration	Container for the request Type: Container Ancestors: 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 the specific configuration method, refer to the section 3.4: Binding Custom Domain Names.
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 the static website hosting mode, the OSS returns the index page for anonymous access to the root domain name of the static website, and returns the result of Get Bucket for signed access to the root domain name of the static website.
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

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>
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 19a86d66-3492-0465-12af-7bec0938d0f9
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: close
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 detailed information about the Bucket Referer Anti-leech Protection, refer to OSS Anti-leech Protection.

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	Essential or Not
RefererConfiguration	Container used for saving the content of referer configuration Type:Container Subnode: AllowEmptyReferer node, RefererList node Parent node: None	Yes
AllowEmptyReferer	to specify whether the access request with the referer field being null is allowed. Type: Enumerated string Valid value: true or false. Default value:true Parent node: RefererConfiguration	Yes
RefererList	Container used for saving the referer access white list. Type: Container Parent node:RefererConfiguration Subnode: Referer	Yes
RefererList	to specify a referer access white list. 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 is returned. the error code is 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 white list with the white list in the RefererList. When the user-uploaded RefererList is empty (containing no referer request element), this operation overwrites the configured white list, that is, the previously configured RefererList is deleted.

4. 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

Request example:

Request example 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>
```

Request example 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>
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 19a86d66-3492-0465-12af-7bec0938d0f9
Date: Fri, 04 May 2012 03:21:12 GMT
Content-Length: 0
Connection: close
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>
</Rule>
</LifecycleConfiguration>
```

Request Elements

Name	Description	Essential or Not
Date	to specify when the rules take effect. The date must conform to the ISO8601 format and always be UTC 00:00 am. Type: String Parent node: Expiration	Days or Date
Days	to specify how many days after the last object modification until the rules take effect. Type: Positive integer Parent node: Expiration	Days or Date
Expiration	to specify the expiration properties of the rules.	Yes

	Type: Container Subnode: Days or Date Parent node: Rule	
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 Subnode: None Parent node: Rule	No
LifecycleConfiguration	Container used for storing lifecycle configurations, which can hold the maximum of 1000 rules. Type: Container Subnode: Rule Parent node: None	Yes
Prefix	to specify the prefix applicable to a rule. Only those objects with the matching prefix can be affected by the rule. Type: String Subnode:None Parent node: Rule	Yes
Rule	to describe a rule Type:Container Subnode: 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 is returned. The error code is AccessDenied.
2. If no lifecycle has been set previously, this operation creates a new lifecycle configuration or overwrites the previous configuration.

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

<Rule>
<ID>delete at a date</ID>
<Prefix>backup</Prefix>
<Status>Enabled</Status>
<Expiration>
<Date>2022-10-11T00:00:00.000Z</Date>
</Expiration>
</Rule>
</LifecycleConfiguration>
```

Return example:

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

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	to set the returned results to begin from the first entry after the marker in alphabetical order. Data type:String Default value: None
max-uploads	to limit the maximum number of objects returned for one request. If not specified, the default value is 100. The max-keys value cannot exceed 1000. Data type:String Default value:100
prefix	to 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	Specifies the encoding of the returned content and the encoding type. The object key uses UTF-8 characters, 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 object key contains control characters not supported by the xml 1.0 standard, you can specify encoding-type to encode the returned object key. Data type:String Default value: None

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
encoding-type	Specifies the encoding type in the returned result. If encoding-type is specified in the request, those elements including Delimiter, Marker, Prefix, NextMarker and Key are encoded in the returned result. 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 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 Parent node:ListBucketResult.Contents
ID	User ID of the bucket owner. Type: String Parent node:ListBucketResult.Contents.Owner
IsTruncated	to indicate 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: Enumerated string

	Valid value:true and false Parent node:ListBucketResult
Key	Object’ s Key. Type:String Parent node:ListBucketResult.Contents
LastModified	Time when the object is last modified. Type: Time Parent node:ListBucketResult.Contents
ListBucketResult	Container used for saving the results of the Get Bucket request. Type: Container subnode: Name, Prefix, Marker, MaxKeys, Delimiter, IsTruncated, Nextmarker, Contents Parent node:None
Marker	to 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 subnode: 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, error 404 Not Found is returned with the error code: NoSuchBucket.

3. If you do not have the permission to access the bucket, error 403 Forbidden is returned with the error code: AccessDenied.
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 exist in the list actually, 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 1000, error 400 Bad Request is returned. The error code is InvalidArgument.
6. If the prefix, marker or delimiter parameters do not meet the length requirement, 400 Bad Request is returned. The error code is InvalidArgument.
7. The prefix and marker parameters are used to achieve display by pages, and the parameter length must be less than 1024 bytes.
8. Setting a prefix as the name of a folder enumerates the files starting with this prefix, recursively returning all files and subfolders 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=
```

Return example:

```
HTTP/1.1 200 OK
```

```
x-oss-request-id: 248c6483-2a95-622e-3022-ebe65d8aad5f
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1866
Connection: close
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>
```

Request example (including prefix):

```
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=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 25cb535f-1feb-1e90-2f22-12176bcb563e
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 1464
Connection: close
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>
```

Request example (including 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=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 0b05f9b1-539e-a858-0a81-9ca13d8a8011
Date: Fri, 24 Feb 2012 08:43:27 GMT
Content-Type: application/xml
Content-Length: 712
Connection: close
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	Container used for saving the results of Get Bucket ACL Type:Container Parent node:None
DisplayName	Name of the bucket owner. (Currently, the name is the same as the bucket owner ID) Type:String Parent node:AccessControlPolicy.Owner
Grant	ACL permissions of the bucket. Type: Enumerated string Valid value: 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=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 6f720c98-40fe-6de0-047b-e7fb08c4059b
Date: Fri, 24 Feb 2012 04:11:23 GMT
Content-Length: 253
Content-Tupe: application/xml
Connection: close
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	Specifies the Region where the bucket resides. Type: String Valid Values: oss-cn-hangzhou, oss-cn-qingdao, oss-cn-beijing, oss-cn-hongkong, oss-cn-shenzhen, oss-cn-shanghai

