

Cloud Object Storage

Agent Bucket



Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

The complete copyright of this document, including all text, data, images, and other content, is solely and exclusively owned by Tencent Cloud Computing (Beijing) Co., Ltd. ("Tencent Cloud"); Without prior explicit written permission from Tencent Cloud, no entity shall reproduce, modify, use, plagiarize, or disseminate the entire or partial content of this document in any form. Such actions constitute an infringement of Tencent Cloud's copyright, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Trademark Notice



This trademark and its related service trademarks are owned by Tencent Cloud Computing (Beijing) Co., Ltd. and its affiliated companies ("Tencent Cloud"). The trademarks of third parties mentioned in this document are the property of their respective owners under the applicable laws. Without the written permission of Tencent Cloud and the relevant trademark rights owners, no entity shall use, reproduce, modify, disseminate, or copy the trademarks as mentioned above in any way. Any such actions will constitute an infringement of Tencent Cloud's and the relevant owners' trademark rights, and Tencent Cloud will take legal measures to pursue liability under the applicable laws.

Service Notice

This document provides an overview of the as-is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre-sales consultation and technical after-sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Agent Bucket

Product Overview

COS Agent Bucket Advantages

Getting Started

API Documentation

Introduction

Intelligent Agent Bucket Enhancement API

Put Space Quota

Get Space Quota

Put Space Extension

Get Space Extension

Copy Object Recursive

Move Object

Move Object Recursive

Delete Object Recursive

Agent Bucket

Product Overview

Last updated: 2026-07-24 14:55:16

Introduction

Tencent Cloud Agent Bucket (COS Agent Bucket) is a new bucket type launched by Tencent Cloud for the AI Agent era. It is specifically designed to address the needs for "account storage" and "personal space" in scenarios such as AI Agents, SaaS-based online drives/photo albums, and B2B2C multi-user environments. Deeply integrated into the Tencent Cloud Object Storage (COS) system, it enables you to use isolated, user/instance-specific cloud space in the same way you use object storage, providing enterprises with large-scale, natively isolated agent storage and management capabilities.

Core Concepts

To understand Tencent Cloud Agent Bucket, you need to grasp the following core concepts:

- **Agent Bucket:** An Agent Bucket is a new type of bucket designed for AI Agent and multi-tenant scenarios. Building upon the features of a standard COS bucket, it introduces a new, independent logical layer called Space and online drive capabilities. Users can gain capabilities such as multi-user isolation, quota management, file sharing, instant upload with deduplication, and a recycle bin, without the need to configure or manage the underlying infrastructure.
- **Space:** A Space is a first-level logical isolation unit within an Agent Bucket, corresponding to business concepts such as Agent instances, end users, or shared drives. Each Space has independent storage quotas, permission policies, AccessTokens, and usage statistics. A single Agent Bucket can host over 100 millions of Space, and a single Space can store tens of billions of files. On the business side, "User Space / Instance Space / shared drives" can be directly mapped to the storage architecture hierarchy, eliminating the need to maintain multi-tenant tables and permission tables at the application layer.

Use Cases

Tencent Cloud Agent Bucket can be widely applied to the following AI Agent and multi-tenant storage scenarios:

- **AI Agent Cloud Space:** It allocates an independent space for each Agent instance or a shared space for multi-user projects, storing intermediate results from Agent operations, user-uploaded files, AI-generated content, and more. Spaces are isolated from each other to enable independent authorization. Administrators can adjust storage quotas and

entitlements for different Space as needed. After a Space is deleted, its data can be transferred to an administrator or moved to the recycle bin.

- **Online Drive/Photo Album Applications:** When enterprise customers provide personal storage services such as online drives and photo albums to their end users, they can create a Space for each end user by leveraging the capabilities of the Agent Bucket. Different quotas, speed limits, and feature usage limits can be configured for basic users and VIP users. End users can directly connect to the backend service using their own access keys to achieve permission isolation. On the business side, entitlements can be independently configured based on the Space dimension, facilitating commercial design and settlement. Furthermore, the Agent Bucket provides essential features for SaaS applications, such as sharing, instant upload, recycle bin, and directory management, eliminating the need for the business side to repeatedly develop these foundational capabilities.
- **Shared Space:** Within an enterprise, there are two types of needs: "public asset accumulation" and "employee private data." Enterprise knowledge bases, skill libraries, and plugin libraries each occupy independent public Space, with their content, permissions, and versions managed uniformly by administrators. Employees or Agent instances are allocated independent personal Space. When an employee leaves, the administrator can transfer or reclaim the data. Both types of Space are managed uniformly under the same Agent Bucket, with clear isolation between them.

