

# Software Repository for Container

## API Reference

Issue 01  
Date 2020-11-04



**Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

---

# Contents

---

<b>1 Before You Start.....</b>	<b>1</b>
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Constraints.....	1
1.5 Concepts.....	2
<b>2 API Overview.....</b>	<b>4</b>
<b>3 Calling APIs.....</b>	<b>8</b>
3.1 Making an API Request.....	8
3.2 Authentication.....	12
3.3 Response.....	14
<b>4 API.....</b>	<b>16</b>
4.1 Creating an Organization.....	16
4.2 Deleting an Organization.....	17
4.3 Querying the List of Organizations.....	18
4.4 Querying the Details of an Organization.....	20
4.5 Creating an Image Repository.....	21
4.6 Deleting an Image Repository.....	23
4.7 Updating the Brief Information on an Image Repository.....	24
4.8 Querying the Brief Information on an Image Repository.....	25
4.9 Querying the List of Images.....	28
4.10 Querying the List of Image Tags.....	32
4.11 Deleting the Image with a Specified Tag.....	34
4.12 Generating a Temporary Login Command.....	35
4.13 Creating an Account Used for Image Sharing.....	37
4.14 Deleting an Account Used for Image Sharing.....	39
4.15 Updating an Account Used for Image Sharing.....	40
4.16 Checking the Existence of an Account Used for Image Sharing.....	41
4.17 Obtaining the List of Accounts Used for Image Sharing.....	43
4.18 Viewing the List of Shared Images.....	45
4.19 Querying All API Versions.....	48
4.20 Querying a Specific API Version.....	50

4.21 Creating Organization Permissions.....	52
4.22 Deleting Organization Permissions.....	53
4.23 Updating Organization Permissions.....	54
4.24 Querying Organization Permissions.....	56
4.25 Creating Image Permissions.....	58
4.26 Deleting Image Permissions.....	59
4.27 Updating Image Permissions.....	61
4.28 Querying Image Permissions.....	62
4.29 Obtaining Information About an Image Synchronization Task.....	64
4.30 Obtaining the List of Automatic Image Synchronization Tasks.....	66
4.31 Creating an Automatic Image Synchronization Task.....	68
4.32 Deleting an Automatic Image Synchronization Task.....	70
4.33 Manually Synchronizing an Image.....	71
4.34 Creating a Trigger.....	73
4.35 Obtaining the Trigger List of a Specific Image Repository.....	75
4.36 Deleting a Trigger.....	77
4.37 Obtaining Information About a Trigger.....	78
4.38 Updating Trigger Settings.....	81
4.39 Adding an Image Retention Policy.....	83
4.40 Obtaining the List of Image Retention Policies.....	85
4.41 Deleting an Image Retention Policy.....	87
4.42 Obtaining an Image Retention Policy Record.....	88
4.43 Modifying an Image Retention Policy.....	90
4.44 Obtaining Image Retention Records.....	92
<b>5 Appendixes.....</b>	<b>95</b>
5.1 Status Code.....	95
5.2 Error Code.....	98
5.3 Obtaining a Project ID.....	109
5.4 Obtaining an Account ID.....	110
<b>6 Change History.....</b>	<b>112</b>

# 1 Before You Start

---

## 1.1 Overview

Welcome to use SoftWare Repository for Container (SWR). SWR provides easy, secure, and reliable management over container images throughout their lifecycles, facilitating the deployment of containerized applications.

This document describes how to use APIs to perform operations on SWR, such as creation, deletion, and query. For details about all supported operations, see [API Overview](#).

If you plan to access SWR through APIs, ensure that you are familiar with SWR concepts. For details, see [Overview](#).

## 1.2 API Calling

SWR supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [Calling APIs](#).

## 1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of SWR, see [Regions and Endpoints](#).

## 1.4 Constraints

- The number of organizations and image tags that you can create is determined by your quotas. For details, see [Service Quota](#). For details about how to increase quotas, see [Quotas](#).
- For more constraints, see API description.

## 1.5 Concepts

- **Account**

An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. To ensure account security, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- **User**

An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).

You can view the account ID and IAM user ID on the [My Credentials](#) page of the management console. API authentication requires information such as the account name, username, and password.
- **Region**

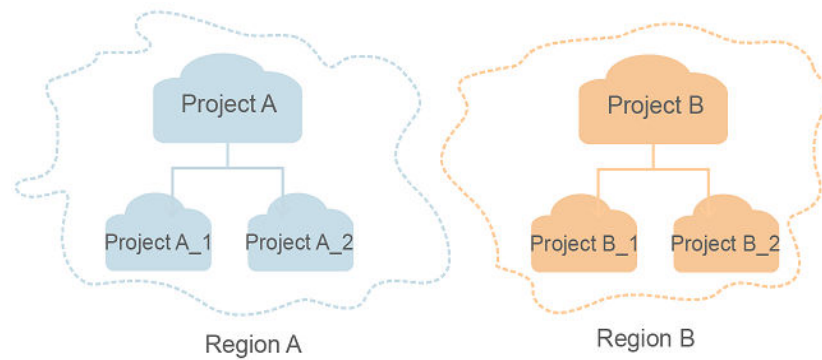
Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

For details, see [Region and AZ](#).
- **AZ**

An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.
- **Project**

A project corresponds to a region. Default projects are defined to a group and have physically isolated resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources in the region under their accounts. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

**Figure 1-1** Project isolation model



To view a project ID, go to the [My Credentials](#) page.

- Enterprise project

Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

For details about enterprise projects and about how to obtain enterprise project IDs, see [Enterprise Management User Guide](#).

# 2 API Overview

You can query information about organizations, image repositories, and images, and manage organizations, image repositories, and images by using interfaces provided by the container image service.

**Table 2-1** API description

API	Description
<a href="#">Creating an Organization</a>	Create an organization.
<a href="#">Deleting an Organization</a>	Delete an organization.
<a href="#">Querying the List of Organizations</a>	Query the list of organizations.
<a href="#">Querying the Details of an Organization</a>	Query the details of an organization, including its ID, name, creator, and user permissions.
<a href="#">Creating an Image Repository</a>	Create an image repository in an organization.
<a href="#">Deleting an Image Repository</a>	Delete an image repository in an organization.
<a href="#">Updating the Brief Information on an Image Repository</a>	Update the brief information on an image in an organization of the account, including the <b>category</b> , <b>is_public</b> , and <b>description</b> .
<a href="#">Querying the Brief Information on an Image Repository</a>	Query brief information on an image in an organization of the account, including the image's repository ID, organization ID, repository name, description of the repository, the creator and capacity of the repository, and the image pull address.
<a href="#">Querying the List of Images</a>	Query the list of images.

API	Description
<a href="#">Querying the List of Image Tags</a>	Query the list of image tags.
<a href="#">Deleting the Image with a Specified Tag</a>	Delete the image with a specified tag in an image repository.
<a href="#">Generating a Temporary Login Command</a>	Generate a temporary login command using the value of <b>X-Swr-Dockerlogin</b> in the response header and the value of <b>host</b> in the response body.
<a href="#">Creating an Account Used for Image Sharing</a>	Create an account used for image sharing. You can share your private images with other users, granting them permissions to pull the images.
<a href="#">Deleting an Account Used for Image Sharing</a>	Delete an account used for image sharing.
<a href="#">Updating an Account Used for Image Sharing</a>	Update an account used for image sharing.
<a href="#">Checking the Existence of an Account Used for Image Sharing</a>	Check whether an account used for image sharing exists.
<a href="#">Obtaining the List of Accounts Used for Image Sharing</a>	Obtain the list of accounts used for image sharing.
<a href="#">Viewing the List of Shared Images</a>	View the list of shared images.
<a href="#">Querying All API Versions</a>	Query all available API versions of SWR.
<a href="#">Querying a Specific API Version</a>	Query a specific API version of SWR.
<a href="#">Creating Organization Permissions</a>	Create organization permissions.
<a href="#">Deleting Organization Permissions</a>	Delete organization permissions.
<a href="#">Updating Organization Permissions</a>	Update organization permissions.
<a href="#">Querying Organization Permissions</a>	Query organization permissions.
<a href="#">Creating Image Permissions</a>	Create image permissions.

API	Description
<a href="#">Deleting Image Permissions</a>	Delete image permissions.
<a href="#">Updating Image Permissions</a>	Update image permissions.
<a href="#">Querying Image Permissions</a>	Query image permissions.
<a href="#">Obtaining Information About an Image Synchronization Task</a>	Obtain information about an image synchronization task.
<a href="#">Obtaining the List of Automatic Image Synchronization Tasks</a>	Obtain the list of automatic image synchronization tasks.
<a href="#">Creating an Automatic Image Synchronization Task</a>	Create an automatic image synchronization task.
<a href="#">Deleting an Automatic Image Synchronization Task</a>	Delete an automatic image synchronization task.
<a href="#">Manually Synchronizing an Image</a>	Manually synchronize an image.
<a href="#">Creating a Trigger</a>	Create a trigger.
<a href="#">Obtaining the Trigger List of a Specific Image Repository</a>	Obtain the trigger list of a specific image repository.
<a href="#">Deleting a Trigger</a>	Delete a trigger.
<a href="#">Obtaining Information About a Trigger</a>	Obtain information about a trigger.
<a href="#">Updating Trigger Settings</a>	Update trigger settings.
<a href="#">Adding an Image Retention Policy</a>	Add an image retention policy.
<a href="#">Obtaining the List of Image Retention Policies</a>	Obtain the list of image retention policies.
<a href="#">Deleting an Image Retention Policy</a>	Delete an image retention policy.

API	Description
<b>Obtaining an Image Retention Policy Record</b>	Obtain the records of image retention policies.
<b>Modifying an Image Retention Policy</b>	Modify an image retention policy.
<b>Obtaining Image Retention Records</b>	Obtain image retention records.

# 3 Calling APIs

## 3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **obtaining a user token** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

### Request URI

A request URI is in the following format:

**{URI-scheme}://{Endpoint}/{resource-path}?{query-string}**

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

**Table 3-1** URI parameter description

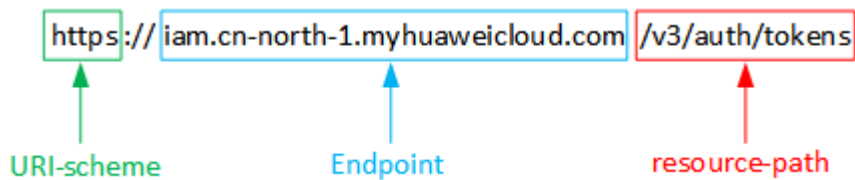
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from <b>Regions and Endpoints</b> . For example, the endpoint of IAM in the <b>CN North-Beijing1</b> region is <b>iam.cn-north-1.myhuaweicloud.com</b> .
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the <b>resource-path</b> of the API used to obtain a user token is <b>/v3/auth/tokens</b> .

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of " <i>Parameter name=Parameter value</i> ". For example, <b>?limit=10</b> indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **CN North-Beijing1** region, obtain the endpoint of IAM (**iam.cn-north-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

**Figure 3-1** Example URI



**NOTE**

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

## Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server.

**Table 3-2** HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.

Method	Description
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to [obtain a user token](#), the request method is **POST**. The request is as follows:

POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens

## Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows.

**Table 3-3** Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of <i>Hostname:Port number</i> . If the port number is not specified, the default port is used. The default port number for <b>https</b> is <b>443</b> .	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value <b>application/json</b> is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in <a href="#">Obtaining a Project ID</a> .	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies a user token. It is a response to the API for <a href="#">obtaining a user token</a> (only this API does not require authentication). After the request is processed, the value of <b>X-Subject-Token</b> in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZlhvcNAQcCo...ggg1BBIINPXsidG9rZ

 **NOTE**

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in [Authentication](#).

The API used to [obtain a user token](#) does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

### (Optional) Request Body

This part is optional. The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to [obtain a user token](#), the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*,

*domainname*, *\*\*\*\*\** (login password), and *xxxxxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from [Regions and Endpoints](#).

 **NOTE**

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see [Obtaining a User Token](#).

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

## 3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure than token-based authentication.

### Token-based Authentication

 **NOTE**

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

You can obtain a token by calling the [Obtaining User Token](#) API. When you call the API, set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

## AK/SK-based Authentication

### NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see [API Request Signing Guide](#).

### NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

## 3.3 Response

### Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits ranging from 1xx to 5xx. It indicates the status of a response. For more information, see [Status Code](#).

If status code 201 is returned for the calling of the API for [obtaining a user token](#), the request is successful.

### Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

[Figure 3-2](#) shows the response header for the API of [obtaining a user token](#), in which **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

**Figure 3-2** Header fields of the response to the request for obtaining a user token

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token
→ MIIVXQVJKoZIhvcNAQcCoIIYJCCEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6ijlwMTktMDItMTNUMC
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgkklQ01wi4JlGzrpd18LGXK5txldfq4lqHCYb8P4NaY0NYejcAgzJVeFYtLWT1GSO0zxKZmlQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jqglFkNPQuFSOUB+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUx3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbvpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

### (Optional) Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to [obtain a user token](#).

```
{
  "token": {
```

```
"expires_at": "2019-02-13T06:52:13.855000Z",  
"methods": [  
  "password"  
],  
"catalog": [  
  {  
    "endpoints": [  
      {  
        "region_id": "xxxxxxx",  
.....
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{  
  "error_msg": "The format of message is error",  
  "error_code": "AS.0001"  
}
```

In the response body, **error\_code** is an error code, and **error\_msg** provides information about the error.