Detail Analysis

1. Only the bucket owner can view the bucket location information, otherwise, error 403 Forbidden is returned with the error code: AccessDenied.
2. Currently, the valid values of LocationConstraint are: oss-cn-hangzhou, oss-cn-qingdao, oss-cn-beijing, oss-cn-hongkong, oss-cn-shenzhen, oss-cn-shanghai, oss-us-west-1 and oss-ap-southeast-1; respectively corresponding to Hangzhou data center, Qingdao data center, Beijing data center, Hong Kong data center, Shenzhen data center, US Silicon Valley 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=
```

Return example with logging rules already set:

```
HTTP/1.1 200
x-oss-request-id: 513836E0F687780D1A690708
Date: Fri, 15 Mar 2013 05:31:04 GMT
Connection: close
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	Container for the response. Type: Container Ancestor: None
LoggingEnabled	Container for logging information. This element and its children are present when logging is enabled; otherwise, this element and its children are absent. Type: Container Ancestor: BucketLoggingStatus
TargetBucket	This element specifies the bucket where server access logs will be delivered. Type: String Ancestor: BucketLoggingStatus.LoggingEnabled
TargetPrefix	Specifies the prefix for the keys that the log files are being stored for. Type: String Ancestor: BucketLoggingStatus.LoggingEnabled

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 view the bucket’s access log configurations. Otherwise, error 403 Forbidden is returned with the error code: AccessDenied.
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=
```

Return example with logging rules already set:

```
HTTP/1.1 200
x-oss-request-id: 7faf664d-0cad-852e-4b38-2ac2232e7e7f
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: close
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>
```

Return example with LOG rules not set:

```
HTTP/1.1 200
x-oss-request-id: 7faf664d-0cad-852e-4b38-2ac2232e7e7f
Date: Fri, 04 May 2012 05:31:04 GMT
Connection: close
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	Container for Key element No Type: Container Ancestors:WebsiteConfiguration
IndexDocument	Container for the Suffix element. Type: Container Ancestors:WebsiteConfiguration
Key	The object key name to use when a 4XX class error occurs Type: String Ancestors:WebsiteConfiguration.ErrorDocument Condition: Required when ErrorDocument is specified
Suffix	A suffix that is appended to a request that is for a directory on the website endpoint (e.g. if the suffix is index.html and you make a request to samplebucket/images/ the data that is returned will be for the object with the key name images/index.html) The suffix must not be empty and must not include a slash character. Type: String Ancestors:WebsiteConfiguration.IndexDocument
WebsiteConfiguration	Container for the request Type: Container Ancestors: 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 view the bucket’s static website hosting status. Otherwise, error 403 Forbidden is returned with the error code: AccessDenied.
3. If the static website hosting function is not set for the source bucket, the 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=
```

Return example with logging 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"?>
<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: 7faf664d-0cad-852e-4b38-2ac2232e7e7f
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: close
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 Lifecycle operation is used to view the referer configuration of a bucket. For detailed information about the Bucket Referer Anti-leech Protection, refer to OSS Anti-leech Protection.

Request Syntax

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

Response Elements

Name	Description
RefererConfiguration	Container used for saving the content of referer configuration Type: Container Subnode: AllowEmptyReferer node, RefererList node Parent node: None
AllowEmptyReferer	to specify whether the access request with the referer field being null is allowed. Type: Enumerated string Valid value: true or false. Default value: true Parent node:RefererConfiguration
RefererList	Container used for saving the referer access white list. Type: Container Parent node: RefererConfiguration Subnode:Referer
RefererList	to specify a referer access white list. Type: String Parent node:RefererList

Detail Analysis

1. If the bucket does not exist, error 404 is returned. The error code is NoSuchBucket.
2. Only the bucket owner can view the bucket’ s referer configuration. Otherwise, error 403 Forbidden is returned with the error code: AccessDenied.
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=
```

Return example with referer 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"?>
<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>
```

Return example with no referer rules being set:

```
HTTP/1.1 200
x-oss-request-id: 7faf664d-0cad-852e-4b38-2ac2232e7e7f
Date: Thu, 13 Sep 2012 07:56:46 GMT
Connection: close
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 bucket owner can view the bucket’ s lifecycle configuration. Otherwise, error 403 Forbidden is returned with the error code: AccessDenied.
2. If the bucket or lifecycle does not exist, error 404 Not Found is returned with the error code: NoSuchBucket or NoSuchLifecycle.

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=
```

Return example with lifecycle already set:

```
HTTP/1.1 200
x-oss-request-id: 534B372974E88A4D89060099
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: close
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>
```

Return example with lifecycle not set:

```
```

```
HTTP/1.1 404
x-oss-request-id: 534B372974E88A4D89060099
Date: Mon, 14 Apr 2014 01:17:29 GMT
Connection: close
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

Delete Bucket 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, error “404 no content” is returned. The error code is NoSuchBucket.
2. To prevent accidental deletion, the OSS does not allow users to delete a non-empty bucket.
3. If you try to delete a non-empty bucket, error 409 Conflict is returned with the error code: BucketNotEmpty.
4. Only the bucket owner can delete this bucket. If you try to delete a bucket you have no permission for, error 403 Forbidden is returned. The error code is 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=
```

Return example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 7faf664d-0cad-852e-4b38-2ac2232e7e7f
Date: Fri, 24 Feb 2012 05:31:04 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

Delete Bucket Logging

The Delete Bucket Logging operation 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 a bucket does not exist, error “404 no content” is returned with the error code: NoSuchBucket.
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, the OSS returns 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: 5051842FC4689A033D0022BB
Date: Fri, 24 Feb 2012 05:35:24 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

Delete Bucket Website

The Delete Bucket Website operation 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 a bucket does not exist, error “404 no content” is returned with the error code: NoSuchBucket.
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, the OSS returns 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=
```

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
```

Delete Bucket Lifecycle

Delete Bucket Lifecycle 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, error 404 Not Found is returned with the error code: NoSuchBucket or NoSuchLifecycle.
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, the OSS returns 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=
```

Return example:

```
HTTP/1.1 204 No Content
x-oss-request-id: 534B372F74E88A4D89060124
```

```
Date: Mon, 14 Apr 2014 01:17:35 GMT
Connection: close
Content-Length: 0
Server: AliyunOSS
```