Domain Name

Agent Bucket provides independent public network and private network access domain names, sharing the domain name namespace with COS Standard buckets.

- **Public network access domain name:** `<BucketName-APPID>.cos.<Region>.myqcloud.com`
- **Private network access domain name:** `<BucketName-APPID>.cos-internal.<Region>.tencentcos.cn`
- **Custom domain name:** It reuses the existing custom domain name capability of COS. For details, see [COS Custom Domain Name](#).

Note

- **BucketName-APPID:** The Agent Bucket name, which is automatically generated by the system.
- **Region:** The region where the Agent Bucket is located.

Supported Regions

The list of regions and their abbreviations currently supported by Agent Bucket is as follows. Other regions will be rolled out gradually.

Region	Region Abbreviation	Access Domain
Chengdu	ap-chengdu	• Public network: <BucketName-APPID>.cos.ap-chengdu.myqcloud.com • Private network: <BucketName-APPID>.cos-internal.ap-chengdu.tencentcos.cn

Use Limits

Limit	Specification
Number of Agent Buckets per Account	Follows the same rules as COS Bucket, with a default of 200.
Number of Space per Agent Bucket	10 million by default, can be increased up to 1 billion by contacting us .
Number of Files per Space	No limit
Bucket Name Naming	Bucket names are automatically generated by the system, are globally unique, and cannot be modified by users after creation.
Space Naming	Supports user customization, supports English letters and numbers, and is 1 to 32 characters in length.

COS Agent Bucket Advantages

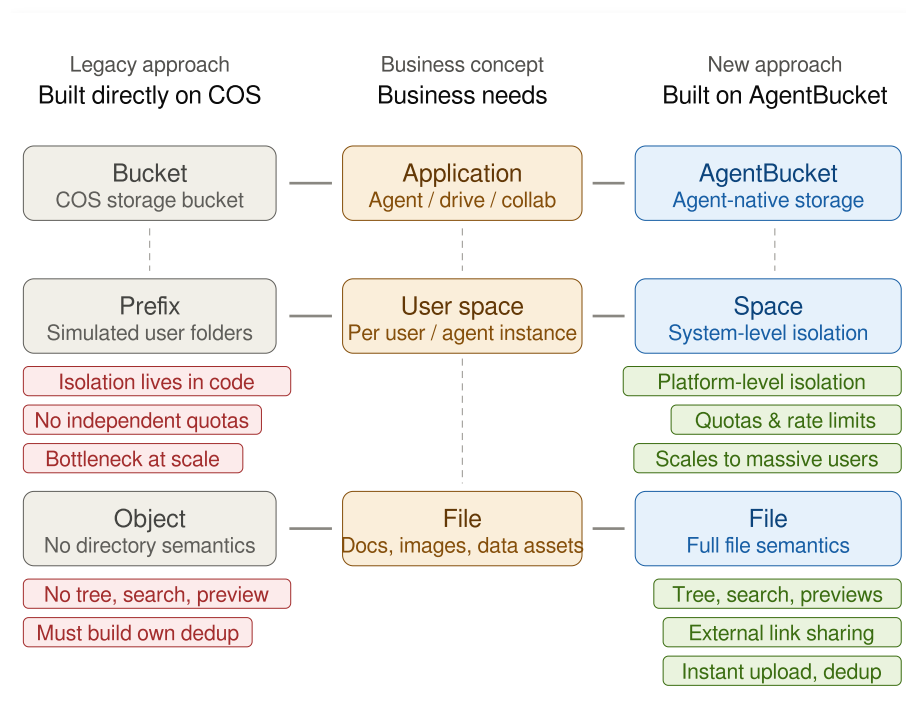
Last updated: 2026-07-17 14:57:26

Introduction

The Tencent Cloud COS Agent Bucket is a new bucket type introduced by Tencent Cloud Object Storage for the AI Agent era. It adds a Space, an independent logical partition, to achieve user/agent-level data isolation. This design adapts to storage setup scenarios for multi-user applications such as intelligent agent cloud spaces, personal online drives, and photo albums.

Architecture Strengths

Implementation: Each application corresponds to a dedicated Intelligent Agent Bucket. The bucket hosts a massive number of Spaces, with each Space bound to a user/agent. This enables independent quotas, permissions, and usage statistics.



User-Dedicated Space

Each agent or user has an independent Space architecture (an isolation unit) and can individually configure entitlements such as dedicated quotas and space speed limits. They can also create multi-user shared spaces for collaborative use.

100-Million Level User Support

It supports a user base of hundreds of millions by default. Object Keys are automatically distributed via hashing, which avoids hot shards at the architectural level and easily handles massive concurrent requests.