# 4 API

## 4.1 Creating an Organization

### Function

Create an organization.

### URI

POST /v2/manage/namespaces

### Request

#### Request parameters

**Table 4-1** Request body parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name. Enter 1 to 64 characters, starting with a lowercase letter and ending with a lowercase letter or digit. Only lowercase letters, digits, periods (.), underscores (_), and hyphens (-) are allowed. Periods, underscores, and hyphens cannot be placed next to each other. A maximum of two consecutive underscores are allowed.

#### Request example

Create an organization named **group**.

```
{  
  "namespace": "group"  
}
```

## Response

```
{}
```

## Status Code

For details about status code, see [Table 4-2](#).

**Table 4-2** Status code

Status Code	Description
201	Created successfully.
400	The request is incorrect or the number of organizations exceeds the upper limit.
401	Authentication failed.
409	The resource already exists.
500	Internal error.

## 4.2 Deleting an Organization

### Function

Delete an organization.

### URI

```
DELETE /v2/manage/namespaces/{namespace}
```

For details about parameters, see [Table 4-3](#).

**Table 4-3** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

### Request

N/A

### Response

N/A

## Status Code

For details about status code, see [Table 4-4](#).

**Table 4-4** Status code

Status Code	Description
204	Deleted successfully.
400	The server failed to process the request.
401	Authentication failed.
404	The organization does not exist.
500	Internal error.

## 4.3 Querying the List of Organizations

### Function

Query the list of organizations.

### URI

GET /v2/manage/namespaces?filter=namespace::{*namespace*}

For details about parameters, see [Table 4-5](#).

**Table 4-5** Parameter description

Parameter	Mandatory	Type	Description
namespace	No	String	Organization name

### Request

N/A

### Response

**Response parameters**

**Table 4-6** Response body parameter description

Parameter	Type	Description
namespaces	Array of objects	Organization list. For details, see <a href="#">Table 4-7</a> .

**Table 4-7** namespaces parameter description

Parameter	Type	Description
ID	Integer	ID
name	String	Organization name
creator_name	String	IAM username
auth	Integer	User permission. <b>7</b> : Manage. <b>3</b> : Write. <b>1</b> : Read.

### Response example

```
{
  "namespaces": [
    {
      "id": 1422,
      "name": "group",
      "creator_name": "username",
      "auth": 7
    }
  ]
}
```

## Status Code

For details about status code, see [Table 4-8](#).

**Table 4-8** Status code

Status Code	Description
200	Query succeeded.
400	The server failed to process the request.
401	Authentication failed.
404	The organization does not exist.
500	Internal error.

## 4.4 Querying the Details of an Organization

### Function

Query the details of an organization.

### URI

GET `/v2/manage/namespaces/{namespace}`

For details about parameters, see [Table 4-9](#).

**Table 4-9** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

### Request

N/A

### Response

#### Response parameters

**Table 4-10** Response body parameter description

Parameter	Type	Description
id	Integer	ID
name	String	Organization name
creator_name	String	IAM username
auth	Integer	User permission. <b>7</b> : Manage. <b>3</b> : Write. <b>1</b> : Read.

#### Response example

```
{
  "id": 1422,
  "name": "group",
  "creator_name": "username",
  "auth": 7
}
```

### Status Code

For details about status code, see [Table 4-11](#).

**Table 4-11** Status code

Status Code	Description
200	Query succeeded.
400	The server failed to process the request.
401	Authentication failed.
404	The organization does not exist.
500	Internal error.

## 4.5 Creating an Image Repository

### Function

Create an image repository in an organization.

### URI

POST /v2/manage/namespaces/{*namespace*}/repos

For details about parameters, see [Table 4-12](#).

**Table 4-12** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name. Enter 1 to 64 characters, starting with a lowercase letter and ending with a lowercase letter or digit. Only lowercase letters, digits, periods (.), underscores (_), and hyphens (-) are allowed. Periods, underscores, and hyphens cannot be placed next to each other. A maximum of two consecutive underscores are allowed.

Parameter	Mandatory	Type	Description
repository	Yes	String	Name of the image repository. Enter 1 to 128 characters, starting and ending with a lowercase letter or digit. Only lowercase letters, digits, periods (.), slashes (/), underscores (_), and hyphens (-) are allowed. Periods, slashes, underscores, and hyphens cannot be placed next to each other. A maximum of two consecutive underscores are allowed.
category	No	String	Repository type. The value can be <b>app_server</b> , <b>linux</b> , <b>framework_app</b> , <b>database</b> , <b>lang</b> , <b>other</b> , <b>windows</b> , or <b>arm</b> .
description	No	String	Brief description of the image repository.
is_public	Yes	Boolean	Whether the repository is a public repository. When the value is <b>true</b> , it indicates the repository is public. When the value is <b>false</b> , it indicates the repository is private.

## Request

### Request example

```
-F "repository=busybox" \  
-F "category=linux" \  
-F "description=this is a busybox repository" \  
-F "is_public=true"
```

## Response

```
{}
```

## Status Code

For details about status code, see [Table 4-13](#).

**Table 4-13** Status code

Status Code	Description
201	Created successfully.
400	Request error. Error information is returned.

Status Code	Description
401	Authentication fails.
409	The repository already exists.
500	Internal error. Error information is returned.

## 4.6 Deleting an Image Repository

### Function

Delete an image repository in an organization.

### URI

DELETE /v2/manage/namespaces/{*namespace*}/repos/{*repository*}

For details about parameters, see [Table 4-14](#).

**Table 4-14** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.
repository	Yes	String	Name of the image repository.

### Request

N/A

### Response

N/A

### Status Code

For details about status code, see [Table 4-15](#).

**Table 4-15** Status code

Status Code	Description
204	The brief information about the image repository is successfully deleted.

Status Code	Description
400	Request error. Error information is returned.
401	Authentication fails.
404	The repository does not exist.
406	The repository cannot be deleted because it contains images.
500	Internal error. Error information is returned.

## 4.7 Updating the Brief Information on an Image Repository

### Function

Update the brief information on an image repository in an organization, including the **category**, **is\_public**, and **description**.

### URI

PATCH /v2/manage/namespaces/{*namespace*}/repos/{*repository*}

For details about parameters, see [Table 4-16](#).

**Table 4-16** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.
repository	Yes	String	Name of the image repository.
category	No	String	Repository type. The value can be <b>app_server</b> , <b>linux</b> , <b>framework_app</b> , <b>database</b> , <b>lang</b> , <b>other</b> , <b>windows</b> , or <b>arm</b> .
description	No	String	Repository description.
is_public	No	Boolean	Whether the repository is a public repository. The value can be either <b>true</b> or <b>false</b> .

### Request

#### Request example

```
-F "repository=busybox" \  
-F "category=linux" \  
-F "description=this is a busybox repository" \  
-F "is_public=true"
```

## Response

```
{}
```

## Status Code

For details about status code, see [Table 4-17](#).

**Table 4-17** Status code

Status Code	Description
201	The brief information about the image repository is successfully updated.
400	Request error. Error information is returned.
401	Authentication fails.
404	The repository does not exist.
500	Internal error. Error information is returned.

## 4.8 Querying the Brief Information on an Image Repository

### Function

Query the brief information on an image repository in an organization.

### URI

GET `/v2/manage/namespaces/{namespace}/repos/{repository}`

For details about parameters, see [Table 4-18](#).

**Table 4-18** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Name of the image repository

## Request

N/A

## Response

### Response parameters

**Table 4-19** Response body parameter description

Parameter	Type	Description
id	Integer	Repository ID.
ns_id	Integer	Organization ID.
name	String	Repository name.
category	String	Repository type. The value can be <b>app_server</b> , <b>linux</b> , <b>framework_app</b> , <b>database</b> , <b>lang</b> , <b>other</b> , <b>windows</b> , or <b>arm</b> .
description	String	Repository description.
creator_id	String	Repository creator ID.
creator_name	String	Repository creator.
size	Integer	Repository size.
is_public	Boolean	Whether the repository is a public repository. The value can be <b>true</b> or <b>false</b> .
num_images	Integer	Number of images in a repository.
num_download	Integer	Download times.
url	String	URL of the repository logo image. This field has been discarded and is left empty by default.
path	String	External image download address. For example, <b>swr.cn-north-1.myhuaweicloud.com/namespace/repository</b> .

Parameter	Type	Description
internal_path	String	Internal image pull address. For example, <b>10.125.0.198:20202/namespace/repository</b> .
created	String	Time when a repository is created. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
updated	String	Time when a repository is updated. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
domain_id	String	Account ID.
priority	Integer	Image sorting priority.

### Response example

```
{
  "id": 0,
  "ns_id": 0,
  "name": "busybox",
  "category": "other",
  "description": "containerops",
  "creator_id": "1",
  "creator_name": "admin",
  "size": 2099575,
  "is_public": false,
  "num_images": 0,
  "num_download": 0,
  "url": "",
  "path": "swr.cn-north-1.myhuaweicloud.com/namespace/repository",
  "internal_path": "10.125.0.198:20202/namespace/repository",
  "created": "2017-08-17T12:00:03.915153267Z",
  "updated": "2017-08-17T12:00:03.915153267Z",
  "domain_id": "da44776c316c4a99b3683bb174f8821b",
  "priority": 0
}
```

### Status Code

For details about status code, see [Table 4-20](#).

**Table 4-20** Status code

Status Code	Description
200	The request is successfully sent.
400	Request error. Error information is returned.
401	Authentication failed.
404	The repository does not exist.
500	Internal error. Error information is returned.

## 4.9 Querying the List of Images

### Function

Query the list of images.

### URI

GET /v2/manage/repos?filter=center::self|namespace::{namespace}|name::{name}|category::{category}|offset::{offset}|limit::{limit}|order\_column::{order\_column}|order\_type::{order\_type}

For details about parameters, see [Table 4-21](#).

#### NOTICE

At least one filter parameter must be transferred.

**Table 4-21** Parameter description

Parameter	Mandatory	Type	Description
center	No	String	<b>self</b> (self image) or <b>thirdparty</b> (third-party image). By default, the value is <b>self</b> .
namespace	No	String	Organization name.
name	No	String	Image repository name.
category	No	String	Image repository type. The value can be <b>app_server</b> , <b>linux</b> , <b>framework_app</b> , <b>database</b> , <b>lang</b> , <b>other</b> , <b>windows</b> , or <b>arm</b> .

Parameter	Mandatory	Type	Description
offset	No	String	Start index Ensure that the <b>offset</b> and <b>limit</b> parameters are used together.
limit	No	String	Number of returned records Ensure that the <b>offset</b> and <b>limit</b> parameters are used together.
order_column	No	String	Sorting criteria. The value can be <b>name</b> , <b>updated_time</b> , or <b>tag_count</b> . Ensure that the <b>order_column</b> and <b>order_type</b> parameters are used together.
order_type	No	String	Sorting type. You can set this parameter to <b>desc</b> (descending sort) and <b>asc</b> (ascending sort). Ensure that the <b>order_column</b> and <b>order_type</b> parameters are used together.

## Request

N/A

## Response

### Response parameters

**Table 4-22** Response header parameter description

Parameter	Type	Description
Content-Range	String	Offset (Start index)-Count (Number of records on the current page)/Total (Total number of records)

### NOTE

If the **offset** and **limit** parameters are transferred in the request, **Content-Range** will be added to the response header.

**Table 4-23** Response body parameter description

Parameter	Type	Description
name	String	Repository name.

Parameter	Type	Description
category	String	Repository type. The value can be <b>app_server</b> , <b>linux</b> , <b>framework_app</b> , <b>database</b> , <b>lang</b> , <b>other</b> , <b>windows</b> , or <b>arm</b> .
description	String	Repository description.
size	Integer	Image repository size.
is_public	Boolean	Whether the image is a public image. The value can be <b>true</b> or <b>false</b> .
num_images	Integer	Number of image tags in a repository.
num_download	Integer	Download times.
created_at	String	Time when a repository is created. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
updated_at	String	Time when a repository is updated. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
logo	String	(Reserved field) Repository logo.
url	String	(Reserved field) URL of the repository logo.
path	String	External container image download address. For example, <b>swr.cn-north-1.myhuaweicloud.com/namespace/repository</b> .

Parameter	Type	Description
internal_path	String	Internal container image download address. For example, <b>10.125.0.198:20202/namespace/repository</b> .
domain_name	String	Username.
namespace	String	Organization name.
tags	Array of strings	Image tag list.
status	Boolean	(Reserved field) Status.

### Response example

```
[
  {
    "name": "nginx",
    "category": "linux",
    "size": 200977474,
    "is_public": false,
    "num_images": 2,
    "num_download": 0,
    "created_at": "2017-01-01T00:00:00Z",
    "updated_at": "2017-09-11T03:00:02.542841141Z",
    "logo": "",
    "url": "",
    "path": "swr.cn-north-1.myhuaweicloud.com/namespace/repository",
    "internal_path": "10.125.0.198:20202/namespace/repository",
    "domain_name": "root",
    "namespace": "root",
    "tags": [
      "latest",
      "log"
    ],
    "status": false
  }
]
```

## Status Code

For details about status code, see [Table 4-24](#).