Object Operations

Put Object

Put Object 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	Specifies the web page caching behavior when an object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Disposition	Specifies the name of an object when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Encoding	Specifies the content encoding format when an 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 computed 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 Constraint:none
Expires	Specifies the expiration time in milliseconds. For details, refer to RFC2616. Type: Integer Default value:none
x-oss-server-side-encryption	Specifies the server-side encryption algorithm when the OSS creates an object. Type: String Valid value: AES256
x-oss-object-acl	Specifies the access permission when the OSS creates an object. Type: String Valid value: public-read, private, public-read-write

Response Headers

Name	Description
Content-Length	The length in bytes of the body in the response. Type: String Default: None
Connection	The connection status between the client and the OSS server. Type: Enum Valid Values: open,close,keep-alive Default: None
Content-MD5	This header is returned for the created Object. It is the based64-encoded 128-bit MD5 hash of the created Object. The client can use this header to perform an end-to-end integrity check. The Content-MD5 value returned is computed by the OSS whether this header is specified in PUT Object request or not. Type: String
Date	The date and time OSS server responded Type: String Default: None
ETag	The entity tag is a hash of the object. The

	ETag reflects changes only to the contents of an object, not its metadata. Type: String
Server	The name of the server that created the response. Type: String Default: AliyunOSS
x-oss-request-id	That unique id is created for identifying the request and troubleshooting Type: string

Detail Analysis

1. 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.
2. 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 an object, but the object size is equal to the size defined by Content-Length, and the remaining data is dropped.
3. If the file of an 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 returns the 200 OK message.
4. 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.
5. If the Content length parameter is not added to the header, the system returns the 411 Length Required message. The error code is MissingContentLength.
6. 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. The error code is RequestTimeout. At this time, the content of this file on the OSS is the data that you have uploaded.
7. If the bucket of the object to be added does not exist, the system returns the 404 Not Found message. The error code is NoSuchBucket.
8. If you have no permission to access the bucket of the object to be added, the system returns the 403 Forbidden message. The error code is AccessDenied.
9. If the length of the added file exceeds 5 GB, the system returns the 400 Bad Request message. The error code is InvalidArgument.
10. If the length of the received object key exceeds 1023 bits, the system returns the 400 Bad Request message. The error code is InvalidObjectName.
11. When you put an object, the OSS supports the following four header fields defined in RFC2616: Cache-Control, Expires, Content-Encoding, and Content-Disposition. 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.

12. 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 returns the 400 message and the error code: InvalidEncryptionAlgorithmError. After this header is specified, the response header also contains this header, and the OSS stores the encryption algorithm of the uploaded object. When this object is downloaded, the response header contains x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this object.

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]
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 61d2042d-1b68-6708-5906-33d81921362e
Date: Fri, 24 Feb 2012 06:03:28 GMT
ETag: "7DCA4FDCA3F27655940C866D52B04C39"
Content-MD5: fcpP3KPydlWUDIZtUrBMOQ==
Connection: close
Content-Length: 0
Server: AliyunOSS
```

Copy Object

Copy Object is used 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 Object operation, and directly performs this operation on the server side. If the Copy Object operation is successful, the system returns new object information to you.This operation is applicable to an object smaller than 1 GB. To copy an object greater than 1 GB, you must use the Multipart Upload operation. For details about this

operation, refer to Upload Part Copy.

Request Syntax

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

Request Header

Name	Description
x-oss-copy-source	Specifies 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 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 received parameter is the same as or later than the modification time of the file, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-modified-since	If the source object has 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-metadata-directive	Valid values include COPY and REPLACE. If

	this parameter is set to COPY, the system copies meta for the new object from the source object. If this parameter is set to REPLACE, the system ignores all meta values of the source object, and uses the meta value specified in this request. If this parameter is set to a value other than COPY and REPLACE, the system returns the 400 Bad Request message. Note that when this parameter is set to COPY, the system does not copy the meta value of x-oss-server-side-encryption of the source object. Type: string Default value: COPY Valid value: COPY, REPLACE
x-oss-server-side-encryption	Specifies the server-side encryption algorithm when the OSS creates the target object. Type: String Valid value: AES256
x-oss-object-acl	Specifies the access permission when the OSS creates an object. Type: String Valid value: public-read, private, 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

1. You can use the Copy Object operation to modify the meta information of an existing object.
2. If the source object address is the same as the target object address in the Copy Object operation, the system directly replace the meta information in the source object regardless of the value of x-oss-metadata-directive.
3. The OSS allows the Copy Object request to contain any number of the four pre-judgment

headers. For details about the related logic, refer to Detail Analysis of Get Object.

4. To complete a Copy Object operation, the requester must have the permission to read the source object.
5. The source object and the target object must belong to the same data center; otherwise, the system returns the 403 AccessDenied message, and the error information is: Target object does not reside in the same data center as source object.
6. In the billing statistics of the Copy Object operation, the number of Get requests increases by 1 in the bucket of the source object, the number of Put requests increases by 1 in the bucket of the target object, and a storage space is added accordingly.
7. In the Copy Object operation, all relevant request headers start from x-oss-, and therefore must be added to the signature string.
8. If the x-oss-server-side-encryption header is specified in the Copy Object request, and its value (AES256) is valid, the target object will be encrypted on the server side after the Copy Object operation is performed no matter whether the source object has been encrypted on the server side. In addition, the Copy Object 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 also contains 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 Object operation, the target object is the data that is not encrypted on the server side no matter whether the source object has been encrypted on the server side.
9. When the x-oss-metadata-directive header in the Copy Object request is set to COPY (default value), the system does not copy the x-oss-server-side-encryption value of the source object. That is, the target object is encrypted on the server side only when x-oss-server-side-encryption is specified accordingly in the Copy Object request.
10. If the x-oss-server-side-encryption header is specified in the Copy Object request and its value is not AES256, the system returns the 400 message and the error code: InvalidEncryptionAlgorithmError.
11. If the size of the file to be copied is greater than 1 GB, the system returns the 400 message and the error code: EntityTooLarge.

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=
```

