SoftWare Repository for Container

API Reference

Issue	1
Date	2021-09-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

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 Introduction in *SWR User Guide*.

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. To obtain regions and endpoints, contact the administrator.

1.4 Constraints

- The number of SWR organizations that you can create is determined by your quota, as shown in **Table 1-1**.

Table 1-1 Quota

Resource	Quota
Number of organizations	5

- For more constraints, see API description.

For details about how to increase quotas, see **Quotas**.

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. For security purposes, 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).

  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. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. 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.

- Enterprise project

Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated.

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.

Organization Management

API	Description
Creating an Organization	Create an organization.
Deleting an Organization	Delete an organization.
Querying the List of Organizations	Query the list of organizations.
Querying the Details of an Organization	Query the details of an organization, including its ID, name, creator, and user permissions.

Image Repository Management

API	Description
Creating an Image Repository in an Organization	Create an image repository in an organization.
Deleting an Image Repository in an Organization	Delete an image repository in an organization.
Updating the Brief Information on an Image Repository	Update the brief information on an image repository in an organization, including the **category**, **is_public**, and **description**.

Querying the Brief Information on an Image Repository	Query the brief information on an image repository in an organization, 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.
Querying the Image Repository List	Query the image repository list.
Viewing the List of Shared Images	View the list of shared images.

Image Tag Management

API	Description
Querying the List of Image Repository Tags	Query all image tags of an image repository.
Deleting the Image with a Specified Tag	Delete the image with a specified tag in an image repository.

Shared Account Management

API	Description
Creating an Account Used for Image Sharing	Create an account used for image sharing. You can share your private images with other users, granting them permissions to pull the images.
Deleting an Account Used for Image Sharing	Delete an account used for image sharing.
Updating an Account Used for Image Sharing	Update an account used for image sharing.
Checking the Existence of an Account Used for Image Sharing	Check whether an account used for image sharing exists.
Obtaining the List of Accounts Used for Image Sharing	Obtain the list of accounts used for image sharing.

API Versions

API	Description

Querying All API Versions	Query all available API versions of SWR.
Querying a Specific API Version	Query a specific API version of SWR.

Organization Permission Management

API	Description
Creating Organization Permissions	Grant permissions of an organization to certain users so that they can manage, edit, and read images from the organization.
Deleting Organization Permissions	Delete the organization operation permissions of certain users.
Updating Organization Permissions	Update the organization operation permission types for certain users.
Querying Organization Permissions	Query organization permissions.

Image Permission Management

API	Description
Creating Image Permissions	Grant permissions of an image to certain users so that they can manage, edit, and read the image.
Deleting Image Permissions	Delete the image operation permissions of certain users.
Updating Image Permissions	Update the image operation permission types for certain users.
Querying Image Permissions	Query image permissions.

Trigger Management

API	Description
Creating a Trigger	Create a trigger.
Obtaining the Trigger List of a Specific Image Repository	Obtain the trigger list of a specific image repository.

Deleting a Trigger	Delete a trigger.
Obtaining Information About a Trigger	Obtain information about a trigger.
Updating Trigger Settings	Update the status of a trigger.

Image Retention Policy Management

API	Description
Adding an Image Retention Policy	Add an image retention policy.
Obtaining the List of Image Retention Policies	Obtain the list of image retention policies.
Deleting an Image Retention Policy	Delete an image retention policy.
Obtaining an Image Retention Policy Record	Obtain the records of image retention policies.
Modifying an Image Retention Policy	Modify an image retention policy.
Obtaining Image Retention Records	Obtain image retention records.

Temporary Login Command

API	Description
Generating a Temporary Login Command	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.

Quota Management

API	Description
Obtaining Quota Information	Obtain quota information.

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 Regions and Endpoints. For example, the endpoint of IAM in region **Dublin** is **iam.myhuaweicloud.eu**.
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.

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 *Parameter name*=*Parameter value*. For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **Dublin** region, obtain the endpoint of IAM (**iam.myhuaweicloud.eu**) 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.myhuaweicloud.eu/v3/auth/tokens
```

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.
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.myhuaweicloud.eu/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 *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	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 **application/json** 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
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4

Parameter	Description	Mandatory	Example Value
X-Auth-Token	Specifies the user token. It is a response to the API for **obtaining a user token** (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQcCo...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.myhuaweicloud.eu/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 *xxxxxxxxxxxxxxxxxx* (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.myhuaweicloud.eu/v3/auth/tokens
Content-Type: application/json
```

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

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 authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.