**Table 4-24** Status code

Status Code	Description
200	The request is successfully sent.
400	Request error. Error information is returned.
401	Authentication failed.
500	Internal error. Error information is returned.

## 4.10 Querying the List of Image Tags

### Function

Query all image tags of an image.

### URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/tags?filter=offset::{offset}|limit::{limit}|order\_column::{order\_column}|order\_type::{order\_type}|tag::{tag}

For details about parameters, see [Table 4-25](#).

**Table 4-25** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
offset	No	Integer	Start index
limit	No	Integer	Number of responses
order_column	No	String	Sorting by column. You can set this parameter to <b>updated_at</b> (sorting by update time).
order_type	No	String	Sorting type. You can set this parameter to <b>desc</b> (descending sort) and <b>asc</b> (ascending sort).
tag	No	String	Image tag name

### Request

N/A

### Response

#### Response parameters

**Table 4-26** Response body parameter description

Parameter	Type	Description
id	Integer	ID

Parameter	Type	Description
repo_id	Integer	Repository ID
Tag	String	Image tag
image_id	String	Image ID
manifest	String	Image manifest
digest	String	SHA value of an image
schema	Integer	Docker protocol version. The value can be <b>1</b> or <b>2</b> .
path	String	External image download address. For example, <b>swr.cn-north-1.myhuaweicloud.com/namespace/repository:tag</b> .
internal_path	String	Internal image download address. For example, <b>10.125.0.198:20202/namespace/repository:tag</b> .
size	Integer	Image size. The value ranges from 0 to 9223372036854775807.
is_trusted	Boolean	By default, the value is <b>false</b> .
created	String	Time when an image is created. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
updated	String	Time when an image is updated. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
deleted	String	Time when an image was deleted.

### Response example

```
[
  {
    "id": 0,
    "repo_id": 0,
    "tag": "latest",
    "image_id": "741f24a795d6d93d7c6edd11780d63c13e16c39615dd9d223378a57a836f2ee6",
    "manifest": "{\"schemaVersion\":2,\"mediaType\":\"application/vnd.docker.distribution.manifest.v2+json\", \"config\":{\"mediaType\":\"application/vnd.docker.container.image.v1+json\", \"size\":1862, \"digest\":\"sha256:741f24a795d6d93d7c6edd11780d63c13e16c39615dd9d223378a57a836f2ee6\"}, \"layers\":[{\"mediaType\":\"application/vnd.docker.image.rootfs.diff.tar.gzip\", \"size\":1292800, \"digest\":\"sha256:8ac8bfaff55af948c796026ee867448c5b5b5d9dd3549f4006d9759b25d4a893\"}, {\"mediaType\":\"application/vnd.docker.image.rootfs.diff.tar.gzip\", \"size\":10240, \"digest\":\"sha256:77ddb3a9fe11e81761a0f9df43a28e3e6f29bbb53c0c8cf71cd7efa69729aed\"}]\"\",
    \"digest\": \"sha256:57b605845a6367c34bfb6ea6477f16852f59aa1861a2b51d10ab77ae0a1dc9c3\",
    \"schema\": 2,
    \"path\": \"swr.cn-north-1.myhuaweicloud.com/namespace/busybox:latest\",
    \"internal_path\": \"10.125.0.198:20202/namespace/busybox:latest\",
    \"size\": 1304902,
    \"is_trusted\": false,
    \"created\": \"2018-07-06T06:18:55Z\",
    \"updated\": \"2018-07-06T06:18:55Z\",
    \"deleted\": null
  }
]
```

### Status Code

For details about status code, see [Table 4-27](#).

**Table 4-27** Status code

Status Code	Description
200	All tags in the image repository are successfully queried.
400	Request error. Error information is returned.
401	Authentication fails.
404	The repository does not exist.
500	Internal error. Error information is returned.

## 4.11 Deleting the Image with a Specified Tag

### Function

Delete the image with a specified tag in an image repository.

### URI

DELETE /v2/manage/namespaces/{*namespace*}/repos/{*repository*}/tags/{*tag*}

For details about parameters, see [Table 4-28](#).

**Table 4-28** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.
repository	Yes	String	Image repository name.
tag	Yes	String	Tag name.

## Request

N/A

## Response

N/A

## Status Code

For details about status code, see [Table 4-29](#).

**Table 4-29** Status code

Status Code	Description
204	The image with a specified tag is deleted successfully.
400	Request error. Error information is returned.
401	Authentication fails.
404	The repository or the image with a specified tag does not exist.
500	Internal error. Error information is returned.

# 4.12 Generating a Temporary Login Command

## Function

Generate a temporary login command using the value of **X-Swr-Dockerlogin** in the response header and the value of **host** in the response body.

## URI

POST /v2/manage/utils/secret?projectname={projectname}

For details about parameters, see [Table 4-30](#).

**Table 4-30** Parameter description

Parameter	Mandatory	Type	Description
projectname	No	String	Project name. The default value is the region name, for example, <b>cn-north-1</b> .

## Request

N/A

## Response

### Response parameters

**Table 4-31** Response header parameter description

Parameter	Type	Description
X-Swr-Dockerlogin	String	A login command, for example, <b>docker login -u cn-north-1@OW*****FPUKBXI -p 6f0779072*****863d1ae3ccef921b7f33</b>

**Table 4-32** Response body parameter description

Parameter	Type	Description
auths	Map<String, <a href="#">authInfo</a> >	Authentication information

**Table 4-33** authInfo parameter description

Parameter	Type	Description
auth	String	Base64-encoded authentication information

### Response example

```
{
  "auths": {
    "10.95.181.176:5300": {
```

```

    "auth": "Y24tbm9ydGg*****hhMTgzMGVmN2RhZjJm"
  }
}
}

```

A temporary login command can be generated using the value of **X-Swr-Dockerlogin** in the response header and the value of **host** in the response body.

For example, in the preceding response example, the value of **X-Swr-Dockerlogin** in the response header is **docker login -u cn-north-1@OW\*\*\*\*\*FPUKBXI -p 6f0779072\*\*\*\*\*863d1ae3ccef921b7f33**. The value of **host** in the response body is **10.95.181.176:5300**. Therefore, the generated temporary login command is as follows:

```

docker login -u cn-north-1@OW*****FPUKBXI -p
6f0779072*****863d1ae3ccef921b7f33 10.95.181.176:5300

```

## Status Code

For details about status code, see [Table 4-34](#).

**Table 4-34** Status code

Status Code	Description
200	The request is successfully sent.
400	Request error. Error information is returned.
401	Authentication failed.
500	Internal error. Error information is returned.

## 4.13 Creating an Account Used for Image Sharing

### Function

Create an account used for image sharing. You can share your private images with other users, granting them permissions to pull the images.

### URI

POST /v2/manage/namespaces/{*namespace*}/repositories/{*repository*}/access-domains

For details about parameters, see [Table 4-35](#).

**Table 4-35** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

## Request

### Request parameters

**Table 4-36** Request body parameter description

Parameter	Mandatory	Type	Description
access_domain	Yes	String	Image sharing account name.
permit	Yes	String	Currently, only the read permission is supported.
deadline	Yes	String	End time (UTC). If the sharing is permanently valid, the value is <b>forever</b> . The end time is calculated by day, and will expire till 00:00:00 of the next day.
description	No	String	Description.

### Request example

```
{
  "access_domain": "group",
  "permit": "read",
  "deadline": "2018-10-01T16:00:00.000Z",
  "description": "description"
}
```

## Response

```
{}
```

## Status Code

For details about status code, see [Table 4-37](#).

**Table 4-37** Status code

Status Code	Description
201	Creation successful.
400	The server failed to process the request.
401	Authentication failed.
409	The repository has been shared.
500	Internal error.

## 4.14 Deleting an Account Used for Image Sharing

### Function

Delete an account used for image sharing.

### URI

DELETE /v2/manage/namespaces/{*namespace*}/repositories/{*repository*}/access-domains/{*access\_domain*}

For details about parameters, see [Table 4-38](#).

**Table 4-38** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.
repository	Yes	String	Image repository name.
access_domain	Yes	String	Image sharing account name

### Request

N/A

### Response

N/A

### Status Code

For details about status code, see [Table 4-39](#).

**Table 4-39** Status code

Status Code	Description
204	Deleted successfully.
400	The server failed to process the request.
401	Authentication failed.
500	Failed to complete the request because of an internal service error.

## 4.15 Updating an Account Used for Image Sharing

### Function

Update an account used for image sharing.

### URI

PATCH `/v2/manage/namespaces/{namespace}/repositories/{repository}/access-domains/{access_domain}`

For details about parameters, see [Table 4-40](#).

**Table 4-40** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
access_domain	Yes	String	Image sharing account name

### Request

#### Request parameters

**Table 4-41** Request body parameter description

Parameter	Mandatory	Type	Description
permit	Yes	String	Currently, only the read permission is supported.

Parameter	Mandatory	Type	Description
deadline	Yes	String	Valid until (UTC). If the sharing is permanent, the value is <b>forever</b> . Otherwise, the sharing is valid until 00:00:00 of the next day.
description	No	String	Description. This parameter is left blank by default.

#### Request example

```
{
  "permit": "read",
  "deadline": "2018-10-01T16:00:00.000Z",
  "description": "description"
}
```

#### Response

N/A

#### Status Code

For details about status code, see [Table 4-42](#).

**Table 4-42** Status code

Status Code	Description
201	Modified successfully.
400	The server failed to process the request.
401	Authentication failed.
500	Internal error.

## 4.16 Checking the Existence of an Account Used for Image Sharing

#### Function

Check whether an account used for image sharing exists.

#### URI

GET /v2/manage/namespaces/{*namespace*}/repositories/{*repository*}/access-domains/{*access\_domain*}

For details about parameters, see [Table 4-43](#).

**Table 4-43** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
access_domain	Yes	String	Image sharing account name

## Request

N/A

## Response

**Table 4-44** Response body parameter description

Parameter	Mandatory	Type	Description
exist	Yes	Boolean	<b>true</b> : Exist. <b>false</b> : Not exist.

### Response example

```
{
  "exist":false
}
```

## Status Code

For details about status code, see [Table 4-45](#).

**Table 4-45** Status code

Status Code	Description
200	Query succeeded.
400	The server failed to process the request.
401	Authentication failed.
500	Internal error.

## 4.17 Obtaining the List of Accounts Used for Image Sharing

### Function

Obtain the list of accounts used for image sharing.

### URI

GET /v2/manage/namespaces/{*namespace*}/repositories/{*repository*}/access-domains

For details about parameters, see [Table 4-46](#).

**Table 4-46** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

### Request

N/A

### Response

#### Response parameters

**Table 4-47** Response body parameter description

Parameter	Type	Description
namespace	String	Organization.
repository	String	Image repository.
access_domain	String	Name of the account used for image sharing.
permit	String	Permission.
deadline	String	Expiration time.
description	String	Description.
creator_id	String	Creator ID.
creator_name	String	Name of the creator.

Parameter	Type	Description
created	String	Time when an image is created. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
updated	String	Time when an image is updated. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
status	Boolean	Status. <b>true</b> : valid. <b>false</b> : expired.

### Response example

```
[
  {
    "namespace": "group",
    "repository": "paas",
    "access_domain": "swr",
    "permit": "read",
    "deadline": "2018-10-01T16:00:00.000Z",
    "description": "description",
    "creator_id": "fb3f175c1fd146ab8cdae3272be6107b",
    "creator_name": "group",
    "created": "2020-07-08T14:12:23Z",
    "updated": "2020-07-13T21:01:11Z",
    "status": false
  }
]
```

### Status Code

For details about status code, see [Table 4-48](#).

**Table 4-48** Status code

Status Code	Description
200	The list of image sharing account is queried successfully.
400	Request error. Error information is returned.
401	Authentication failed.

Status Code	Description
500	Internal error. Error information is returned.

## 4.18 Viewing the List of Shared Images

### Function

View the list of shared images.

### URI

GET /v2/manage/shared-repositories?filter=center::{*center*}|name::{*name*}|limit::{*limit*}|offset::{*offset*}

For details about parameters, see [Table 4-49](#).

**Table 4-49** Parameter description

Parameter	Mandatory	Type	Description
center	Yes	String	<b>self</b> : my shared images <b>thirdparty</b> : third-party shared images
name	No	String	Image repository name.
limit	No	String	Number of returned records Ensure that the <b>offset</b> and <b>limit</b> parameters are used together.
offset	No	String	Start index. Ensure that the <b>offset</b> and <b>limit</b> parameters are used together.

### Request

N/A

### Response

#### Response parameters

**Table 4-50** Response header parameter description

Parameter	Type	Description
Content-Range	String	Offset (Start index)–Count (Number of records on the current page)/Total (Total number of records)

 NOTE

If the **offset** and **limit** parameters are transferred in the request, **Content-Range** will be added to the response header.