Return example:

```
```

```
HTTP/1.1 200 OK
x-oss-request-id: 3dfb2597-72a0-b3f7-320f-8b6627a96e68
Content-Type: application/xml
Content-Length: 193
Connection: close
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

Get Object is used to obtain an object which you must have the permission to read.

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

When sending a GET request, you can customize the following headers in the OSS returned request:

Name	Description
response-content-type	Specifies the content-type header in the OSS returned request. Type:string Default value: none
response-content-language	Specifies the content-language header in the OSS returned request. Type:string Default value: none
response-expires	Specifies the expires header in the OSS returned request. Type:string Default value: none
response-cache-control	Specifies the cache-control header in the OSS returned request.

	Type:string Default value: none
response-content-disposition	Specifies the content-disposition header in the OSS returned request. Type:string Default value: none
response-content-encoding	Specifies the content-encoding header in the OSS returned request. Type:string Default value: none

Request Header

Name	Description
Range	Specifies the range of file transfer. For example, if Range is set to bytes=0-9, the system transfers 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
If-Unmodified-Since	If the time specified by the received parameter is the same or later than the modification time of the file, the system transfers the file normally, and returns the 200 OK message; 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 transfers the file normally, and returns the 200 OK message; 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 transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 304 Not Modified message. Type: string Default value: none

Response Headers

Name	Description
Content-Length	The length in bytes of the body in the response. Type: String Default: None
Content-Type	The MIME content type of the request content in the body. Type: String Default: None
Connection	The connection status between the client and the OSS server. Type: Enum Valid Values: open,close,keep-alive Default: None
Content-MD5	This header is returned when client reads an Object created by PUT Object operation as a whole. It is a based64-encoded 128-bit MD5 digest of the created Object. This header is returned for the client to verify the integrity of the data of the entire Object during transport. This header is not returned for a Range GET. Type: String
Date	The date and time OSS server responded Type: String Default: None
ETag	The entity tag is a hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. Type: String
Server	The name of the server that created the response. Type: String Default: AliyunOSS
x-oss-request-id	That unique id is created for identifying the request and troubleshooting Type: string

Detail Analysis

1. The Range parameter in the Get Object request can be set to support resumable data transfer from breakpoints. This function is recommended if the object size is large.
2. If the Range parameter is specified in the request header, the returned message contains 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 transfers the entire file, and does not mention Content-Range in the result.
3. If the time specified by If-Modified-Since does not match the actual modification time, the system directly returns the file, as well as the 200 OK message.
4. If-Modified-Since can coexist with If-Unmodified-Since. If-Match can also coexist with If-None-Match.
5. 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 returns the 412 Precondition Failed message.
6. 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 returns the 304 Not Modified message.
7. If the file does not exist, the system returns the 404 Not Found message. The error code is NoSuchKey.
8. The OSS does not allow you to customize the headers in the OSS returned request using request parameters in the GET Object request during an anonymous access.
9. When you customize some headers in the OSS returned request, the OSS sets these headers to the values specified by parameters in the GET Object Request only when the request is successfully processed, that is, when the system returns the 200 OK message.
10. If this object is encrypted on the server side, the system automatically returns the decrypted object on receiving the GET Object request, and returns 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.
11. 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 determines 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 does 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.

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=
```

Return 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: "7DCA4FDCA3F27655940C866D52B04C39"
Content-MD5: fcpP3KPydlWUDIZtUrBMOQ==
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=
```

Return 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 the returned message header customized:

```
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
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 1144d124-055c-4052-2c65-a1e3439d41c1
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: "7DCA4FDCA3F27655940C866D52B04C39"
```

```
Content-MD5: fcpP3KPydlWUDIZtUrBMOQ==
Content-Length: 344606
Connection: close
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]
```

Append Object

Append Object is used to upload files in appending mode. The type of the objects created with the Append Object operation is Appendable Object, and 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	Specifies the web page caching behavior when an object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Disposition	Specifies the name of an object when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Encoding	Specifies the content encoding format when an 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 computed 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 Constraint:none
Expires	Specifies the expiration time in milliseconds. For details, refer to RFC2616. Type: Integer Default value:none
x-oss-server-side-encryption	Specifies the server-side encryption algorithm when the OSS creates an object. Type: String Valid value: AES256
x-oss-object-acl	Specifies the access permission when the OSS creates an object. Type: String Valid value: public-read, private, public-read-write

Response Header

Name	Description
x-oss-next-append-position	Specifies 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-digit integer
x-oss-hash-crc64ecma	Specifies the 64-bit CRC value of the object. This 64-bit CRC is computed according to ECMA-182. Type: 64-digit integer

Association with Other Operations

1. 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 returns the 409 message and the error code ObjectNotAppendable.
2. If you perform the Put Object operation on an existing appendable object, this appendable object is overwritten by the new object, and the type of this object is changed to Normal Object.
3. After the Head Object operation is performed, the system returns 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 also returns x-oss-next-append-position and x-oss-hash-crc64ecma.
4. In the response XML of the Get Bucket (List Objects) request, the type of an appendable object is set to Appendable.
5. You can neither use Copy Object to copy an appendable object, nor change the server-side encryption attribute of this object. You can, however, use Copy Object to change the customized metadata.

Detail Analysis:

1. The two URL parameters, append and position, are both CanonicalizedResource, and must be contained in the signature.
2. URL parameters must also contain append, which specifies that the operation is an Append Object operation.
3. URL query parameters must contain position, which specifies the position from where 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.
4. If the position value is different from the current object length, the OSS returns the 409 message and the error code 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.
5. 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 is successful; otherwise, the system regards that the position and object length are mismatched.
6. If the position value is 0 and an object with the same name does not exist, headers (such as x-oss-server-side-encryption) can be set in the Append Object request like the Put Object request. This is the same as 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. Later, if meta needs to be modified, you can use the Copy Object request.

7. Due to the concurrency, even if you set the value of position to x-oss-next-append-position, this request can still fails due to PositionNotEqualToLength.
8. The length limit of an object generated by Append Object is the same as that of an object generated by Put Object.
9. After each Append Object operation, the last modification time of this object will be updated.
10. If the position value is correct and the content with the length of 0 is appended to an existing appendable object, this operation does not change the status of the object.

CRC64 Computing Method

The CRC of an appendable object is computed according to ECMA-182. Its computing method is the same as that of XZ. CRC64 can be computed 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 computed 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]
```

Return example:

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

Delete Object

Delete Object 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

1. To delete an object with Delete Object, you must have the permission to write this object.
2. If the object to be deleted does not exist, the OSS returns the 204 No Content message.
3. If the bucket of the object does not exist, the system returns the 404 Not Found message.