File System Semantics

It provides file system semantics and Mountpoint system mounting capabilities, meeting the needs of intelligent agents for workspace file management.

Content-Level Deduplication and Instant Upload

The built-in SHA-256 full-content hash calculation enables instant uploads and cross-user global deduplication, significantly improving upload efficiency and reducing storage costs.

Getting Started

Last updated: 2026-07-17 10:47:56

Introduction

The Tencent Cloud COS Agent Bucket, a new bucket type designed for AI Agent and multi-user scenarios, simplifies the process of building multi-user cloud spaces. This document describes the complete workflow from service activation to the first successful file upload and download. The workflow consists of four main steps: creating an Intelligent Agent Bucket, creating a space, uploading files, and viewing usage and downloading/sharing files.

Preparations

Enabling COS Service

In the [Tencent Cloud console](#), search for Object Storage and go to the COS console. Then, activate the COS service by following the on-screen instructions. If you have already activated the service, skip this step.

Enabling SMH Service

The underlying service of the Agent Bucket is powered by Tencent Cloud Smart Media Hosting (SMH). Before using it, go to the [SMH console](#) to activate the SMH service.

Granting Permissions to the Service Role

When you create an Agent Bucket for the first time, you must grant the service-linked role SMH_QCSLinkedRoleCOS access to your COS resources. Go to the [Role Management page](#) to complete the authorization.

Configuring the Python SDK

Some examples use the Python SDK to perform operations. Therefore, you must complete the environment configuration and preparation for the SDK in advance. The Agent Bucket uses the same SDK as COS. For details, see the [Quick Start](#) guide for the COS Python SDK.

Operation Steps

Step 1: Creating an Agent Bucket

We need to create a bucket for AI Agent or multi-tenant scenarios.

1. In the left sidebar of the [COS console](#), click **Bucket List** to go to the Bucket List page.

2. Select the **Agent Bucket List** tab, click **Create Agent Bucket**, and optionally add remarks for business identification, for example, `agent-bucket-demo`.
3. Click **Confirm** to complete the creation.

Step 2: Creating a Space

After an Agent Bucket is created, you must create a Space within the bucket to host data for a single Agent instance or end user.

1. In the left sidebar of the **COS console**, click **Bucket List** to go to the Bucket List page.
2. Select the **Agent Bucket List** tab, click the bucket you just created, and go to the bucket details page.
3. Click the **Space Management** tab to go to the Space List page.
4. Click **Create Space**, enter the following configuration information, and keep the default settings for other configurations.
 - **Space ID:** It supports letters and numbers and must be 1 to 32 characters in length.
 - **Quota:** Set the storage quota for this space, for example, 5GB. Leaving it blank indicates no quota limit.
 - **Remarks:** Optional. It is used for business identification, for example, `user-001` or `agent-instance-A`.
5. Click **Confirm Creation** to complete the creation.

Step 3: Uploading Files

The Agent Bucket is fully compatible with the COS protocol, and file paths are organized as `<SpaceID>/<ObjectKey>`. Currently, you can use the console, REST API, or COS SDK to upload files.

Note:

- **Console operation entry for uploading files to an Agent Bucket:** On the Space Details page in Space Management, click the **Upload File** button.
- **SpaceId:** It corresponds to the first-level directory of an ObjectKey. To create the corresponding SpaceId, you must first upload an empty first-level directory object using the PUT Object API.
- **ObjectKey:** The object path under the corresponding Space.

Using the Python SDK as an example, this section demonstrates how to upload files to a specified space. For descriptions of the field parameters of the `put_object` method, see the [PUT Object API](#) documentation.

```
# -*- coding=utf-8 -*-
import os
from qcloud_cos import CosConfig
from qcloud_cos import CosS3Client

# 1. Set User Attributes
secret_id = os.getenv("COS_SECRET_ID")      # The user's SecretId
secret_key = os.getenv("COS_SECRET_KEY")     # The user's SecretKey
region = 'ap-chengdu'                       # The region where the Agent
Bucket is located
scheme = 'https'

config = CosConfig(
    Region=region,
    SecretId=secret_id,
    SecretKey=secret_key,
    Scheme=scheme,
)
client = CosS3Client(config)

# 2. Upload Files to a Specified Space
# The Key format is <SpaceID>/<ObjectKey>
# Body is the object content
response = client.put_object(
    Bucket='smh3km4897kxxxxxx-12596xxxxx',
    Key='sp-001abc/avatar.png',
    Body=open('./avatar.png', 'rb'),
)
print(response['ETag'])
```

Step 4: Viewing Usage and Downloading Shares

After the file upload is completed, you can view space usage, download files, or generate shareable links in the console or via the SDK.