**Table 4-51** Response body parameter description

Parameter	Type	Description
name	String	Repository name.
category	String	Repository type. The value can be <b>app_server</b> , <b>linux</b> , <b>framework_app</b> , <b>database</b> , <b>lang</b> , <b>other</b> , <b>windows</b> , or <b>arm</b> .
description	String	Repository description.
size	Integer	Image repository size.
is_public	Boolean	Whether the image is a public image. The value can be <b>true</b> or <b>false</b> .
num_images	Integer	Number of image tags in a repository.
num_download	Integer	Download times.
created_at	String	Time when a repository is created. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
updated_at	String	Time when a repository is updated. It is the UTC standard time. Users need to calculate the offset based on the local time. For example, GMT+8 is 8 hours ahead the GMT time.
url	String	(Reserved field) URL of the repository logo.
path	String	External container image download address. For example, <b>swr.cn-north-1.myhuaweicloud.com/namespace/repository</b> .
internal_path	String	Internal container image download address. For example, <b>10.125.0.198:20202/namespace/repository</b> .
domain_name	String	Account name.
namespace	String	Organization name.

Parameter	Type	Description
tags	Array of strings	Image tag list.
status	Boolean	Query the images shared by a third party: Check whether the sharing is expired.  Query the images shared by me: The default value is <b>false</b> , indicating that no images are shared.

### Response example

```
[
  {
    "name": "group",
    "category": "other",
    "description": "",
    "size": 115645792,
    "is_public": false,
    "num_images": 1,
    "num_download": 0,
    "created_at": "2018-03-21T12:22:37Z",
    "updated_at": "2018-03-21T12:22:37Z",
    "logo": "",
    "url": "",
    "path": "swr.cn-north-1.myhuaweicloud.com/namespace/group",
    "internal_path": "10.125.0.198:20202/namespace/repository",
    "domain_name": "domian",
    "namespace": "namespace",
    "tags": [
      "latest"
    ],
    "status": true
  }
]
```

## Status Code

For details about status code, see [Table 4-52](#).

**Table 4-52** Status code

Status Code	Description
200	The request is successfully sent.
400	Request error. Error information is returned.
401	Authentication failed.
500	Internal error. Error information is returned.

## 4.19 Querying All API Versions

### Function

Query all available API versions of SWR.

### URI

GET /

### Request

N/A

### Response

#### Response parameters

**Table 4-53** Response body parameter description

Parameter	Type	Description
versions	Array of objects	A list of objects related to the version. For details, see <a href="#">Table 4-54</a> .

**Table 4-54** versions parameter description

Parameter	Type	Description
id	String	Version ID (version number), for example, v2.
links	String	API URL. For details, see <a href="#">Table 4-55</a> .
version	String	If the APIs of this version support microversions, set this parameter to the supported maximum microversion. If the microversion is not supported, leave this parameter blank.
status	String	Version status. The options are as follows: <b>CURRENT</b> : The version is the primary version. <b>SUPPORTED</b> : The version is an earlier version, but it is still supported. <b>DEPRECATED</b> : The version is a deprecated version, which may be deleted later.

Parameter	Type	Description
updated	String	Version release time, which must be the Coordinated Universal time (UTC) time. For example, the release time of v1 is 2014-06-28T12:20:21Z.
min_version	String	If APIs of this version support microversions, set this parameter to the supported minimum microversion. If the microversion is not supported, leave this parameter blank.

**Table 4-55** links parameter description

Parameter	Type	Description
href	String	Link
rel	String	Description

### Response example

```
{
  "versions": [
    {
      "id": "v2",
      "links": {
        "href": "https://swr-api.xxxx.myhuaweicloud.com/v2/",
        "rel": "self"
      }
    }
  ],
  "min_version": "2.0",
  "status": "CURRENT",
  "updated": "2020-07-09T00:00:00Z",
  "version": "2.26"
}
```

## Status Code

For details about status code, see [Table 4-56](#).

**Table 4-56** Status code

Status Code	Description
200	Request successful.
500	Failed to complete the request because of an internal service error.

## 4.20 Querying a Specific API Version

### Function

Query a specific API version of SWR.

### URI

GET /{api\_version}

For details about parameters, see [Table 4-57](#).

**Table 4-57** Parameter description

Parameter	Type	Mandatory	Description
api_version	String	Yes	API version

### Request

N/A

### Response

#### Response parameters

**Table 4-58** Response body parameter description

Parameter	Type	Description
version	Object	A list of objects related to the version. For details, see <a href="#">Table 4-59</a> .

**Table 4-59** version parameter description

Parameter	Type	Description
id	String	Version ID (version number), for example, v2.
links	String	API URL. For details, see <a href="#">Table 4-60</a> .
version	String	If the APIs of this version support microversions, set this parameter to the supported maximum microversion. If the microversion is not supported, leave this parameter blank.

Parameter	Type	Description
status	String	Version status. The options are as follows: <b>CURRENT</b> : The version is the primary version. <b>SUPPORTED</b> : The version is an earlier version, but it is still supported. <b>DEPRECATED</b> : The version is a deprecated version, which may be deleted later.
updated	String	Version release time, in Coordinated Universal time (UTC). For example, the release time of v1 is 2014-06-28T12:20:21Z.
min_version	String	If APIs of this version support microversions, set this parameter to the supported minimum microversion. If the microversion is not supported, leave this parameter blank.

**Table 4-60** links parameter description

Parameter	Type	Description
href	String	Link
rel	String	Description

#### Response example

```
{
  "version": {
    "id": "v2",
    "links": {
      {
        "href": "https://swr-api.xxxx.myhuaweicloud.com/v2/",
        "rel": "self"
      }
    }
  },
  "min_version": "2.0",
  "status": "CURRENT",
  "updated": "2017-12-09T00:00:00Z",
  "version": "2.26"
}
```

## Status Code

For details about status code, see [Table 4-61](#).

**Table 4-61** Status code

Status Code	Description
200	Request successful.

Status Code	Description
500	Failed to complete the request because of an internal service error.

## 4.21 Creating Organization Permissions

### Function

Create organization permissions.

### URI

POST /v2/manage/namespaces/{*namespace*}/access

**Table 4-62** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

### Request

#### Request parameters

**Table 4-63** Request body parameter description

Parameter	Mandatory	Type	Description
namespace_auth_array	Yes	Array of objects	Array of organization permissions to be created. For details, see <a href="#">Table 4-64</a> .

**Table 4-64** namespace\_auth\_array parameter description

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID, which needs to be obtained from the Identity and Access Management (IAM) service.
user_name	Yes	String	Username, which needs to be obtained from the IAM service.

Parameter	Mandatory	Type	Description
auth	Yes	Integer	User permission. 7: Manage. 3: Write. 1: Read.

#### Request example

```
[
  {
    "user_id": "fb3f175c1fd146ab8cdae3272be6107b",
    "user_name": "user01",
    "auth": 1
  }
]
```

#### Response

##### Response example

```
{}
```

#### Status Code

Status Code	Description
201	Created successfully.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
409	The organization permission already exists.
500	Internal error.

## 4.22 Deleting Organization Permissions

#### Function

Delete organization permissions.

#### URI

DELETE /v2/manage/namespaces/{*namespace*}/access

**Table 4-65** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

## Request

### Request parameters

**Table 4-66** Request body parameter description

Parameter	Mandatory	Type	Description
user_id_array	Yes	Array of strings	ID array of users whose permissions need to be deleted.

### Request example

```
["fb3f175c1fd146ab8cdae3272be6107b"]
```

## Response

### Response example

```
N/A
```

## Status Code

Status Code	Description
204	Deletion succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Internal error.

## 4.23 Updating Organization Permissions

### Function

Update organization permissions.

## URI

PATCH /v2/manage/namespaces/{*namespace*}/access

**Table 4-67** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

## Request

### Request parameters

**Table 4-68** Request body parameter description

Parameter	Mandatory	Type	Description
namespace_auth_array	Yes	Array of objects	Array of organization permissions to be updated. For details, see <a href="#">Table 4-69</a> .

**Table 4-69** namespace\_auth\_array parameter description

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID, which needs to be obtained from the Identity and Access Management (IAM) service.
user_name	Yes	String	Username, which needs to be obtained from the IAM service.
auth	Yes	Integer	User permission. <b>7</b> : Manage. <b>3</b> : Write. <b>1</b> : Read.

### Request example

```
[
  {
    "user_id": "fb3f175c1fd146ab8cdae3272be6107b",
    "user_name": "user01",
    "auth": 7
  }
]
```

## Response

### Response example

N/A

## Status Code

Status Code	Description
201	Update succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Internal error.

## 4.24 Querying Organization Permissions

### Function

Query organization permissions.

### URI

GET /v2/manage/namespaces/{*namespace*}/access

**Table 4-70** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

### Request

#### Request example

N/A

### Response

#### Response parameters

**Table 4-71** Response body parameters

Parameter	Type	Description
id	Integer	ID.
name	String	Organization name.

Parameter	Type	Description
creator_name	String	Organization creator.
self_auth	Object	Permissions of the current user. For details, see <a href="#">Table 4-72</a> .
others_auths	Array of objects	Permissions of other users. For details, see <a href="#">Table 4-73</a> .

**Table 4-72** self\_auth parameter description

Parameter	Type	Description
user_id	String	User ID.
user_name	String	Username.
auth	Integer	User permission. <b>7</b> : Manage. <b>3</b> : Write. <b>1</b> : Read.

**Table 4-73** others\_auths parameter description

Parameter	Type	Description
user_id	String	User ID.
user_name	String	Username.
auth	Integer	User permission. <b>7</b> : Manage. <b>3</b> : Write. <b>1</b> : Read.

### Response example

```
{
  "id": 1422,
  "name": "test",
  "creator_name": "user01",
  "self_auth": {
    "user_id": "3059e6b5562241fda3fa441cca6f228b",
    "user_name": "user",
    "auth": 7
  },
  "others_auths": [
    {
      "user_id": "fb3f175c1fd146ab8cdae3272be6107b",
      "user_name": "user01",
      "auth": 7
    }
  ]
}
```

## Status Code

Status Code	Description
200	Request successful.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Internal error.

## 4.25 Creating Image Permissions

### Function

Create image permissions.

### URI

#### URI format

POST `/v2/manage/namespaces/{namespace}/repos/{repository}/access`

#### Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image name

### Request

#### Request parameters

Table 4-74 Request header parameter description

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Enter <b>application/json;charset=utf8</b> for this field.
X-Auth-Token	Yes	String	User token, which needs to be obtained from the Identity and Access Management (IAM) service.

**Table 4-75** Request body parameter description

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID, which needs to be obtained from the IAM service.
user_name	Yes	String	Username, which needs to be obtained from the IAM service.
auth	Yes	Integer	User permission that is configured.

**Request example**

```
[
  {
    "user_id": "fb3f175c1fd146ab8cdae3272be6107b",
    "user_name": "liugenping01",
    "auth": 1
  }
]
```

**Response**

N/A

**Status Code**

Status Code	Description
201	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The image does not exist.
500	Internal error.

## 4.26 Deleting Image Permissions

**Function**

Delete image permissions.

**URI**

**URI format**

DELETE /v2/manage/namespaces/{*namespace*}/repos/{*repository*}/access

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image name

## Request

### Request parameters

**Table 4-76** Request header parameter description

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Enter <b>application/json;charset=utf8</b> for this field.
X-Auth-Token	Yes	String	User token, which needs to be obtained from the Identity and Access Management (IAM) service.

**Table 4-77** Request body parameter description

Parameter	Mandatory	Type	Description
-	Yes	Array of String	User ID, which needs to be obtained from the IAM service.

### Request example

```
["fb3f175c1fd146ab8cdae3272be6107b"]
```

## Response

N/A

## Status Code

Status Code	Description
204	Deleted successfully.
400	Request error.

Status Code	Description
401	Authentication failed.
404	The image does not exist.
500	Internal error.

## 4.27 Updating Image Permissions

### Function

Update image permissions.

### URI

#### URI format

PATCH /v2/manage/namespaces/{*namespace*}/repos/{*repository*}/access

#### Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image name

### Request

#### Request parameters

Request header parameter description

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Enter <b>application/json;charset=utf8</b> for this field.
X-Auth-Token	Yes	String	User token, which needs to be obtained from the Identity and Access Management (IAM) service.

Request body parameter description

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID, which needs to be obtained from the IAM service.
user_name	Yes	String	Username, which needs to be obtained from the IAM service.
auth	Yes	Integer	User permission that is configured.

#### Request example

```
[
  {
    "user_id": "fb3f175c1fd146ab8cdae3272be6107b",
    "user_name": "liugenping01",
    "auth": 7
  }
]
```

#### Response

N/A

#### Status Code

Status Code	Description
201	Updated successfully.
400	Request error.
401	Authentication failed.
404	The image or the image permission does not exist.
500	Internal error.

## 4.28 Querying Image Permissions

### Function

Query image permissions.