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=
```

Return example:

```
HTTP/1.1 204 NoContent
```

```
x-oss-request-id: 1a61ecd1-5de8-4e2e-20b5-c66e135bc379
Date: Fri, 24 Feb 2012 07:45:28 GMT
Content-Length: 0
Connection: close
Server: AliyunOSS
```

Delete Multiple Objects

Delete Multiple Objects allows you to delete multiple objects in the same bucket with one HTTP request. You can delete up to 1,000 objects with one Delete Multiple Objects operation, and two return modes are provided: verbose and quiet.

- Verbose: The message body returned by the OSS contains the deletion result of each object.
- Quiet: The message body returned by the OSS contains only the results of those objects that encounter deletion errors. If all the objects are successfully deleted, the OSS does not return any 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 Elements

Name	Description
Delete	Specifies the container that saves the Delete Multiple Objects request. Type: Container Subnode: One or more object elements and the optional quiet element Parent node: None.
Key	Specifies the name of the object to be

	deleted. Type: String Parent node: Object
Object	Specifies the container that saves the information about the object. Type: Container Subnode: Key Parent node: Delete.
Quiet	Enables or disables the quiet mode. Type: Enumerated string Valid value: True, False Default value: False Parent node: Delete
encoding-type	Specifies the encoding of the returned content and the encoding type.The object key uses UTF-8 characters, 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 object key contains control characters not supported by the xml 1.0 standard, you can specify encoding-type to encode the returned object key.

Data type: String
Default value: None|

Response Elements

Name	Description
Deleted	Specifies the container that saves the successfully deleted objects. Type: Container Subnode: Key Parent node: DeleteResult
DeleteResult	Specifies the container that saves the result of the Delete Multiple Objects request. Type: Container Subnode: Deleted Parent node:None
Key	Specifies the name of the object on which the OSS performs the Delete operation. Type: String Parent node: Deleted
encoding-type	Specifies the encoding type in the returned result. If encoding-type is specified in the request, the Key will be encoded in the returned result. Type: String Parent node: Container

Detail Analysis

1. The Content-Length and Content-MD5 fields must be specified in the Delete Multiple Objects request. The OSS verifies that the received message body is correct based on the two fields, and then performs the Delete operation.
2. Method for generating the content of the Content-MD5 field: Encrypt the Delete Multiple Objects request using MD5 to obtain a 128-byte array, and encode the array using Base64. The final string obtained is the content of the Content-MD5 field.
3. The return mode of the Delete Multiple Objects request is Verbose by default.
4. If the Delete Multiple Objects request is used to delete a non-existing object, the operation is still regarded as successful.
5. 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 returns the MalformedXML error code.
6. The Delete Multiple Objects request can be used to delete at most 1,000 objects at a time. If the number of objects to be deleted at a time exceeds 1,000, the system returns the MalformedXML error code.
7. 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

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>
```

Return 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: close
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>
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 501ad9bb-1383-771d-0ee9-59a810bd5fde
Date: Wed, 29 Feb 2012 12:33:45 GMT
Content-Length: 0
Connection: close
Server: AliyunOSS
```

Head Object

Head Object 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

Response Headers

Name	Description
Content-Length	The length in bytes of the body in the response. Type: String Default: None
Content-Type	The MIME content type of the request content in the body. Type: String Default: None
Connection	The connection status between the client and the OSS server. Type: Enum Valid Values: open,close,keep-alive Default: None
Content-MD5	This header is returned if the Object created by PUT Object operation. It is a based64-encoded 128-bit MD5 digest of the created Object. Type: String
Date	The date and time OSS server responded Type: String Default: None
ETag	The entity tag is a hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. Type: String
Server	The name of the server that created the response. Type: String Default: AliyunOSS
x-oss-request-id	That unique id is created for identifying the request and troubleshooting Type: String

Detail Analysis

1. 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.
2. 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 returns the 304 Not Modified message.
3. 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 is returned.
4. If the file does not exist, the system returns the 404 Not Found message.

5. 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.

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=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 06d4be30-2216-9264-757a-8f8b19b254bb
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: "7DCA4FDCA3F27655940C866D52B04C39"
Content-MD5: fcpP3KPydlWUDIZtUrBMOQ==
Content-Length: 344606
Content-Type: image/jpg
Connection: close
Server: AliyunOSS
```

Put Object ACL

Put Object ACL is used to modify the access permission of an object. Currently, three access permissions are available for an object, including 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 is returned; otherwise, the corresponding error code and prompt message are 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	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	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	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.

Detail Analysis:

1. Read operations of an object include reading the source object in the GetObject, HeadObject, CopyObject, and UploadPartCopy operations. Write operations of an object include writing a new object in the PutObject, PostObject, AppendObject, DeleteObject, DeleteMultipleObjects, CompleteMultipartUpload, and CopyObject operations.
2. x-oss-object-acl must be set to one of the preceding three permissions; otherwise, the OSS returns the 400 Bad Request message and the error code: InvalidArgument.
3. You can use Put Object 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 Put Object ACL. For example, if the header of the Put Object request carries x-oss-object-acl, you can set the ACL of an object while uploading the object.
4. If a user who has no permission to read an object attempts to read this object, the OSS returns the 403 Forbidden message. The error code is AccessDenied. The prompt displayed is You do not have read permission on this object.
5. If a user who has no permission to write an object attempts to write this object, the OSS returns the 403 Forbidden message. The error code is AccessDenied. The prompt displayed is You do not have write permission on this object.
6. Only the owner of a bucket has the permission to use Put Object ACL to modify the ACL for an object in this bucket. If a user who is not the bucket owner attempts to use Put Object ACL, the OSS returns the 403 Forbidden message. The error code is AccessDenied. The prompt displayed is You do not have write acl permission on this object.
7. 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: Wen, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:KU5h8YMUC78M30dXqf3JxrTZHiA=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 248c6483-2a95-622e-3022-ebe65d8aad5f
Date: Wen, 29 Apr 2015 05:21:12 GMT
Content-Length: 0
Connection: close
Server: AliyunOSS
```

Get Object ACL

Get Object ACL 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	Specifies the container that stores the Get Object ACL result. Type:Container Parent node:None

DisplayName	Name of the bucket owner. (Currently, the name is the same as the bucket owner ID.) Type: String Parent node:AccessControlPolicy.Owner
Grant	Specifies the ACL permission of an object. Type: Enumerated string Valid value: 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 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 returns the 403 Forbidden message. The error code is AccessDenied. The prompt displayed is You do not have read acl permission on this object.
2. 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: Wen, 29 Apr 2015 05:21:12 GMT
Authorization: OSS qn6qrrqxo2oawuk53otfjbyc:CTkuxpLAi4XZ+WwIfNm0FmgbrQ0=
```