Viewing Usage

Console operation instructions are as follows:

Go to the Overview tab on the bucket details page to view the relevant data:

- **Total Spaces:** The total number of spaces under the current bucket.

- **Total Storage:** Total storage of all spaces.
- **Total Files:** Total number of files in all spaces.
- **Basic Information:** Name, region, access domain, and other information.

Downloading Files for Sharing

The following example uses the Python SDK to demonstrate how to download files from a specified space and obtain shareable links.

```
# -*- coding=utf-8 -*-
import os
from qcloud_cos import CosConfig
from qcloud_cos import CosS3Client

secret_id = os.getenv("COS_SECRET_ID")
secret_key = os.getenv("COS_SECRET_KEY")
region = 'ap-chengdu'
scheme = 'https'

config = CosConfig(
    Region=region,
    SecretId=secret_id,
    SecretKey=secret_key,
    Scheme=scheme,
)
client = CosS3Client(config)

# Download Files to the Local Machine
response = client.get_object(
    Bucket='smh3km4897kxxxxx-12596xxxxx',
    Key='sp-001abc/avatar.png',
)
response['Body'].get_stream_to_file('./avatar-downloaded.png')
```

API Documentation

Introduction

Last updated: 2026-07-17 10:49:33

The Tencent Cloud COS Agent Bucket is designed to use the XML API, a lightweight, stateless interface. By calling this API, you can send requests and receive responses directly over HTTP/HTTPS to interact with the backend of the Tencent Cloud Agent Bucket.

Note:

- If you use the Tencent Cloud Agent Bucket API, you acknowledge that you have read and agreed to the [Tencent Cloud Service Agreement](#) and the [Tencent Cloud Object Storage Service Level Agreement](#).
- For details on the available regions for the Agent Bucket, refer to the [Product Overview](#) document.
- Before you make requests using the API, we recommend that you read the [Product Overview](#) document to understand information such as the domain names for initiating access, security authentication concepts, and public/private network access checks.

Terminology Information

When using the API, you will encounter some key concepts and terms. Refer to the table below:

Term	Description
APPID	The unique resource identifier at the user level that a developer possesses when accessing COS services, used to identify resources. It can be obtained on the API Key Management page.
SecretId	The project identity ID owned by a developer, used for identity authentication. It can be obtained on the API Key Management page.
SecretKey	The project identity key owned by a developer. It can be obtained on the API Key Management page.
Agent Bucket	A bucket for storing agent data. For more information about the agent bucket, see the Product Overview document.

BucketName-APPID	The naming format for the Agent Bucket. When you use the console or API, the system automatically generates the bucket name. For example, smhxxxxxxxxxxxxx-1250000000 indicates that the agent bucket smhxxxxxxxxxxxxx belongs to the user with APPID 1250000000.
Region	The abbreviation for a region. For currently available regions, see the Product Overview document.

Intelligent Agent Bucket Enhancement API

Put Space Quota

Last updated: 2026-07-17 10:50:00

Overview

Set the storage quota for a specified Space.

Request

Request Syntax

```
PUT /<SpaceId>/?space-quota HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Content-Type: application/xml
Content-Length: Content Length
Authorization: Auth String

[Request Body]
```

Note:

- **SpaceId:** It corresponds to the first-level directory of an ObjectKey. To create the corresponding SpaceId, you must first upload an empty first-level directory object using the PUT Object API.
- **Host:** <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- **Authorization:** Auth String (For details, see [Request Signature](#)).

Request Parameters

This API has no request parameters.

Request Body

Submit a request with **application/xml** data that contains the storage quota information to configure for the Space.

```
<SpaceQuota>
  <Capacity>1099511627776</Capacity>
</SpaceQuota>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type	Required
SpaceQuota	None	Contains all request information for the Put Space Quota operation	Container	Yes

The content of the Container node SpaceQuota:

Node Name (Keyword)	Parent Node	Description	Type	Required
Capacity	SpaceQuota	Quota size, in bytes (string). Must be a valid non-negative integer and not exceed 9223372036854775807 (the maximum value of int64).	number	Yes

Response

Response Headers

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

Response Body

If the request succeeds, **application/xml** data is returned, containing the space's quota information.