### URI

#### URI format

GET /v2/manage/namespaces/{namespace}/repos/{repository}/access

#### Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image name

## Request

N/A

## Response

### Response parameters

**Table 4-78** Response body parameter description

Parameter	Type	Description
id	Integer	Permission ID
name	String	Image name
self_auth	Object	{ "user_id": "3059e6b5562241fda3fa441cca6f228b", "user_name": "liugenping", "auth": 7 }
others_auth	Object	[ { "user_id": "fb3f175c1fd146ab8cdae3272be6107b", "user_name": "liugenping01", "auth": 7 } ]

### Response example

```
{
  "id": 1422,
  "name": "group",
  "self_auth": {
    "user_id": "3059e6b5562241fda3fa441cca6f228b",
    "user_name": "liugenping",
    "auth": 7
  },
  "others_auths": [
    {
      "user_id": "fb3f175c1fd146ab8cdae3272be6107b",
      "user_name": "liugenping01",
      "auth": 7
    }
  ]
}
```

## Status Code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The image or the image permission does not exist.
500	Internal error.

## 4.29 Obtaining Information About an Image Synchronization Task

### Function

Obtain information about an image synchronization task.

### URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/sync\_job?  
filter=limit::{limit}|offset::{offset}|order::{order}

[Table 4-79](#) describes the parameters.

**Table 4-79** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.
repository	Yes	String	Image repository name.
limit	Yes	String	Number of returned records. Ensure that the <b>limit</b> and <b>offset</b> parameters are used together.
offset	Yes	String	Start index. Ensure that the <b>limit</b> and <b>offset</b> parameters are used together.
order	No	String	Sorting type. The value can be <b>desc</b> or <b>asc</b> , indicating the results are sorted by creation time in descending or ascending order respectively.

## Request

### Request parameters

NA

## Response

### Response parameters

**Table 4-80** Response header parameter description

Parameter	Type	Description
Content-Range	String	Offset (Start index)–Count (Number of records on the current page)/Total (Total number of records)

#### NOTE

If the **offset** and **limit** parameters are transferred in the request, **Content-Range** will be added to the response header.

**Table 4-81** Response body parameter description

Parameter	Type	Description
createdAt	String	Creation time
domainID	String	Account ID
domainName	String	Account name
id	Integer	ID
namespace	String	Organization name
override	Bool	Whether to overwrite
remoteNamespace	String	Target organization
remoteRegionId	String	Target region
repoName	String	Image repository
status	String	Synchronization status. The value can be <b>waiting</b> , <b>running</b> , <b>success</b> , <b>failed</b> , <b>timeout</b> , <b>cancel</b> , or <b>existed</b> .
syncOperatorId	String	Operator account ID
syncOperatorName	String	Operator account name

Parameter	Type	Description
tag	String	Image tag
updatedAt	String	Update time

### Response example

```
[
  {
    "createdAt": "2020-03-12T08:11:36.518144Z",
    "domainID": "0657*****f60",
    "domainName": "h*****4",
    "id": 1,
    "namespace": "test",
    "override": false,
    "remoteNamespace": "remtens",
    "remoteRegionId": "cn-north-1",
    "repoName": "testrepo",
    "status": "success",
    "syncOperatorId": "065*****10",
    "syncOperatorName": "h*****4",
    "tag": "8",
    "updatedAt": "2020-03-12T08:11:41.505499Z"
  }
]
```

## Status Code

[Table 4-82](#) describes the status codes of this API.

**Table 4-82** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.30 Obtaining the List of Automatic Image Synchronization Tasks

### Function

Obtain the list of automatic image synchronization tasks.

## URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/sync\_repo

[Table 4-83](#) describes the parameters.

**Table 4-83** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

## Request

### Request parameters

NA

## Response

### Response parameters

**Table 4-84** Response body parameter description

Parameter	Type	Description
createdAt	String	Creation time
domainID	String	Account ID
domainName	String	Account name
id	Integer	ID
namespace	String	Organization name
override	Bool	Whether to overwrite
remoteNamespace	String	Target organization
remoteRegionId	String	Target region
repoName	String	Repository name
syncAuto	Bool	Automatic synchronization
updatedAt	String	Update time

### Response example

```
[
  {
    "createdAt": "2020-04-17T02:11:03.53183Z",
    "domainID": "8b1e81*****633fb",
    "domainName": "p*****59",
    "id": 45,
    "namespace": "test",
    "override": true,
    "remoteNamespace": "sdsad",
    "remoteRegionId": "cn-north-1",
    "repoName": "grpc-helloworld",
    "syncAuto": true,
    "updatedAt": "2020-04-17T02:11:03.531831Z"
  }
]
```

## Status Code

[Table 4-85](#) describes the status codes of this API.

**Table 4-85** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization or the repository does not exist.
500	Failed to complete the request because of an internal service error.

## 4.31 Creating an Automatic Image Synchronization Task

### Function

Create an automatic image synchronization task.

### URI

POST /v2/manage/namespaces/{namespace}/repos/{repository}/sync\_repo

[Table 4-86](#) describes the parameters.

**Table 4-86** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

Parameter	Mandatory	Type	Description
repository	Yes	String	Image repository name

## Request

### Request parameters

**Table 4-87** Request body parameter description

Parameter	Mandatory	Type	Description
remoteRegionId	Yes	String	Target region ID
remoteNamespace	Yes	String	Target organization
syncAuto	No	Bool	(Reserved field) Automatic synchronization. You are recommended to retain the default value <b>false</b> .
override	No	Bool	Whether to overwrite. The default value is <b>false</b> , which indicates not to overwrite any nonidentical image that has the same name in the target organization.

### Request example

```
{
  "remoteRegionId": "cn-north-208",
  "remoteNamespace": "sdsad",
  "syncAuto": true,
  "override": true
}
```

## Response

NA

## Status Code

[Table 4-88](#) describes the status codes of this API.

**Table 4-88** Status code

Status Code	Description
200	Operation succeeded.

Status Code	Description
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.32 Deleting an Automatic Image Synchronization Task

### Function

Delete an automatic image synchronization task.

### URI

DELETE /v2/manage/namespaces/{namespace}/repos/{repository}/sync\_repo

[Table 4-89](#) describes the parameters.

**Table 4-89** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

### Request

#### Request parameters

**Table 4-90** Request body parameter description

Parameter	Mandatory	Type	Description
remoteRegionId	Yes	String	Target region
remoteNamespace	Yes	String	Target organization

### Request example

```
{  
  "remoteRegionId": "cn-north-208",  
  "remoteNamespace": "sdsad"  
}
```

### Response

NA

### Status Code

[Table 4-91](#) describes the status codes of this API.

**Table 4-91** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.33 Manually Synchronizing an Image

### Function

Manually synchronize an image.

### URI

POST /v2/manage/namespaces/{namespace}/repos/{repository}/sync\_images

[Table 4-92](#) describes the parameters.

**Table 4-92** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

## Request

### Request parameters

**Table 4-93** Request body parameter description

Parameter	Mandatory	Type	Description
imageTag	Yes	Array of String	Image tag list.
override	No	Bool	Whether to overwrite. The default value is <b>false</b> .
remoteNamespace	Yes	String	Target organization.
remoteRegionId	Yes	String	Target region.

### Request example

```
{
  "imageTag": [
    "v1"
  ],
  "override": true,
  "remoteNamespace": "sdsad",
  "remoteRegionId": "cn-north-208"
}
```

## Response

### Response parameters

**Table 4-94** Response body parameter description

Parameter	Type	Description
-	Array of String	Image tags that fail to be synchronized

### Response example

```
[]
```

## Status Code

[Table 4-95](#) describes the status codes of this API.

**Table 4-95** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.34 Creating a Trigger

### Function

Create a trigger.

### URI

POST /v2/manage/namespaces/{namespace}/repos/{repository}/triggers

[Table 4-96](#) describes the parameters.

**Table 4-96** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Repository name

### Request

#### Request parameters

**Table 4-97** Request body parameter description

Parameter	Mandatory	Type	Description
action	Yes	String	Action to trigger. The value is <b>update</b> .
app_type	Yes	String	Application type. The value can be <b>deployments</b> or <b>statefulsets</b> .
application	Yes	String	Application name.

Parameter	Mandatory	Type	Description
cluster_id	No	String	Cluster ID. If <b>trigger_mode</b> is set to <b>cci</b> , leave this parameter blank.
cluster_name	No	String	Cluster name. If <b>trigger_mode</b> is set to <b>cci</b> , leave this parameter blank.
cluster_ns	Yes	String	Namespace where the application is located.
condition	Yes	String	Trigger condition. When <b>trigger_type</b> is set to <b>all</b> , set this parameter to <b>.*</b> . When <b>trigger_type</b> is set to <b>tag</b> , set this parameter to specific image tags separated by semicolons (;). When <b>trigger_type</b> is set to <b>regular</b> , set this parameter to a regular expression.
container	No	String	Name of the container to be updated. By default, all containers using this image are updated.
enable	Yes	String	Whether to enable the trigger.
name	Yes	String	Trigger name.
trigger_mode	No	String	Trigger type. The value can be <b>cce</b> or <b>cci</b> . The default value is <b>cce</b> .
trigger_type	Yes	String	Trigger condition. The value can be <b>all</b> , <b>tag</b> , or <b>regular</b> .

### Request example

```
{
  "action": "update",
  "app_type": "deployments",
  "application": "adsfa",
  "cluster_id": "5b28c12f-dc10-11e9-97b3-0255ac1001b5",
  "cluster_name": "test",
  "cluster_ns": "default",
  "condition": ".*",
  "container": "",
  "enable": "true",
  "name": "test",
  "trigger_mode": "cce",
  "trigger_type": "all"
}
```

### Response

```
{}
```

## Status Code

[Table 4-98](#) describes the status codes of this API.

**Table 4-98** Status code

Status Code	Description
201	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.35 Obtaining the Trigger List of a Specific Image Repository

### Function

Obtain the trigger list of a specific image repository.

### URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/triggers

[Table 4-99](#) describes the parameters.

**Table 4-99** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Repository name

### Request

NA

### Response

**Response parameters**

**Table 4-100** Response body parameter description

Parameter	Type	Description
action	String	Action to trigger. The value is <b>update</b> .
app_type	String	Application type. The value can be <b>deployments</b> or <b>statefulsets</b> .
application	String	Application name.
cluster_id	String	Cluster ID. If <b>trigger_mode</b> is set to <b>cci</b> , leave this parameter blank.
cluster_name	String	Cluster name. If <b>trigger_mode</b> is set to <b>cci</b> , leave this parameter blank.
cluster_ns	String	Namespace where the application is located.
condition	String	Trigger condition. When <b>trigger_type</b> is set to <b>all</b> , set this parameter to <b>.*</b> . When <b>trigger_type</b> is set to <b>tag</b> , set this parameter to specific image tags separated by semicolons (;). When <b>trigger_type</b> is set to <b>regular</b> , set this parameter to a regular expression.
container	String	Name of the container to be updated. By default, all containers using this image are updated.
created_at	String	Creation time.
creator_name	String	Creator.
enable	String	Whether to enable the trigger.
name	String	Trigger name.
trigger_history	Object	(Reserved field) This parameter is not in use.
trigger_mode	String	Trigger type. The value can be <b>cce</b> or <b>cci</b> .
trigger_type	String	Trigger condition. The value can be <b>all</b> , <b>tag</b> , or <b>regular</b> .

**Response example**

```
[
  {
    "action": "update",
    "app_type": "deployments",
    "application": "abcde",
    "cluster_id": "*****",
    "cluster_name": "test123",
    "cluster_ns": "default",
    "condition": ".*",
```

```
"container": "",  
"created_at": "2020-04-16T02:04:55.922236Z",  
"creator_name": "*****",  
"enable": "true",  
"name": "test1",  
"trigger_history": null,  
"trigger_mode": "cce",  
"trigger_type": "all"  
}  
]
```

## Status Code

[Table 4-101](#) describes the status codes of this API.

**Table 4-101** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.36 Deleting a Trigger

### Function

Delete a trigger.

### URI

DELETE /v2/manage/namespaces/{namespace}/repos/{repository}/triggers/{trigger}

[Table 4-102](#) describes the parameters.

**Table 4-102** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Repository name
trigger	Yes	String	Trigger name

## Request

NA

## Response

NA

## Status Code

[Table 4-103](#) describes the status codes of this API.

**Table 4-103** Status code

Status Code	Description
204	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.37 Obtaining Information About a Trigger

### Function

Obtain information about a trigger.

### URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/triggers/{trigger}

[Table 4-104](#) describes the parameters.

**Table 4-104** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Repository name
trigger	Yes	String	Trigger name

## Request

NA

## Response

### Response parameters

**Table 4-105** Response body parameter description

Parameter	Type	Description
action	String	Action to trigger. The value can be <b>update</b> .
app_type	String	Application type. The value can be <b>deployments</b> or <b>statefulsets</b> .
application	String	Application name.
cluster_id	String	Cluster ID. If <b>trigger_mode</b> is set to <b>cci</b> , leave this parameter blank.
cluster_name	String	Cluster name. If <b>trigger_mode</b> is set to <b>cci</b> , leave this parameter blank.
cluster_ns	String	Namespace where the application is located.
condition	String	Trigger condition. When <b>trigger_type</b> is set to <b>all</b> , set this parameter to <b>.*</b> . When <b>trigger_type</b> is set to <b>tag</b> , set this parameter to specific image tags separated by semicolons (;). When <b>trigger_type</b> is set to <b>regular</b> , set this parameter to a regular expression.
container	String	Name of the container to be updated. By default, all containers using this image are updated.
created_at	String	Creation time.
creator_name	String	Creator.
enable	String	Whether to enable the trigger.
name	String	Trigger name.
trigger_history	Array of object	For details, see <a href="#">Table 4-106</a> .
trigger_mode	String	Trigger type. The value can be <b>cce</b> or <b>cci</b> .
trigger_type	String	Trigger condition. The value can be <b>all</b> , <b>tag</b> , or <b>regular</b> .