Token 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.

IMS is a project-level service. When you call the API, set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",   // IAM user name
                    "password": "********",  // IAM user password
                    "domain": {
                        "name": "domainname"  // Name of the account to which the IAM user belongs
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"    // Project name
            }
        }
    }
}
```

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.myhuaweicloud.eu/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

NOTE

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

In AK/SK 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, which is 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 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 request. For more information, see **Status Codes**.

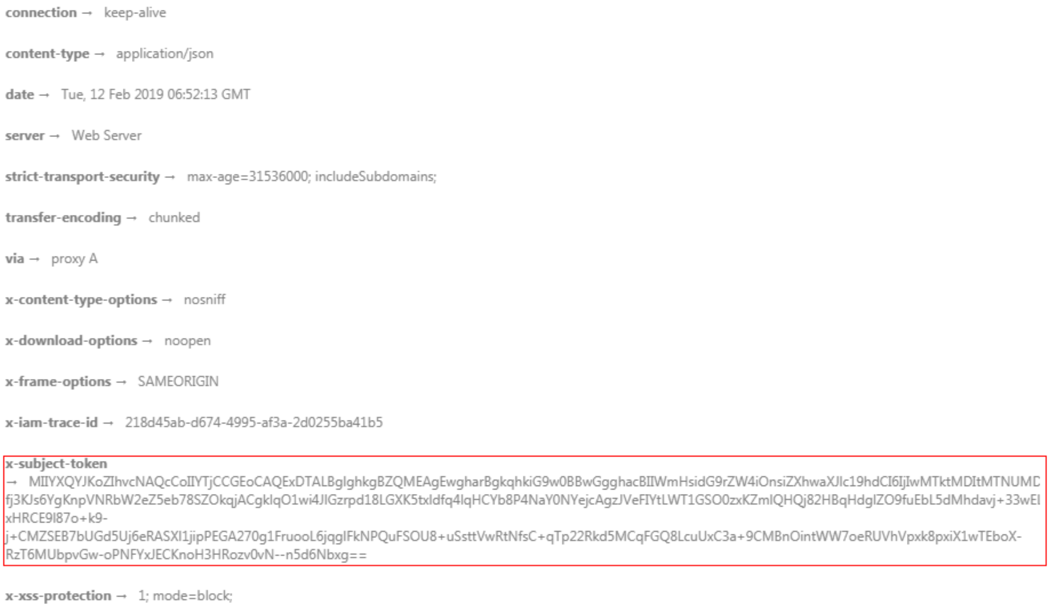
For example, if status code **201** is returned for calling the API used to **obtain a user token**, the request is successful.

Response Header

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

Figure 3-1 shows the response header fields for the API used to **obtain a user token**. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-1 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
→ MIIYXQYJKoZIhvcNAQcCoIIYTjCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjIwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5txldfq4lqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1GSO0zxKZmIQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-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": "az-01",
......
```

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 Organization Management

4.1.1 Creating an Organization

Function

Create an organization.

URI

POST /v2/manage/namespaces

Request Parameters

Table 4-1 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format). The value can be **application/json;charset=utf-8 application/json** The default value is **application/json**.
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API used to obtain a user token. The value of **X-Subject-Token** in the response header is the user token.

Table 4-2 Request body parameter

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.

Response Parameters

N/A

Example Request

```
POST https://{endpoint}/v2/manage/namespaces

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

Example Response

N/A

Status Code

Status Code	Description
201	Creation successful.
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.

4.1.2 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. 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 Parameters

Table 4-4 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format). The value can be **application/json;charset=utf-8 application/json** The default value is **application/json**.
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API used to obtain a user token. The value of **X-Subject-Token** in the response header is the user token.

Response

N/A

Status Code

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

4.1.3 Querying the List of Organizations

Function

Query the list of organizations.

URI

GET /v2/manage/namespaces

Table 4-5 Query parameters

Parameter	Mandatory	Type	Description
namespace	No	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
filter	No	String	Enter **namespace::{*namespace*}|mode::{*mode*}**. **{namespace}** indicates the organization name. If **{mode}** is not set, the list of authorized organizations is displayed. If **{mode}** is set to **visible**, the list of visible organizations is displayed. (Some organizations can be viewed by the repository, but cannot by the organization.)

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces?filter=namespace::group
```