```
<SpaceQuota>
  <QuotaId>12345</QuotaId>
  <Capacity>102400</Capacity>
</SpaceQuota>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
SpaceQuota	None	Contains all information for the Put Space Quota result.	Container

The content of the Container node SpaceQuota:

Node Name (Keyword)	Parent Node	Description	Type
QuotaId	SpaceQuota	The unique identifier of a quota	string
Capacity	SpaceQuota	Quota capacity, in bytes	number

Examples

Request

```
PUT /test-space/?space-quota HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
Content-Type: application/xml
Content-Length: 64
Authorization: q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-
list=content-length;content-md5;content-type;date;host&q-url-param-
list=delete&q-signature=*****

<SpaceQuota>
  <Capacity>102400</Capacity>
</SpaceQuota>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 118
```

Date: Tue, 20 Aug 2019 11:59:35 GMT

x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****

<SpaceQuota>

<QuotaId>1</QuotaId>

<Capacity>102400</Capacity>

</SpaceQuota>

Get Space Quota

Last updated: 2026-07-17 10:50:13

Overview

Query the storage quota of a specified Space. The SpaceId is extracted from the URL path and must be a single-level directory name.

Request

Request Syntax

```
GET /{spaceId}/?space-quota HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Authorization: Auth String
```

Note:

- SpaceId: It corresponds to the first-level directory of an ObjectKey. To create the corresponding SpaceId, you must first upload an empty first-level directory object using the PUT Object API.
- Host: <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- Authorization: Auth String (For details, see [Request Signature](#)).

Request Parameters

This API has no request parameters.

Request Body

This API has no request body.

Response

Response Headers

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

Response Body

The request is successful, returning **application/xml** data that contains the Quota information of the Space.

```
<SpaceQuota>
  <QuotaId>12345</QuotaId>
  <Capacity>102400</Capacity>
</SpaceQuota>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
SpaceQuota	None	Contains all information for the Get Space Quota result.	Container

The content of the Container node SpaceQuota:

Node Name (Keyword)	Parent Node	Description	Type
Quotaid	SpaceQuota	The unique identifier of a quota	string
Capacity	SpaceQuota	Quota capacity, in bytes	number

Examples

Request

```
GET /test-space/?space-quota HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-
list=content-length;content-md5;content-type;date;host&q-url-param-
list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: 118
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****

<SpaceQuota>
  <QuotaId>1</QuotaId>
  <Capacity>102400</Capacity>
</SpaceQuota>
```

Put Space Extension

Last updated: 2026-07-17 10:50:27

Feature Description

Set the extended attributes of a specified Space. The SpaceId is extracted from the URL path and must be a single-level directory name.

Request

Request Syntax

```
PUT /{spaceId}/?space-extension HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Content-Type: application/xml
Content-Length: Content Length
Authorization: Auth String

[Request Body]
```

Note:

- **SpaceId:** It corresponds to the first-level directory of an ObjectKey. To create the corresponding SpaceId, you must first upload an empty first-level directory object using the PUT Object API.
- **Host:** <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- **Authorization:** Auth String (For details, see [Request Signature](#)).

Request Parameters

This API has no request parameters.

Request Body

Submit **application/xml** request data containing the extended attribute information of the Space to be configured.

```
<SpaceExtension>
```

```
<SpaceTag>personal</SpaceTag>
</SpaceExtension>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type	Required
SpaceExtension	None	All request information for the Put Space Extension operation.	Container	Yes

The content of the Container node SpaceExtension:

Node Name (Keyword)	Parent Node	Description	Type	Required
SpaceTag	SpaceExtension	A Space tag used to distinguish between Space types, such as personal Spaces and team Spaces.	string	Yes

Response

Response Headers

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

Response Body

The response body is empty.

Practical Case

Request

```
PUT /test-space/?space-extension HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
Content-Type: application/xml
Content-Length: 64
Authorization: q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-
```

```
list=content-length;content-md5;content-type;date;host&q-url-param-  
list=delete&q-signature=*****
```

```
<SpaceExtension>  
  <SpaceTag>personal</SpaceTag>  
</SpaceExtension>
```

Response

```
HTTP/1.1 200 OK  
Content-Length: 0  
Date: Tue, 20 Aug 2019 11:59:35 GMT  
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****
```

Get Space Extension

Last updated: 2026-07-24 14:52:49

Overview

Query the extended attributes of a specified Space.

Request

Request Syntax

```
GET /<SpaceId>/?space-extension HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
Authorization: Auth String
```

Note:

- SpaceId: It corresponds to the first-level directory of an ObjectKey. To create the corresponding SpaceId, you must first upload an empty first-level directory object using the PUT Object API.
- Host: <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- Authorization: Auth String (For details, see [Request Signature](#)).

Request Parameters

This API has no request parameters.

Request Body

This API has no request body.