Return example:

```
HTTP/1.1 200 OK
x-oss-request-id: 6f720c98-40fe-6de0-047b-e7fb08c4059b
Date: Wen, 29 Apr 2015 05:21:12 GMT
Content-Length: 253
```

```
Content-Tupe: application/xml
Connection: close
Server: AliyunOSS

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

Post Object

Post Object is used to upload an object 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	Specifies the access key ID of the bucket owner. Type: string Default value: None Constraint: This form field is required when the bucket does not allow public-read-write, or when the Policy (or Signature) form field is provided.	Conditional
policy	Specifies 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 Constraint: This form field is required when the bucket	Conditional

	does not allow public-read-write, or when the OSSAccessKeyId (or Signature) form field is provided.	
Signature	Specifies 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 Constraint: 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	Specifies 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	Specifies the object name of 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	Required

	set to /user/a/${filename}, the final object name is /user/a/b.jpg. Type: string Default value: None	
success_action_redirect	Specifies 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 returns an error code, and the client is not redirected to any URL. Type: string Default value: None	Optional
success_action_status	Specifies 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-*	Specifies the user meta value set by the user. The OSS does not check or use this value. Type: string Default value: None	Optional
x-oss-server-side-encryption	Specifies the server-side encryption algorithm when the OSS creates an object. Type: String Valid value: AES256	Optional
x-oss-object-acl	Specifies the access permission when the OSS creates an object. Type: String Valid value: public-read, private, public-read-write	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	Specifies the container that saves the result of the Post Object request. Type: Container Subnode: Bucket, ETag, Key, Location
Bucket	Specifies the bucket name. Type: String Parent node: PostResponse
ETag	Specifies the entity tag (ETag), which is created when an object is generated. For an object created by Post Object, the ETag value is the MD5 value of the object, and can be used to check whether the content of the object has changed. Type: String Parent node: PostResponse
Location	Specifies the URL of the newly created object. Type: String Parent node: PostResponse

Detail Analysis

1. 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 Access Key Secret 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.
2. 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 returns the error code: InvalidArgument.
3. 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.

4. 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.
5. The policy sets forth 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 returns the error code: AccessDenied. When checking validity of the policy, the OSS does not check irrelevant form fields in the policy.
6. The form and policy must be encoded with UTF-8. The policy is a JSON text encoded with UTF-8 and Base64.
7. The Post Object request can contain extra form fields. The OSS checks validity of these form fields based on the policy.
8. 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.
9. If the Post Object request contains the Header signature or URL signature, the OSS does not check these signatures.
10. 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.
11. The total length of the body in the Post Object request cannot exceed 5 GB. If the file size is too large, the system returns the error code: EntityTooLarge.
12. If you upload a request in which the x-oss-server-side-encryption header is specified, the value of this header must be set to AES256; otherwise, the system returns the 400 message and the error code: InvalidEncryptionAlgorithmError. After this header is specified, the response header also contains this header, and the OSS stores the encryption algorithm of the uploaded object. When this object is downloaded, the response header contains x-oss-server-side-encryption, the value of which is set to the encryption algorithm of this object.
13. 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--
```

Return 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: close
Content-Length: 0
Server: AliyunOSS
```

Post Policy

The policy form field in the Post Object request is used to verify 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 when a bucket that allows public-read-write is

uploaded, it is strongly recommended that this form field be used to limit the Post Object request.

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, and is expressed in ISO601 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	Specifies 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	Specifies the object name of the uploaded file. This condition supports the exact match and starts-with match modes.
success_action_redirect	Specifies 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	Specifies 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-*	Specifies 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	Specifies 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 Character	Description
\/	Slash
\	Backslash
\”	Double quotes
\$	Dollar sign
\b	Blank
\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 policy encoded with UTF-8.
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 Access Key Secret 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.

Multipart Upload Operations

Introduction to Multipart Upload

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 an object 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 Header

Name	Description
Cache-Control	Specifies the web page caching behavior when an object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Disposition	Specifies the name of an object when the object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Content-Encoding	Specifies the content encoding format when an object is downloaded. For details, refer to RFC2616. Type: string Default value: none
Expires	Specifies the expiration time in milliseconds. For details, refer to RFC2616. Type: Integer Default value:none
x-oss-server-side-encryption	Specifies 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	Container used for storing the result of the Initiate Multipart Upload request. Type: Container Subnode: 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

Detail Analysis

1. When using this operation to calculate the authentication signature, you need to add "?uploads" to "CanonicalizedResource".
2. The Initiate Multipart Upload request supports the following standard HTTP request headers: Cache-Control, Content-Disposition, Content-Encoding, Content-Type, Expires, and user-defined headers prefixed with "x-oss-meta-". For the specific meanings of these headers, refer to the PUT Object interface.
3. The Initiate Multipart Upload request does not affect an existing object with the same name.
4. 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.
5. 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 them 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 error 400 and the corresponding error prompt: 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 error 400 and the corresponding error prompt: 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=
```