Response

- Response parameters

Table 4-6 Response body parameter description

Parameter	Type	Description
namespaces	Array of **objects**	Organization list

Table 4-7 namespaces parameter description

Parameter	Type	Description
ID	Integer	Organization ID
name	String	Organization name
creator_name	String	IAM username
auth	Integer	User permission ● 7: Manage ● 3: Write ● 1: Read

- Example response

```
{
    "namespaces": [
        {
            "id": 1343008,
            "name": "group",
            "creator_name": "username",
            "auth": 7
        }
    ]
}
```

Status Code

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

4.1.4 Querying the Details of an Organization

Function

Query the details of an organization by its name.

URI

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

For details about parameters, see **Table 4-8**.

Table 4-8 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group
```

Response

- Response parameters

Table 4-9 Response body parameter description

Parameter	Type	Description
id	Integer	Organization ID
name	String	Organization name
creator_name	String	IAM username
auth	Integer	User permission ● 7: Manage ● 3: Write ● 1: Read

- Example response

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

Status Code

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

4.2 Image Repository Management

4.2.1 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 4-10**.

Table 4-10 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

- Request parameters

Table 4-11 FormData parameter description

Parameter	Mandatory	Type	Description
repository	Yes	String	Image repository name. 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 **app_server**, **linux**, **framework_app**, **database**, **lang**, **other**, **windows** or **arm**.
description	No	String	Brief description of the image repository.

Parameter	Mandatory	Type	Description
is_public	Yes	Boolean	Whether the repository is a public repository. When the value is **true**, it indicates the repository is public. When the value is **false**, it indicates the repository is private.

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/repos
```

  Body:

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

  Or

```
{
    "repository": "busybox",
    "category": "linux",
    "description": "this is a busybox repository",
    "is_public": true
}
```

NOTE

The form format will no longer be supported soon. You are advised to use the body in the JSON format to call the API.

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

Status Code	Description
201	Creation successful.
400	Request error. Error information is returned.
401	Authentication failed.
409	The repository already exists.
500	Internal error. Error information is returned.

4.2.2 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 4-12**.

Table 4-12 Parameter description

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

Request

- Request parameters

  N/A

- Example request

```
DELETE https://{Endpoint}/v2/manage/namespaces/group/repos/busybox
```

Response

N/A

Status Code

Status Code	Description
204	The brief information about the image repository is successfully deleted.
400	Request error. Error information is returned.
401	Authentication failed.
404	The repository does not exist.
406	The repository cannot be deleted because it contains images.
500	Internal error. Error information is returned.

4.2.3 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 4-13**.

Table 4-13 Parameter description

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

Request

- Request parameters

Table 4-14 Request body parameter description

Parameter	Mandatory	Type	Description
category	No	String	Repository type. The value can be **app_server**, **linux**, **framework_app**, **database**, **lang**, **other**, **windows** or **arm**.
description	No	String	Repository description.
is_public	Yes	Boolean	Whether the repository is a public repository. The value can be either **true** or **false**.

- Example request

```
PATCH https://{Endpoint}/v2/manage/namespaces/group/repos/busybox
```

Body:

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

Or

```
{
   "category": "linux",
   "description": "this is a busybox repository",
   "is_public": false
}
```

NOTE

The form format will no longer be supported soon. You are advised to use the body in the JSON format to call the API.

Response

- Response parameters

  N/A
- Example response

```
{}
```

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 failed.
404	The repository does not exist.
500	Internal error. Error information is returned.

4.2.4 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 4-15**.