Response

Response Headers

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

Response Body

The request is successful, returning **application/xml** data that contains the extended attributes of the Space.

```
<SpaceExtension>
  <IsPublicRead>false</IsPublicRead>
  <RecognizeSensitiveContent>false</RecognizeSensitiveContent>
  <SpaceTag>personal</SpaceTag>
</SpaceExtension>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
SpaceExtension	None	Stores all information for the Get Space Extension result.	Container

The content of the Container node SpaceExtension:

Node Name (Keyword)	Parent Node	Description	Type
IsPublicRead	SpaceExtension	Whether public read is enabled. When a Space is set to public read, some read APIs do not require an access token when the Space is requested through the SMH interface. For details, see Query Tenant Space Properties .	Boolean
RecognizeSensitiveContent	SpaceExtension	Whether to detect sensitive content	Boolean
SpaceTag	SpaceExtension	Space Tag	string

Examples

Request

```
GET /test-space/?space-extension HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-
ak=*****&q-sign-
```

```
time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-  
list=content-length;content-md5;content-type;date;host&q-url-param-  
list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK  
Content-Type: application/xml  
Content-Length: 118  
Date: Tue, 20 Aug 2019 11:59:35 GMT  
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****  
  
<SpaceExtension>  
  <IsPublicRead>false</IsPublicRead>  
  <RecognizeSensitiveContent>false</RecognizeSensitiveContent>  
  <SpaceTag>personal</SpaceTag>  
</SpaceExtension>
```

Copy Object Recursive

Last updated: 2026-07-17 10:50:58

Overview

Recursively copy the entire directory tree, including all subfiles and subdirectories, to the target location.

⚠ Attention:

Errors may be returned when a COS Agent Bucket receives a copy request or is in the process of copying an object. If an error occurs before the copy operation starts, you receive a standard error response. If an error occurs during the copy operation, the system still returns HTTP 200 OK and includes the error in the response body. This means an HTTP 200 OK response can indicate either success or failure. When using this API, you must further determine the success or failure of the copy request based on the response body content and handle the result accordingly.

Request

Request Syntax

```
PUT /<ObjectKey>?recursive HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
x-cos-copy-source: <BucketName-APPID>.cos.
<Region>.myqcloud.com/<SourceObjectKey>
Authorization: Auth String
```

ⓘ Note:

- Host: <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- ObjectKey: The path to the target object.
- SourceObjectKey: The path to the object to be copied.
- Authorization: Auth String (For details, see [Request Signature](#)).

Request Parameters

Term	Description	Type	Required
recursive	Enables recursive directory copy mode. No value assignment is required; it only needs to appear in the URL.	No value	Yes
conflict_resolution_strategy	Conflict resolution policy. For recursive directory copy, the optional values are ask and rename, with ask as the default. - In case of a conflict, the system returns HTTP 409 Conflict and the SameNameDirectoryOrFileExists error code. - rename: Automatically renames the last-level directory in case of a conflict.	string	No

Request Body

This API has no request body.

Response

Response Headers

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

Response Body

The request is successful, returning **application/xml** data.

```
<CopyObjectResult>
  <ETag>#{34;etag-value#{34;</ETag>
  <LastModified>2025-01-15T10:30:00Z</LastModified>
  <CRC64>xxx</CRC64>
</CopyObjectResult>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
------------------------	----------------	-------------	------

CopyObjectResult	None	All information for the Copy Object Recursive result.	Container
------------------	------	---	-----------

CopyObjectResult content of Container nodes:

Node Name (Keyword)	Parent Node	Description	Type
ETag	CopyObjectResult	The ETag value of the target object (enclosed in double quotes).	string
LastModified	CopyObjectResult	The last modification time of the target object, in RFC 3339 format.	date
CRC64	CopyObjectResult	The CRC64 value of the target object.	integer

Examples

Example One: Simple Case

Request

```
PUT /myspace/bbb?recursive HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
x-cos-copy-source: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com/myspace/aaa
Authorization: q-sign-algorithm=sha1&q-ak=*****&q-sign-time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: keep-alive
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****
```

```
<CopyObjectResult>
  <ETag>#34;d-1780579801973809791&#34;</ETag>
  <LastModified>2026-06-04T13:30:01Z</LastModified>
  <CRC64>5404623458528365127</CRC64>
</CopyObjectResult>
```

Example Two: Destination Already Exists During Recursive Copy

Request

```
PUT /myspace/bbb?recursive HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
x-cos-copy-source: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com/myspace/aaa
Authorization: q-sign-algorithm=sha1&q-ak=*****&q-sign-time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: keep-alive
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****

<Error>
  <Code>FileAlreadyExists</Code>
  <Message>The destination directory already exists. Recursive directory copy does not support overwriting an existing destination directory. Use conflict_resolution_strategy=rename to auto-rename, or remove the existing destination first.</Message>
  <Resource>/smh26zn7j7lfnwnc-1251081331/myspace/bbb</Resource>
</Error>
```

Move Object

Last updated: 2026-07-17 10:51:11

Overview

Move an object (file or directory) to a new location within the same media library.