Return example:

```
HTTP/1.1 200 OK
Content-Length: 230
Server: AliyunOSS
Connection: close
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

1. 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.
2. In the Multipart Upload mode, besides 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.

3. The OSS puts the MD5 value of the part of data received by the server in the ETag header and returns it to the user.
4. The part number ranges from 1 to 10,000. If the part number exceeds this range, the OSS will return the InvalidArgument error code.
5. 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.
6. 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: close
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 and 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 an object larger than 1 GB, you must use the Upload Part Copy method. If you want to copy an object 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	Specifies 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 transfers 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 received parameter is the same as or later than the modification time of the file, the system

	transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-modified-since	If the source object has 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

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 received parameter is the same as or later than the modification time of the file, the system transfers the file normally, and returns the 200 OK message; otherwise, the system returns the 412 Precondition Failed message. Type: string Default value: none
x-oss-copy-source-if-modified-since	If the source object has 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

Detail Analysis

1. 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.
2. In the Multipart Upload mode, besides 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.
3. 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.
4. 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.

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: close
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 Elements

Name	Description
CompleteMultipartUpload	Container used for storing the content of the Complete Multipart Upload request. Type: Container Subnode: 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	Container used for storing uploaded data parts. Type: Container Subnode: ETag, PartNumber Parent node:InitiateMultipartUploadResult
PartNumber	Number of parts. Type: Integer Parent node:Part

Response Elements

Name	Description
Bucket	Specifies the bucket name. Type: String Parent node: CompleteMultipartUploadResult
CompleteMultipartUploadResult	Container used for storing the result of the Complete Multipart Upload request. Type: Container Subnode: 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	Specifies the URL of the newly created object. Type: String Parent node: CompleteMultipartUploadResult
Key	Name of the newly created object. Type: String Parent node: CompleteMultipartUploadResult

Detail Analysis

1. 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.
2. 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.
3. In the part list submitted by the user, part numbers can be discontinuous. For example, the first part number is 1 and the second part number is 5.
4. After the OSS successfully processes the Complete Multipart Upload request, the corresponding Upload ID will become invalid.
5. The same object may have different Upload IDs. When an Upload ID is completed, other Upload IDs of this object are not affected.
6. 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.

7. 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

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: close
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

1. 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.
2. If the entered Upload ID does not exist, the OSS returns 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: close
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	The maximum number of Multipart Upload events returned for one request. If not specified, the default value is 1000. The max-uploads value cannot exceed 1000. 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 contains: Multipart events 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 contains: Multipart events in which the lexicographic orders of all object names are greater than the value of the key-marker parameter and all Multipart Upload events in which the object name is the same as the value of the key-marker parameter but the Upload ID is greater than the value of the upload-id-marker parameter.

	Type: String
prefix	to 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 contains:Multipart events in which the lexicographic orders of all object names are greater than the value of the key-marker parameter and all Multipart Upload events in which the object name is the same as the value of the key-marker parameter but the Upload ID is greater than the value of the upload-id-marker parameter. Type: String
encoding-type	Specifies the encoding of the returned content and the encoding type.The object key uses UTF-8 characters, 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 object key contains control characters not supported by the xml 1.0 standard, you can specify encoding-type to encode the returned object key. Data type:String Default value: None

Response Elements

Name	Description
ListMultipartUploadsResult	Container used for storing the result of the List Multipart Upload request. Type: Container Subnode: Bucket, KeyMarker, UploadIdMarker, NextKeyMarker, NextUploadIdMarker, MasUploads, Delimiter, Prefix, CommonPrefixes, IsTruncated, Upload Parent node:None
Bucket	Specifies the bucket name. Type: String Parent node: ListMultipartUploadsResult
EncodingType	Specifies the encoding type in the returned result. 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	Specifies whether the returned Multipart Upload result list is truncated. "true" indicates that not all results are returned; "false" indicates that all results are returned. Type: Enumerated string Valid values: true and false Default value:False Parent node: ListMultipartUploadsResult
Upload	Container used for storing the information about the Multipart Upload event. Type: Container Subnode: 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

1. The maximum value of the "max-uploads" parameter is 1000.
2. 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.
3. Using the prefix parameter, you can flexibly manage objects in a bucket in groups (similar to the folder function).
4. 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=
```

Return example:

```
HTTP/1.1 200
Server: AliyunOSS
Connection: close
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 List Parts command can be used to list all successfully uploaded parts under 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	Specifies the encoding of the returned content and the encoding type.The object key uses UTF-8 characters, 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 object key contains control characters not supported by the xml 1.0 standard, you can specify encoding-type to encode the returned

	object key. Data type:String Default value: None

Response Elements

Name	Description
ListPartsResult	Container used for storing the result of the List Parts request. Type: Container Subnode: Bucket, Key, UploadId, PartNumberMarker, NextPartNumberMarker, MaxParts, IsTruncated, Part Parent node:none
Bucket	Specifies the bucket name. Type: String Parent node: ListPartsResult
EncodingType	Specifies the encoding type in the returned result. If 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. "true" indicates that not all results are returned; "false" indicates that all results are returned. Type: Enumerated string Valid values: true

	and false Parent node:ListPartsResult
Part	Container used for storing part information. Type: String Subnode: 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

1. List Parts supports two request parameters: max-parts and part-number-marker.
2. The maximum value of the max-parts parameter is 1000; its default value is also 1000.
3. The results returned by the OSS are listed in ascending order based on the part numbers.
4. 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=
```

Return example:

```
HTTP/1.1 200
Server: AliyunOSS
Connection: close
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>&quot;3349DC700140D7F86A078484278075A9&quot;</ETag>
      <Size>6291456</Size>
   </Part>
   <Part>
      <PartNumber>2</PartNumber>
      <LastModified>2012-02-23T07:01:12.000Z</LastModified>
      <ETag>&quot;3349DC700140D7F86A078484278075A9&quot;</ETag>
      <Size>6291456</Size>
   </Part>
   <Part>
      <PartNumber>5</PartNumber>
      <LastModified>2012-02-23T07:02:03.000Z</LastModified>
      <ETag>&quot;7265F4D211B56873A381D321F586E4A9&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	Yes

	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	Yes	
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	No	
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:	No	

	CORSRule		
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
```

OSS Error Response

If an error occurs when a user accesses the OSS, the OSS returns the error code and error information for the user to locate the problem and handle it properly.