**Table 4-106** Parameter description

Parameter	Type	Description
action	String	Action to trigger. The value can be <b>update</b> .
app_type	String	Application type. The value can be <b>deployments</b> or <b>statefulsets</b> .
application	String	Application name.
cluster_id	String	(Reserved field) Cluster ID.
cluster_name	String	(Reserved field) Cluster name.
cluster_ns	String	(Reserved field) Namespace where the application is located.
condition	String	Trigger condition. When <b>trigger_type</b> is set to <b>all</b> , set this parameter to <b>.*</b> . When <b>trigger_type</b> is set to <b>tag</b> , set this parameter to specific image tags separated by semicolons (;). When <b>trigger_type</b> is set to <b>regular</b> , set this parameter to a regular expression.
container	String	Name of the container to be updated. By default, all containers using this image are updated.
created_at	String	Creation time.
creator_name	String	Creator.
detail	String	Details.
result	String	Update result. The value can be <b>success</b> or <b>failed</b> .
tag	String	Image tag that triggers the update.

**Response example**

```
{
  "action": "update",
  "app_type": "deployments",
  "application": "cci-deployment-20204161",
  "cluster_id": "",
  "cluster_name": "",
  "cluster_ns": "cci-namespace-21539491",
  "condition": ".*",
  "container": "",
  "created_at": "2020-04-16T01:35:08.639589Z",
  "creator_name": "paas_swr_z00232859",
  "enable": "true",
  "name": "test",
  "trigger_history": [
    {
```

```

    "action": "update",
    "app_type": "deployments",
    "application": "cci-deployment-20204161",
    "cluster_id": "",
    "cluster_name": "",
    "cluster_ns": "",
    "condition": ".*",
    "container": "",
    "created_at": "2020-04-16T01:36:36.155119Z",
    "creator_name": "*****",
    "detail": "xxxxxxxxxxxx",
    "result": "failed",
    "tag": "test"
  }
],
"trigger_mode": "cci",
"trigger_type": "all"
}

```

## Status Code

[Table 4-107](#) describes the status codes of this API.

**Table 4-107** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.38 Updating Trigger Settings

### Function

Update trigger settings.

### URI

PATCH /v2/manage/namespaces/{namespace}/repos/{repository}/triggers/{trigger}

[Table 4-108](#) describes the parameters.

**Table 4-108** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

Parameter	Mandatory	Type	Description
repository	Yes	String	Repository name
trigger	Yes	String	Trigger name

## Request

### Request parameters

**Table 4-109** Request body parameter description

Parameter	Mandatory	Type	Description
enable	Yes	String	Whether to enable the trigger. The value can be <b>true</b> , which indicates enabling the trigger, and <b>false</b> , which indicates disabling the trigger.

### Request example

```
{
  "enable": "true"
}
```

## Response

```
{}
```

## Status Code

[Table 4-110](#) describes the status codes of this API.

**Table 4-110** Status code

Status Code	Description
201	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.39 Adding an Image Retention Policy

### Function

Add an image retention policy.

### URI

POST /v2/manage/namespaces/{namespace}/repos/{repository}/retentions

[Table 4-111](#) describes the parameters.

**Table 4-111** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

### Request

#### Request parameters

**Table 4-112** Request body parameter description

Parameter	Mandatory	Type	Description
algorithm	Yes	String	Retention policy matching rule. The value is <b>or</b> .
rules	Yes	Array of object	For details, see <a href="#">Table 4-113</a> .

**Table 4-113** rules parameter description

Parameter	Mandatory	Type	Description
params	Yes	Object	If <b>template</b> is set to <b>date_rule</b> , set <b>params["days"]</b> . If <b>template</b> is set to <b>tag_rule</b> , set <b>params["num"]</b> .

Parameter	Mandatory	Type	Description
tag_selectors	Yes	Array of object	Image tags that are not counted in the retention policy. For details, see <a href="#">Table 4-114</a> .
template	Yes	String	Retention policy type. The value can be <b>date_rule</b> or <b>tag_rule</b> .

**Table 4-114** tag\_selectors parameter description

Parameter	Mandatory	Type	Description
kind	Yes	String	Matching rule. The value can be <b>label</b> or <b>regexp</b> .
pattern	Yes	String	If <b>kind</b> is set to <b>label</b> , set this parameter to specific image tags. If <b>kind</b> is set to <b>regexp</b> , set this parameter to a regular expression.

#### Request example

```
{
  "algorithm": "or",
  "rules": [
    {
      "params": {
        "days": "30"
      },
      "tag_selectors": [
        {
          "kind": "label",
          "pattern": "v5"
        },
        {
          "kind": "label",
          "pattern": "1.0.1"
        },
        {
          "kind": "regexp",
          "pattern": "^123$"
        }
      ]
    },
    {
      "template": "date_rule"
    }
  ]
}
```

## Response

### Response parameters

**Table 4-115** Response body parameter description

Parameter	Type	Description
id	Integer	ID

**Response example**

```
{"id": 34}
```

**Status Code**

**Table 4-116** describes the status codes of this API.

**Table 4-116** Status code

Status Code	Description
201	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.40 Obtaining the List of Image Retention Policies

**Function**

Obtain the list of image retention policies.

**URI**

GET /v2/manage/namespaces/{namespace}/repos/{repository}/retentions

**Table 4-117** describes the parameters.

**Table 4-117** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name

## Request

NA

## Response

### Response parameters

**Table 4-118** Response body parameter description

Parameter	Type	Description
algorithm	String	Retention policy matching rule. The value is <b>or</b> .
id	Integer	ID
rules	Array of object	For details, see <a href="#">Table 4-113</a> .
scope	String	Reserved field

### Response example

```
[
  {
    "algorithm": "or",
    "id": 222,
    "rules": [
      {
        "params": {
          "days": "30"
        },
        "tag_selectors": [
          {
            "kind": "label",
            "pattern": "v5"
          },
          {
            "kind": "label",
            "pattern": "1.0.1"
          },
          {
            "kind": "regexp",
            "pattern": "^123$"
          }
        ],
        "template": "date_rule"
      }
    ],
    "scope": ""
  }
]
```

## Status Code

[Table 4-119](#) describes the status codes of this API.

**Table 4-119** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.41 Deleting an Image Retention Policy

### Function

Delete an image retention policy.

### URI

DELETE /v2/manage/namespaces/{namespace}/repos/{repository}/retentions/{retention\_id}

[Table 4-120](#) describes the parameters.

**Table 4-120** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
retention_id	Yes	String	Image retention policy ID

### Request

NA

### Response

{}

### Status Code

[Table 4-121](#) describes the status codes of this API.

**Table 4-121** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.42 Obtaining an Image Retention Policy Record

### Function

Obtain the details of an image retention policy.

### URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/retentions/{retention\_id}

[Table 4-122](#) describes the parameters.

**Table 4-122** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
retention_id	Yes	String	Image retention policy ID

### Request

NA

### Response

**Response parameters**

**Table 4-123** Response body parameter description

Parameter	Type	Description
algorithm	String	Retention policy matching rule. The value is <b>or</b> .
id	Integer	ID
rules	Array of object	For details, see <a href="#">Table 4-113</a> .
scope	String	Reserved field

**Response example**

```
{
  "algorithm": "or",
  "id": 222,
  "rules": [
    {
      "params": {
        "days": "30"
      },
      "tag_selectors": [
        {
          "kind": "label",
          "pattern": "v5"
        },
        {
          "kind": "label",
          "pattern": "1.0.1"
        },
        {
          "kind": "regexp",
          "pattern": "^123$"
        }
      ]
    }
  ],
  "template": "date_rule"
},
"scope": ""
}
```

**Status Code**

[Table 4-124](#) describes the status codes of this API.

**Table 4-124** Status code

Status Code	Description
200	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.

Status Code	Description
500	Failed to complete the request because of an internal service error.

## 4.43 Modifying an Image Retention Policy

### Function

Modify an image retention policy.

### URI

PATCH /v2/manage/namespaces/{namespace}/repos/{repository}/retentions/{retention\_id}

[Table 4-125](#) describes the parameters.

**Table 4-125** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
retention_id	Yes	String	Image retention policy ID

### Request

#### Request parameters

**Table 4-126** Request body parameter description

Parameter	Mandatory	Type	Description
rules	Yes	Array of object	For details, see <a href="#">Table 4-127</a> .

**Table 4-127** rules parameter description

Parameter	Mandatory	Type	Description
params	Yes	Object	If <b>template</b> is set to <b>date_rule</b> , set <b>params["days"]</b> . If <b>template</b> is set to <b>tag_rule</b> , set <b>params["num"]</b> .
tag_selectors	Yes	Array of object	Image tags that are not counted in the retention policy. For details, see <a href="#">Table 4-128</a> .
template	Yes	String	Retention policy type. The value can be set <b>date_rule</b> and <b>tag_rule</b> .

**Table 4-128** tag\_selectors parameter description

Parameter	Mandatory	Type	Description
kind	Yes	String	Matching rule. The value can be <b>label</b> or <b>regexp</b> .
pattern	Yes	String	If <b>kind</b> is set to <b>label</b> , set this parameter to specific image tags. If <b>kind</b> is set to <b>regexp</b> , set this parameter to a regular expression.

**Request example**

```
{
  "rules": [
    {
      "params": {
        "days": "30"
      },
      "tag_selectors": [
        {
          "kind": "label",
          "pattern": "v5"
        },
        {
          "kind": "label",
          "pattern": "1.0.1"
        },
        {
          "kind": "regexp",
          "pattern": "^123$"
        }
      ],
      "template": "date_rule"
    }
  ]
}
```

## Response

{}

## Status Code

[Table 4-129](#) describes the status codes of this API.

**Table 4-129** Status code

Status Code	Description
201	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

## 4.44 Obtaining Image Retention Records

### Function

Obtain image retention records.

### URI

GET /v2/manage/namespaces/{namespace}/repos/{repository}/retentions/  
histories?offset={offset}&limit={limit}

[Table 4-130](#) describes the parameters.

**Table 4-130** Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
offset	No	Integer	Offset
limit	No	Integer	Count

### Request

NA

## Response

### Response parameters

**Table 4-131** Response header parameter description

Parameter	Type	Description
Content-Range	String	Offset (Start index)–Count (Number of records on the current page)/Total (Total number of records)

#### NOTE

If the **offset** and **limit** parameters are transferred in the request, **Content-Range** will be added to the response header.

**Table 4-132** Response body parameter description

Parameter	Type	Description
retention_log	Array of object	For details, see <a href="#">Table 4-133</a> .
total	Integer	Total number.

**Table 4-133** retention\_log parameter description

Parameter	Type	Description
created_at	String	Creation time
id	Integer	ID
namespace	String	Organization name
repo	String	Image repository name
retention_id	Integer	Image retention policy ID
rule_type	String	Policy
tag	String	Image tag

### Response example

```
{
  "retention_log": [
    {
      "created_at": "2020-04-17T08:01:09.658309Z",
      "id": 9088,
      "namespace": "hwstaff_l00283074",
      "repo": "experience_2048",
      "retention_id": 91,

```

```
    "rule_type": "{\"template\":\"tag_rule\",\"params\":{\"num\":\"7\"},\"tag_selectors\":[]}\",  
    "tag": "4"  
  }  
],  
"total": 1  
}
```

## Status Code

[Table 4-134](#) describes the status codes of this API.

**Table 4-134** Status code

Status Code	Description
201	Operation succeeded.
400	Request error.
401	Authentication failed.
404	The organization does not exist.
500	Failed to complete the request because of an internal service error.

# 5 Appendixes

## 5.1 Status Code

For details about status code, see [Table 5-1](#).

**Table 5-1** Status Code

Status Code	Message	Description
100	Continue	The server has received the initial part of the request and the client should continue to send the remaining part.
101	Switching Protocols	The requester has asked the server to switch protocols and the server has agreed to do so. The target protocol must be more advanced than the source protocol.  For example, the current HTTP protocol is switched to a later version of HTTP.
201	Created	The request has been fulfilled, resulting in the creation of a new resource.
202	Accepted	The request has been accepted for processing, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.
204	NoContent	The server has successfully processed the request, but does not return any content.  The status code is returned in response to an HTTP OPTIONS request.

Status Code	Message	Description
205	Reset Content	The server has successfully processed the request, but does not return any content. Unlike a 204 response, this response requires that the requester reset the content.
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	There are multiple options for the requested resource. For example, this code could be used to present a list of resource characteristics and addresses from which the client such as a browser may choose.
301	Moved Permanently	This and all future requests should be permanently directed to the given URI indicated in this response.
302	Found	The requested resource was temporarily moved.
303	See Other	The response to the request can be found under another URI using a GET or POST method.
304	Not Modified	The requested resource has not been modified. In such case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.
305	Use Proxy	The requested resource is available only through a proxy.
306	Unused	This HTTP status code is no longer used.
400	BadRequest	The request is invalid. The client should modify the request instead of re-initiating it.
401	Unauthorized	The authorization information provided by the client is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.
403	Forbidden	The server has received the request and understood it, but the server is refusing to respond to it. The client should modify the request instead of re-initiating it.