Attention:

Errors may be returned when a COS Agent Bucket receives a move request or is moving an object. If an error occurs before the move operation starts, you will receive a standard error response. If an error occurs during the move operation, the interface still returns HTTP 200 OK, with the error included in the response body. This means an HTTP 200 OK response can indicate either success or failure. When using this API, you should further determine the success or failure of the move request based on the response body content and handle the result accordingly.

Request

Request Syntax

```
PUT /<ObjectKey> HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
x-cos-move-source: <BucketName-APPID>.cos.
<Region>.myqcloud.com/<SourceObjectKey>
Authorization: Auth String
```

Note:

- Host: <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- SourceObjectKey: The path to the object to be moved.
- Authorization: Auth String (For details, see [Request Signature](#)).

Request Parameters

Term	Description	Type	Required
conflict_resolution_strategy	The conflict resolution policy when the target already exists. Valid values: ask, rename, overwrite • In case of a conflict, the system returns HTTP 409 Conflict and the SameNameDirectoryOrFileExists error code. • rename: Automatically renames the last-level directory in case of a conflict. • overwrite: If the conflicting target is a directory, the system returns HTTP 409 Conflict and the SameNameDirectoryOrFileExists error code. Otherwise, it overwrites the existing file.	string	No

Request Body

This API has no request body.

Response

Response Headers

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

Response Body

The request is successful, returning **application/xml** data.

```
<MoveObjectResult>
  <ETag>#34;etag-value#34;</ETag>
  <LastModified>2025-01-15T10:30:00Z</LastModified>
  <CRC64>xxx</CRC64>
</MoveObjectResult>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
------------------------	----------------	-------------	------

MoveObjectResult	None	Contains all information for the Move Object Recursive result.	Container
------------------	------	--	-----------

MoveObjectResult content of Container nodes:

Node Name (Keyword)	Parent Node	Description	Type
ETag	MoveObjectResult	The ETag value of the target object (enclosed in double quotes).	string
LastModified	MoveObjectResult	The last modification time of the target object, in RFC 3339 format.	date
CRC64	MoveObjectResult	The CRC64 value of the target object.	integer

Examples

Request

```
PUT /myspace/bbb HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
x-cos-move-source: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com/myspace/aaa
Authorization: q-sign-algorithm=sha1&q-ak=*****&q-sign-time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: keep-alive
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDJfYzN18zMGM1****
```

```
<MoveObjectResult>  
  <ETag>#34;d-1780579801973809791#34;</ETag>  
  <LastModified>2026-06-04T13:30:01Z</LastModified>  
  <CRC64>5404623458528365127</CRC64>  
</MoveObjectResult>
```

Move Object Recursive

Last updated: 2026-07-17 10:51:24

Overview

Recursively move the entire directory tree, including all subfiles and subdirectories, to the target location.

⚠ Attention:

Errors may be returned when a COS Agent Bucket receives a move request or is moving an object. If an error occurs before the move operation starts, you will receive a standard error response. If an error occurs during the move operation, the interface still returns HTTP 200 OK, with the error included in the response body. This means an HTTP 200 OK response can indicate either success or failure. When using this API, you should further determine the success or failure of the move request based on the response body content and handle the result accordingly.

Request

Request Syntax

```
PUT /<ObjectKey>?recursive HTTP/1.1
Host: <BucketName-APPID>.cos.<Region>.myqcloud.com
x-cos-move-source: <BucketName-APPID>.cos.
<Region>.myqcloud.com/<SourceObjectKey>
Authorization: Auth String
```

ⓘ Note:

- Host: <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- Authorization: Auth String (For details, see [Request Signature](#)).

Request Parameters

Term	Description	Type	Required
recursive	Enables recursive directory move mode. No value assignment is required; it only needs to appear in the URL.	No value	Yes
conflict_resolution_strategy	Conflict resolution policy. For recursive directory move, the optional values are ask and rename, with ask as the default. - In case of a conflict, the system returns HTTP 409 Conflict and the SameNameDirectoryOrFileExists error code. - rename: Automatically renames the last-level directory in case of a conflict.	string	No

Request Body

This API has no request body.

Response

Response Headers

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

Response Body

The request is successful, returning `application/xml` data.