OSS Error Response Format

If an error occurs when the user accesses the OSS, the OSS returns an HTTP status code 3xx, 4xx, or 5xx and a message body in application/xml format.

Example of an error response message body:

```
<?xml version="1.0" ?>
<Error xmlns=" http://doc.oss-cn-hangzhou.aliyuncs.com" >
<Code>
AccessDenied
</Code>
<Message>
Query-string authentication requires the Signature, Expires and OSSAccessKeyId parameters
</Message>
<RequestId>
1D842BC5425544BB
</RequestId>
<HostId>
oss-cn-hangzhou.aliyuncs.com
</HostId>
</Error>
```

All error message bodies include the following elements:

- Code: indicates an error code returned by the OSS to a user.
- Message: indicates the detailed error information provided by the OSS.
- RequestId: indicates a UUID of the request. If you cannot solve the problem, you can seek help from OSS development engineers by providing this RequestId.
- HostId: used to identify the accessed OSS cluster (which is uniformly oss-cn-hangzhou.aliyuncs.com at present).

For special error information elements, see specific request descriptions.

OSS Error Codes

The following table lists the OSS error codes:

Error Code	Description	HTTP Status Code
AccessDenied	The access is denied	403
BucketAlreadyExists	The bucket already exists	409
BucketNotEmpty	The bucket is not empty	409
EntityTooLarge	The entity is too large	400
EntityTooSmall	The entity is too small	400
FileGroupTooLarge	The file group is too large	400
InvalidLinkName	The object link is the same as the linked object name	400
LinkPartNotExist	The object to which the object link is linked does not exist.	400
ObjectLinkTooLarge	There are too many objects in the object link	400
FieldItemTooLong	The table field in the post request is too large	400
FilePartInterity	The file part has been changed	400
FilePartNotExist	The file part does not exist	400
FilePartStale	The file part has expired	400
IncorrectNumberOfFilesInPOSTRequest	The number of files in the post request is incorrect	400
InvalidArgument	The parameter format is invalid	400
InvalidAccessKeyId	The Access Key ID does not exist	403

InvalidBucketName	The bucket name is invalid	400
InvalidDigest	The digest is invalid	400
InvalidEncryptionAlgorithmError	The specified entropy encryption algorithm is invalid	400
InvalidObjectName	The object name is invalid	400
InvalidPart	The part is invalid	400
InvalidPartOrder	The part order is invalid	400
InvalidPolicyDocument	The policy document is invalid	400
InvalidTargetBucketForLogging	An invalid target bucket exists in the logging operation	400
InternalError	An error occurs in the OSS	500
MalformedXML	The XML format is invalid	400
MalformedPOSTRequest	The format of the body in the post request is invalid	400
MaxPOSTPreDataLengthExceededError	The body, except the file content, uploaded in the post request is too large	400
MethodNotAllowed	The method is not supported	405
MissingArgument	A parameter is missing	411
MissingContentLength	Then content length is missing	411
NoSuchBucket	The bucket does not exist	404
NoSuchKey	The file does not exist	404
NoSuchUpload	The multipart upload ID does not exist	404
NotImplemented	The method cannot be processed	501
PreconditionFailed	The preconditioning fails	412
RequestTimeTooSkewed	The request initiation time exceeds the server time by 15 minutes	403
RequestTimeout	The request times out	400
RequestIsNotMultiPartContent	The content type of the post request is invalid	400
SignatureDoesNotMatch	The signature is incorrect	403
TooManyBuckets	The user has too many buckets	400

InvalidEncryptionAlgorithmError	The specified entropy encryption algorithm is invalid	400

Operations Not Supported by the OSS

If an operation not supported by the OSS is used to access a resource, the OSS returns error 405 Method Not Allowed.

Example of an invalid request:

```
ABC /1.txt HTTP/1.1
Host: bucketname.oss-cn-shanghai.aliyuncs.com
Date: Thu, 11 Aug 2016 03:53:40 GMT
Authorization: signatureValue
```

Return example:

```
HTTP/1.1 405 Method Not Allowed
Server: AliyunOSS
Date: Thu, 11 Aug 2016 03:53:44 GMT
Content-Type: application/xml
Content-Length: 338
Connection: keep-alive
x-oss-request-id: 57ABF6C8BC4D25D86CBA5ADE
Allow: GET DELETE HEAD PUT POST OPTIONS
<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>MethodNotAllowed</Code>
<Message>The specified method is not allowed against this resource.</Message>
<RequestId>57ABF6C8BC4D25D86CBA5ADE</RequestId>
<HostId>bucketname.oss-cn-shanghai.aliyuncs.com</HostId>
<Method>abc</Method>
<ResourceType>Bucket</ResourceType>
</Error>
```

Note: If the accessed resource is /bucket/, ResourceType should be bucket; if the accessed resource is /bucket/object, ResourceType should be object.

Operations Supported by the OSS But Not Supported by Parameters

If parameters not supported by the OSS are added to an operation supported by the OSS (for example, an If-Modified-Since parameter is added to the PUT operation), the OSS returns error 400 Bad Request

Example of an invalid request:

```
PUT /abc.zip HTTP/1.1
Host: bucketname.oss-cn-shanghai.aliyuncs.com
Accept: */*
Date: Thu, 11 Aug 2016 01:44:50 GMT
If-Modified-Since: Thu, 11 Aug 2016 01:43:51 GMT
Content-Length: 363
```

Return example:

```
HTTP/1.1 400 Bad Request
Server: AliyunOSS
Date: Thu, 11 Aug 2016 01:44:54 GMT
Content-Type: application/xml
Content-Length: 322
Connection: keep-alive
x-oss-request-id: 57ABD896CCB80C366955187E
x-oss-server-time: 0
<?xml version="1.0" encoding="UTF-8"?>
<Error>
<Code>NotImplemented</Code>
<Message>A header you provided implies functionality that is not implemented.</Message>
<RequestId>57ABD896CCB80C366955187E</RequestId>
<HostId>bucketname.oss-cn-shanghai.aliyuncs.com</HostId>
<Header>If-Modified-Since</Header>
</Error>
```