Status Code	Message	Description
404	NotFound	The requested resource cannot be found. The client should modify the request instead of re-initiating it.
405	MethodNotAllowed	A request method is not supported for the requested resource. The client should modify the request instead of re-initiating it.
406	Not Acceptable	The server cannot fulfill the request based on the content characteristics of the request.
407	Proxy Authentication Required	This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The server timed out waiting for the request. The client may re-initiate the request without modifications at any later time.
409	Conflict	The request cannot be processed due to a conflict. This status code indicates that the resource that the client attempts to create already exists, or the request fails to be processed because of the update of the conflict request.
410	Gone	The requested resource cannot be found. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refused to process the request because the request does not specify the length of its content.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The server refuses to process a request because the request entity is too large. The server may disable the connection to prevent the client from sending requests consecutively. If the server temporarily cannot process the request, the response will contain a Retry-After header field.
414	Request-URI Too Large	The URI provided was too long for the server to process.

Status Code	Message	Description
415	Unsupported Media Type	The server does not support the media type in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the Expect request-header field.
422	UnprocessableEntity	The request is well-formed but is unable to be processed due to semantic errors.
429	TooManyRequests	The client sends excessive requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server receives excessive requests within a given time (beyond its processing capability). In this case, the client should repeat requests after the time specified in the Retry-After header of the response expires.
500	InternalServerError	The server is able to receive the request but it could not understand the request.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server acting as a gateway or proxy receives an invalid response from a remote server.
503	ServiceUnavailable	The requested service is invalid. The client should modify the request instead of re-initiating it.
504	ServerTimeout	The server could not return a timely response. The response will reach the client only if the request carries a timeout parameter.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

## 5.2 Error Code

If an error occurs in API calling, no result is returned. Identify the causes of errors based on the error codes of each API. If an error occurs in API calling, HTTP status code 4xx or 5xx is returned. The response body contains the specific error code and information. If you are unable to identify the cause of an error, contact customer service and provide the error code so that we can help you solve the problem as soon as possible.

## Format of an Error Response Body

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
[
  {
    "errorCode": "SVCSTG.SWR.4001048",
    "errorMessage": "Failed to get Token from Context",
    "detail": ""
  }
]
```

In the preceding information, **errorCode** indicates error code, **errorMessage** indicates error description, and **detail** provides error details.

## Error Code Description

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [API Gateway Error Codes](#).

Status Code	Error Code	Description
401	SVCSTG.SWR.4010000	Authentication fails.
500	SVCSTG.SWR.5000001	Internal server error: Failed to obtain the application description.
500	SVCSTG.SWR.5000002	Internal server error: Failed to obtain the image information.
500	SVCSTG.SWR.5000003	Internal server error: Failed to obtain the image list.
404	SVCSTG.SWR.4040004	The software package does not exist.
500	SVCSTG.SWR.5000005	Internal server error: Deletion fails.
409	SVCSTG.SWR.4090006	The image already exists.
500	SVCSTG.SWR.5000007	Internal server error: Operation fails.
500	SVCSTG.SWR.5000008	Internal server error: Failed to obtain the list.
404	SVCSTG.SWR.4040009	The image list cannot be found.
404	SVCSTG.SWR.4040010	The image does not exist.
500	SVCSTG.SWR.5000011	Internal server error: Update fails.

Status Code	Error Code	Description
406	SVCSTG.SWR. 4060012	Failed to delete the image repository because it contains images.
400	SVCSTG.SWR. 4000013	Invalid digest.
400	SVCSTG.SWR. 4000014	Invalid parameter.
400	SVCSTG.SWR. 4000015	Invalid package name.
400	SVCSTG.SWR. 4000016	Invalid version number.
500	SVCSTG.SWR. 5000017	Internal server error: The application signature function is disabled.
500	SVCSTG.SWR. 5000018	Internal server error: Failed to read public keys.
500	SVCSTG.SWR. 5000019	Internal server error: Failed to find local files.
500	SVCSTG.SWR. 5000020	Internal server error: Failed to obtain session locks.
500	SVCSTG.SWR. 5000021	Internal server error: Failed to synchronize files.
500	SVCSTG.SWR. 5000022	Internal server error: Failed to synchronize meta files.
409	SVCSTG.SWR. 4090023	The system is busy.
500	SVCSTG.SWR. 5000024	Internal server error: Failed to generate the SHA value.
413	SVCSTG.SWR. 4130025	The file exceeds the maximum size allowed.
500	SVCSTG.SWR. 5000027	Internal server error: Failed to decompress the package.
400	SVCSTG.SWR. 4000028	Incorrect image format.
400	SVCSTG.SWR. 4000029	Incorrect image format: The image name is inconsistent.
500	SVCSTG.SWR. 5000030	Internal server error: Failed to locate the file pointer.
400	SVCSTG.SWR. 4000031	Invalid AK/SK.

Status Code	Error Code	Description
500	SVCSTG.SWR. 5000032	Internal server error: JSON conversion fails.
500	SVCSTG.SWR. 5000033	Internal server error: Failed to save the application file.
500	SVCSTG.SWR. 5000034	Internal server error: Failed to delete the application.
500	SVCSTG.SWR. 5000035	Internal server error: Failed to save the application to the database.
500	SVCSTG.SWR. 5000036	Internal server error: Failed to update the application database.
500	SVCSTG.SWR. 5000037	Internal server error: Failed to create the signature.
500	SVCSTG.SWR. 5000038	Internal server error: Failed to delete the signature.
400	SVCSTG.SWR. 4000039	Invalid session ID.
500	SVCSTG.SWR. 5000040	Failed to update status.
500	SVCSTG.SWR. 5000041	Internal server error: Failed to read the session.
500	SVCSTG.SWR. 5000042	Internal server error: Failed to reclaim the session.
404	SVCSTG.SWR. 4040043	The session does not exist.
500	SVCSTG.SWR. 5000044	Internal server error: Failed to generate the session.
400	SVCSTG.SWR. 4000045	Internal server error: Failed to bind the session.
500	SVCSTG.SWR. 5000046	Internal server error: Creation fails.
400	SVCSTG.SWR. 4000047	This version is not supported.
404	SVCSTG.SWR. 4040048	The image tag does not exist.
500	SVCSTG.SWR. 5000049	Internal server error: Failed to delete the image.
500	SVCSTG.SWR. 5000050	Internal server error: Failed to generate the digest.

Status Code	Error Code	Description
500	SVCSTG.SWR.5000051	Internal server error: Failed to obtain the SHA list.
500	SVCSTG.SWR.5000052	Internal server error: Failed to reduce image references.
406	SVCSTG.SWR.4060053	Deletion fails. Please first delete the software package.
500	SVCSTG.SWR.5000054	Internal server error: Saving fails.
400	SVCSTG.SWR.4000055	No search content.
400	SVCSTG.SWR.4000056	Invalid page size.
500	SVCSTG.SWR.5000057	Internal server error: Failed to obtain the organization.
404	SVCSTG.SWR.4040058	The organization does not exist.
500	SVCSTG.SWR.5000059	Internal server error: Failed to create the organization.
409	SVCSTG.SWR.4090060	The organization already exists.
500	SVCSTG.SWR.5000061	Failed to obtain the IP address of the software instance.
500	SVCSTG.SWR.5000062	Failed to obtain the local IP address of the software.
500	SVCSTG.SWR.5000063	Failed to obtain the tag list.
400	SVCSTG.SWR.4000064	Invalid image: The manifest file contains empty RepoTags.
400	SVCSTG.SWR.4000065	Invalid image: The manifest file does not have layer information.
400	SVCSTG.SWR.4000066	Invalid image: The manifest file does not have configuration information.
400	SVCSTG.SWR.4000067	Invalid image package: The image package does not have a manifest file.
400	SVCSTG.SWR.4000068	Invalid image: Failed to parse the manifest file.
400	SVCSTG.SWR.4000069	Invalid image: The image format is incorrect or not supported.

Status Code	Error Code	Description
507	SVCSTG.SWR. 5070070	Your OBS service is disabled or incorrect. Please first check your OBS service.
500	SVCSTG.SWR. 5000071	Internal server error: Failed to obtain the user list.
500	SVCSTG.SWR. 5000073	Internal server error: Restoration fails.
500	SVCSTG.SWR. 5000075	Internal server error: Failed to update the organization.
409	SVCSTG.SWR. 4090076	The organization permission already exists.
500	SVCSTG.SWR. 5000077	Internal server error: Failed to obtain the organization permission.
404	SVCSTG.SWR. 4040078	The organization permission does not exist.
500	SVCSTG.SWR. 5000079	Internal server error: Failed to update the organization permission.
500	SVCSTG.SWR. 5000080	Internal server error: Failed to create the organization permission.
500	SVCSTG.SWR. 5000081	Internal server error: Failed to delete the organization permission.
500	SVCSTG.SWR. 5000082	Internal server error: Failed to parse the JSON file.
404	SVCSTG.SWR. 4040083	The requested URL cannot be found.
500	SVCSTG.SWR. 5000084	Internal server error: Failed to obtain the service.
500	SVCSTG.SWR. 5000085	Internal server error: Failed to create the tag.
500	SVCSTG.SWR. 5000086	Internal server error: Failed to update the tag.
500	SVCSTG.SWR. 5000087	Internal server error: Failed to obtain the image layer.
500	SVCSTG.SWR. 5000088	Internal server error: Failed to delete the image layer from UDS.
409	SVCSTG.SWR. 4090089	The permission already exists.
500	SVCSTG.SWR. 5000090	Internal server error: Failed to obtain the permission.

Status Code	Error Code	Description
404	SVCSTG.SWR. 4040091	The permission does not exist.
500	SVCSTG.SWR. 5000092	Internal server error: Failed to update the permission.
500	SVCSTG.SWR. 5000093	Internal server error: Failed to create the permission.
400	SVCSTG.SWR. 4000097	The maximum number of organizations has been exceeded.
400	SVCSTG.SWR. 4000098	The maximum number of images has been exceeded.
400	SVCSTG.SWR. 4000099	The maximum number of image tags has been exceeded.
500	SVCSTG.SWR. 5000112	Internal server error. Failed to obtain the image.
404	SVCSTG.SWR. 4040113	The image does not exist.
500	SVCSTG.SWR. 5000114	Internal server error. Failed to obtain the image tag.
404	SVCSTG.SWR. 4040115	The image tag does not exist.
400	SVCSTG.SWR. 4001000	<b>Content-Type</b> must be set to <b>multipart/form-data</b> .
413	SVCSTG.SWR. 4131001	The request exceeds the maximum size allowed.
500	SVCSTG.SWR. 5001002	Internal server error: The external address is disabled.
400	SVCSTG.SWR. 4001003	Failed to obtain the form file.
400	SVCSTG.SWR. 4001004	<b>Type</b> must be set to <b>app</b> .
400	SVCSTG.SWR. 4001005	Failed to parse the form.
400	SVCSTG.SWR. 4001006	Invalid meta file format.
400	SVCSTG.SWR. 4001008	Invalid organization or name.
400	SVCSTG.SWR. 4001009	<b>is_public</b> must be set to <b>true</b> or <b>false</b> .

Status Code	Error Code	Description
400	SVCSTG.SWR. 4001010	<b>Type</b> must be set to <b>app</b> or <b>image</b> .
400	SVCSTG.SWR. 4001011	<b>Type</b> must be set to <b>app</b> , <b>image</b> or left blank.
413	SVCSTG.SWR. 4131012	The description exceeds the maximum length allowed.
400	SVCSTG.SWR. 4001013	<b>ak</b> or <b>sk</b> cannot be left blank.
400	SVCSTG.SWR. 4001014	Mandatory parameters must be set.
400	SVCSTG.SWR. 4001015	The form file cannot be found.
400	SVCSTG.SWR. 4001016	At least <b>is_public</b> and <b>manifests</b> must be set.
400	SVCSTG.SWR. 4001018	Incorrect request body type.
400	SVCSTG.SWR. 4001019	Failed to parse the form file.
500	SVCSTG.SWR. 5001020	Internal server error: Failed to read the form file.
400	SVCSTG.SWR. 4001021	The version number cannot be left blank.
400	SVCSTG.SWR. 4001022	Incorrect file type.
400	SVCSTG.SWR. 4001023	Unsupported file type.
400	SVCSTG.SWR. 4001024	Failed to query parameters.
400	SVCSTG.SWR. 4001025	Invalid parameter.
500	SVCSTG.SWR. 5001026	Failed to obtain the manifest file.
400	SVCSTG.SWR. 4001027	<b>Status</b> must be set to <b>done</b> or <b>error</b> .
400	SVCSTG.SWR. 4001028	Invalid organization name.
400	SVCSTG.SWR. 4001029	Invalid image name.

Status Code	Error Code	Description
400	SVCSTG.SWR. 4001030	Invalid package.
403	SVCSTG.SWR. 4031032	The number of requests exceeds the maximum number allowed.
403	SVCSTG.SWR. 4031033	Unavailable request.
400	SVCSTG.SWR. 4001035	Invalid filtering criteria.
400	SVCSTG.SWR. 4001036	Parameters are lost.
400	SVCSTG.SWR. 4001037	The registration center type is not supported.
500	SVCSTG.SWR. 5001038	The registration center is unavailable.
500	SVCSTG.SWR. 5001039	Search fails.
400	SVCSTG.SWR. 4001041	Invalid <b>domain</b> .
403	SVCSTG.SWR. 4031042	There are too many requests.
403	SVCSTG.SWR. 4031043	Invalid JWT token request.
401	SVCSTG.SWR. 4011044	The image to be uploaded has been created by another user. You do not have the permission to upload the image.
400	SVCSTG.SWR. 4001045	The <b>SWRServer</b> parameter is missing.
500	SVCSTG.SWR. 5001046	Internal server error: Failed to check the organization.
500	SVCSTG.SWR. 5001047	Internal server error: Failed to save the organization.
400	SVCSTG.SWR. 4001048	Failed to obtain the token.
500	SVCSTG.SWR. 5001049	Internal server error: Failed to delete the organization.
500	SVCSTG.SWR. 5001050	Internal server error: Failed to obtain the image layer.

Status Code	Error Code	Description
404	SVCSTG.SWR. 4041051	The image layer does not exist.
500	SVCSTG.SWR. 5001052	Internal server error: Invalid image layer.
500	SVCSTG.SWR. 5001053	Internal server error: Invalid upload of an image layer.
500	SVCSTG.SWR. 5001054	Internal server error: Failed to create the image layer path.
500	SVCSTG.SWR. 5001055	Internal server error: Failed to save the image layer.
404	SVCSTG.SWR. 4041056	The image tag does not exist.
500	SVCSTG.SWR. 5001056	Internal server error: Failed to obtain the tag list.
404	SVCSTG.SWR. 4041057	The tag list does not exist.
500	SVCSTG.SWR. 5001058	Internal server error: Failed to save the image.
500	SVCSTG.SWR. 5001059	Internal server error: Failed to obtain the image.
404	SVCSTG.SWR. 4041060	The manifest file does not exist.
404	SVCSTG.SWR. 4041061	The organization does not exist.
500	SVCSTG.SWR. 5001062	Internal server error: Failed to update the organization.
500	SVCSTG.SWR. 5001063	Internal server error: Failed to delete the image layer.
401	SVCSTG.SWR. 4011064	Insufficient permissions.
400	SVCSTG.SWR. 4001065	Invalid organization permissions.
400	SVCSTG.SWR. 4001066	Invalid permissions.
400	SVCSTG.SWR. 4001067	Invalid username.
500	SVCSTG.SWR. 5001076	Failed to obtain the service token.

Status Code	Error Code	Description
401	SVCSTG.S2I. 4019000	Authentication error: No user token or user AK/SK information found.
403	SVCSTG.SWR. 4039001	You cannot apply for or operate the resources that do not belong to you.
403	SVCSTG.SWR. 4039002	You do not have the permission to perform this operation.
400	SVCSTG.SWR. 4009010	Invalid organization.
400	SVCSTG.SWR. 4009011	Invalid image name.
400	SVCSTG.SWR. 4009012	Invalid code source type.
400	SVCSTG.SWR. 4009013	Invalid ID.
400	SVCSTG.SWR. 4009014	Invalid filtering criteria.
400	SVCSTG.SWR. 4009015	Failed to parse the build task.
400	SVCSTG.SWR. 4009016	Invalid file path.
400	SVCSTG.SWR. 4009017	Invalid source code owner.
400	SVCSTG.SWR. 4009018	Invalid source code repository.
400	SVCSTG.SWR. 4009019	Invalid source code repository ID.
400	SVCSTG.SWR. 4009020	Invalid source code branch.
400	SVCSTG.SWR. 4009021	Invalid image label.
400	SVCSTG.SWR. 4009022	Invalid source code username.
400	SVCSTG.SWR. 4009023	Invalid commit ID.
400	SVCSTG.SWR. 4006030	Failed to execute the build task.
400	SVCSTG.SWR. 4006031	The build record already exists.

Status Code	Error Code	Description
404	SVCSTG.SWR. 4046032	The build record does not exist.
500	SVCSTG.SWR. 5006033	Failed to obtain the build record from the database.
403	SVCSTG.SWR. 4037017	You do not have the permission to edit or manage any organization. Please first create an organization or obtain the edit/management permission of the organization.
403	SVCSTG.SWR. 4037018	You do not have the permission to edit or manage any image in the image repository. Please first create an image or obtain an editable/manageable image.
500	SVCSTG.SWR. 5007019	The maximum number of images have been reached.
403	SVCSTG.SWR. 4037020	You do not have the permission to operate the image repository resource. Please improve your organization permissions.

## 5.3 Obtaining a Project ID

### Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- [Obtain the Project ID by Calling an API](#)
- [Obtain the Project ID from the Console](#)

### Obtain the Project ID by Calling an API

You can obtain the project ID by calling the API used to [query for projects based on specified criteria](#).

The API used to obtain a project ID is GET `https://{Endpoint}/v3/projects`. `{Endpoint}` is the IAM endpoint and can be obtained from [Regions and Endpoints](#). For details about API authentication, see [Authentication](#).

The following is an example response. The value of `id` is the project ID.

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,