```
<MoveObjectResult>
  <ETag>#34;etag-value#34;</ETag>
  <LastModified>2025-01-15T10:30:00Z</LastModified>
  <CRC64>xxx</CRC64>
</MoveObjectResult>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
------------------------	-------------	-------------	------

MoveObjectResult	None	Stores all information for the Move Object Recursive result.	Container
------------------	------	--	-----------

The content of the Container node SpaceQuota:

Node Name (Keyword)	Parent Node	Description	Type
ETag	MoveObjectResult	The ETag value of the target object (enclosed in double quotes).	string
LastModified	MoveObjectResult	The last modification time of the target object, in RFC 3339 format.	date
CRC64	MoveObjectResult	The CRC64 value of the target object.	integer

Examples

Example One: Simple Case

Request

```
PUT /myspace/bbb?recursive HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
x-cos-move-source: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com/myspace/aaa
Authorization: q-sign-algorithm=sha1&q-ak=*****&q-sign-time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-list=content-length;content-md5;content-type;date;host&q-url-param-list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: keep-alive
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDJfYzN18zMGM1****
```

```
<MoveObjectResult>
  <ETag>#34;d-1780579801973809791&#34;</ETag>
  <LastModified>2026-06-04T13:30:01Z</LastModified>
  <CRC64>5404623458528365127</CRC64>
</MoveObjectResult>
```

Example Two: Destination Already Exists During Recursive Move

Request

```
PUT /myspace/bbb?recursive HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
x-cos-move-source: smhxxx-1250000000.cos.ap-
chengdu.myqcloud.com/myspace/ccc
Authorization: q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-
list=content-length;content-md5;content-type;date;host&q-url-param-
list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: keep-alive
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****

<Error>
  <Code>FileAlreadyExists</Code>
  <Message>Recursive move operation failed: File already exists.
</Message>
  <Resource>/smhxxx-1250000000/myspace/bbb</Resource>
</Error>
```

Delete Object Recursive

Last updated: 2026-07-17 10:51:38

Overview

Recursively delete the entire directory tree, including all subfiles and subdirectories.

Attention:

- Errors may be returned when a COS Agent Bucket receives a delete request or is deleting an object. If an error occurs before the delete operation starts, you will receive a standard error response. If an error occurs during the delete operation, the interface still returns HTTP 200 OK, with the error included in the response body. This means an HTTP 200 OK response can indicate either success or failure. When using this API, you should further determine the success or failure of the delete request based on the response body content and handle the result accordingly.
- Deleted data cannot be recovered. Proceed with caution.

Request

Request Syntax

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

Note:

- Host: <BucketName-APPID>.cos.<Region>.myqcloud.com. Here, <BucketName-APPID> is the bucket name with the APPID suffix, for example, smhxxxxxxxxxxxxx-1250000000. <Region> is the available region. For details, see [Product Overview](#).
- ObjectKey: The object path.
- Authorization: Auth String (For details, see [Request Signature](#)).

Request Parameters

Term	Description	Type	Required
recursive	Enables recursive deletion of the entire directory tree. No value assignment is required; it only needs to appear in the URL.	No Value	Yes

Request Body

This API has no request body.

Response

Response Headers

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

Response Body

The request is successful, returning **application/xml** data.

```
<DeleteResult>
  <Deleted>
    <Key>key</Key>
  </Deleted>
</DeleteResult>
```

The detailed node descriptions are as follows:

Node Name (Keyword)	Parent Node	Description	Type
DeleteResult	None	Save all information for the Delete Object Recursive result.	Container

DeleteResult content of container nodes:

Node Name (Keyword)	Parent Node	Description	Type
Deleted	DeleteResult	Entries of successfully deleted objects	Container

Key	Deleted	Path of the successfully deleted object.	string
-----	---------	--	--------

Examples

Request

```
DELETE /myspace/bbb?recursive HTTP/1.1
Host: smhxxx-1250000000.cos.ap-chengdu.myqcloud.com
Authorization: q-sign-algorithm=sha1&q-
ak=*****&q-sign-
time=1566302375;1566309575&q-key-time=1566302375;1566309575&q-header-
list=content-length;content-md5;content-type;date;host&q-url-param-
list=delete&q-signature=*****
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Transfer-Encoding: chunked
Connection: keep-alive
Date: Tue, 20 Aug 2019 11:59:35 GMT
x-cos-request-id: NWQ1YmUwYTdfM2FiMDJhMDlfYzczN18zMGM1****
```

```
<DeleteResult>
  <Deleted>
    <Key>myspace/bbb</Key>
  </Deleted>
</DeleteResult>
```