```

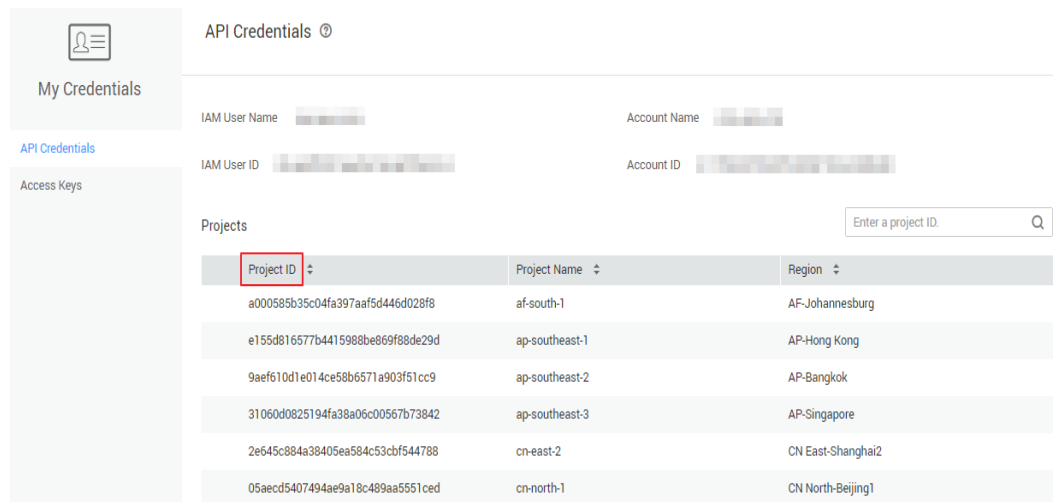
```
"parent_id": "65382450e8f64ac0870cd180d14e684b",
"name": "project_name",
"description": "",
"links": {
  "next": null,
  "previous": null,
  "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
},
"id": "a4a5d4098fb4474fa22cd05f897d6b99",
"enabled": true
}
],
"links": {
  "next": null,
  "previous": null,
  "self": "https://www.example.com/v3/projects"
}
}
```

## Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.  
On the **API Credentials** page, view the project ID in the project list.

**Figure 5-1** Viewing the project ID



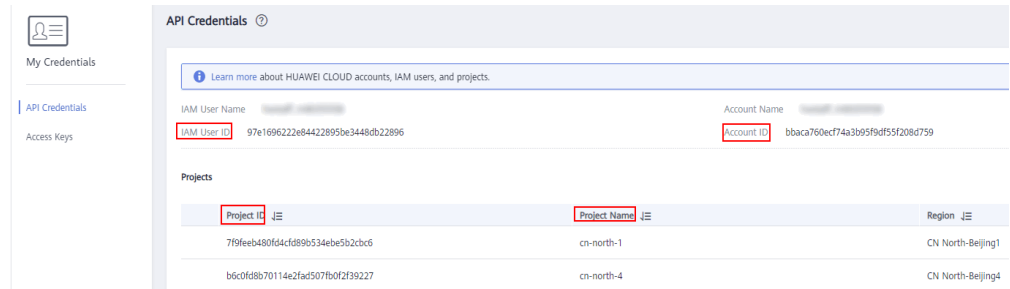
## 5.4 Obtaining an Account ID

An account ID (domain ID) is required for some URLs when an API is called. To obtain an account ID, perform the following operations:

1. Log in to the management console.
2. Click the username in the upper right corner and choose **My Credentials** from the drop-down list.

On the **API Credentials** page, view account ID.

**Figure 5-2** Obtaining your account ID



# 6 Change History

**Table 6-1** Change history

Release Date	Description
2020-10-23	Added <ul style="list-style-type: none"><li>• <a href="#">Creating Image Permissions</a>, <a href="#">Deleting Image Permissions</a>, <a href="#">Updating Image Permissions</a>, and <a href="#">Querying Image Permissions</a></li><li>• <a href="#">Obtaining Information About an Image Synchronization Task</a>, <a href="#">Obtaining the List of Automatic Image Synchronization Tasks</a>, <a href="#">Creating an Automatic Image Synchronization Task</a>, <a href="#">Deleting an Automatic Image Synchronization Task</a>, and <a href="#">Manually Synchronizing an Image</a></li><li>• <a href="#">Creating a Trigger</a>, <a href="#">Obtaining the Trigger List of a Specific Image Repository</a>, <a href="#">Deleting a Trigger</a>, <a href="#">Obtaining Information About a Trigger</a>, and <a href="#">Updating Trigger Settings</a></li><li>• <a href="#">Adding an Image Retention Policy</a>, <a href="#">Obtaining the List of Image Retention Policies</a>, <a href="#">Deleting an Image Retention Policy</a>, <a href="#">Obtaining an Image Retention Policy Record</a>, <a href="#">Modifying an Image Retention Policy</a>, and <a href="#">Obtaining Image Retention Records</a></li></ul>
2020-05-30	Updated <ul style="list-style-type: none"><li>• The description of response parameters in <a href="#">Querying the List of Organizations</a></li><li>• The information of the category parameter in <a href="#">Creating an Image Repository</a>, <a href="#">Updating the Brief Information on an Image Repository</a>, <a href="#">Querying the Brief Information on an Image Repository</a>, <a href="#">Querying the List of Images</a>, and <a href="#">Viewing the List of Shared Images</a></li></ul> Deleted <ul style="list-style-type: none"><li>• The request parameter list in <a href="#">Updating the Brief Information on an Image Repository</a></li></ul>

Release Date	Description
2019-12-30	Added authorization management APIs. <ul style="list-style-type: none"> <li>● <a href="#">Creating Organization Permissions</a></li> <li>● <a href="#">Deleting Organization Permissions</a></li> <li>● <a href="#">Updating Organization Permissions</a></li> <li>● <a href="#">Querying Organization Permissions</a></li> </ul>
2019-08-30	Added APIs used for querying API version information. <ul style="list-style-type: none"> <li>● <a href="#">Querying All API Versions</a></li> <li>● <a href="#">Querying a Specific API Version</a></li> </ul>
2019-06-15	Updated the information of the access_domain parameter. <ul style="list-style-type: none"> <li>● <a href="#">Creating an Account Used for Image Sharing</a></li> <li>● <a href="#">Deleting an Account Used for Image Sharing</a></li> <li>● <a href="#">Updating an Account Used for Image Sharing</a></li> <li>● <a href="#">Checking the Existence of an Account Used for Image Sharing</a></li> <li>● <a href="#">Obtaining the List of Accounts Used for Image Sharing</a></li> </ul>
2018-12-30	Updated parameter information. <ul style="list-style-type: none"> <li>● <a href="#">Creating an Organization</a></li> <li>● <a href="#">Querying the List of Organizations</a></li> <li>● <a href="#">Creating an Image Repository</a></li> <li>● <a href="#">Updating the Brief Information on an Image Repository</a></li> <li>● <a href="#">Querying the Brief Information on an Image Repository</a></li> <li>● <a href="#">Querying the List of Images</a></li> <li>● <a href="#">Creating an Account Used for Image Sharing</a></li> <li>● <a href="#">Updating an Account Used for Image Sharing</a></li> <li>● <a href="#">Checking the Existence of an Account Used for Image Sharing</a></li> <li>● <a href="#">Obtaining the List of Accounts Used for Image Sharing</a></li> </ul>
2018-11-15	Added <ul style="list-style-type: none"> <li>● <a href="#">Viewing the List of Shared Images</a></li> </ul>
2018-04-10	Added APIs used for private image sharing among accounts. <ul style="list-style-type: none"> <li>● <a href="#">Creating an Account Used for Image Sharing</a></li> <li>● <a href="#">Deleting an Account Used for Image Sharing</a></li> <li>● <a href="#">Updating an Account Used for Image Sharing</a></li> <li>● <a href="#">Checking the Existence of an Account Used for Image Sharing</a></li> <li>● <a href="#">Obtaining the List of Accounts Used for Image Sharing</a></li> </ul>
2018-01-26	This issue is the first official release.