Table 4-15 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

Parameter	Mandatory	Type	Description
repository	Yes	String	Image repository name

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox
```

Response

- Response parameters

Table 4-16 Response body parameter description

Parameter	Type	Description
id	Integer	Image repository ID.
ns_id	Integer	Organization ID.
name	String	Image repository name.
category	String	Image repository type. The value can be **app_server**, **linux**, **framework_app**, **database**, **lang**, **other**, **windows**, or **arm**.
description	String	Brief description of the image repository.
creator_id	String	Image repository creator ID.
creator_name	String	Image repository creator.
size	Integer	Image repository size.
is_public	Boolean	Whether the image repository is a public repository. The value can be **true** or **false**.
num_images	Integer	Number of images in an image repository.
num_download	Integer	Download times of an image repository.

Parameter	Type	Description
url	String	URL of the image repository logo image. This field has been discarded and is left empty by default.
path	String	External image pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}.
internal_path	String	Internal image pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}.
created	String	Time when an image repository is created. It is the UTC standard time.
updated	String	Time when an image repository is updated. It is the UTC standard time.
domain_id	String	Account ID.
priority	Integer	Image sorting priority.

- Example response

```
{
    "id": 865151,
    "ns_id": 1334795,
    "name": "busybox",
    "category": "other",
    "description": "containerops",
    "creator_id": "050b12577f00269a1fcfc01f65239697",
    "creator_name": "admin",
    "size": 2099575,
    "is_public": false,
    "num_images": 1,
    "num_download": 0,
    "url": "",
    "path": "{Repository address}/group/busybox",
    "internal_path": "{Repository address}/group/busybox",
    "created": "2021-06-02T09:59:25.51307Z",
    "updated": "2021-06-02T17:59:25.538056Z",
    "domain_id": "da44776c316c4a99b3683bb174f8821b",
    "priority": 0
}
```

Status Code

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

4.2.5 Querying the Image Repository List

Function

Query the image repository list.

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 4-17**.

⚠ CAUTION

At least one filter parameter must be transferred.

Table 4-17 Parameter description

Parameter	Mandatory	Type	Description
center	No	String	The value can only be self, indicating that the image is a self-owned image.
namespace	No	String	Organization name.
name	No	String	Image repository name.
category	No	String	Image repository type. The value can be **app_server**, **linux**, **framework_app**, **database**, **lang**, **other**, **windows**, or **arm**.

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

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/repos?filter=center::self|namespace::group|name::busybox
```

Response

- Response parameters

Table 4-18 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 4-19 Response body parameter description

Parameter	Type	Description
[Array element]	Array of **objects**	Image repository list.

Table 4-20 *[Array element]* parameters description

Parameter	Type	Description
name	String	Image repository name.
category	String	Image repository type. The value can be **app_server**, **linux**, **framework_app**, **database**, **lang**, **other**, **windows**, or **arm**.
description	String	Image repository description.
size	Integer	Total size of the images in an image repository.
is_public	Boolean	Whether the image is a public image. The value can be **true** or **false**.
num_images	Integer	Number of image tags in an image repository.
num_download	Integer	Download times.
created_at	String	Time when an image repository is created. It is the UTC standard time.
updated_at	String	Time when an image repository is updated. It is the UTC standard time.
logo	String	(Reserved field) Image repository logo address.
url	String	(Reserved field) URL of the image repository logo.

Parameter	Type	Description
path	String	External Docker pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}.
internal_path	String	Internal Docker pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}.
domain_name	String	Account name.
namespace	String	Organization name.
tags	Array of strings	Image tag list.
status	Boolean	(Reserved field) Status.
total_range	Integer	Total number of records.

- Example response

```
[
   {
      "name": "busybox",
      "category": "linux",
      "description": "this is a busybox repository",
      "size": 200977474,
      "is_public": false,
      "num_images": 2,
      "num_download": 0,
      "created_at": "2021-06-10T03:24:44.784402Z",
      "updated_at": "2021-06-10T03:40:59.199352Z",
      "logo": "",
      "url": "",
      "path": "{Repository address}/group/busybox",
      "internal_path": "{Repository address}/group/busybox",
      "domain_name": "root",
      "namespace": "group",
      "tags": [
         "latest",
         "log"
      ],
      "status": false,
      "total_range": 1
   }
]
```

Status Code

Status Code	Description
200	Request successful.
400	Request error. Error information is returned.

Status Code	Description
401	Authentication failed.
500	Internal error. Error information is returned.

4.2.6 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 4-21**.

Table 4-21 Parameter description

Parameter	Mandatory	Type	Description
center	Yes	String	● **self**: images shared by you. ● **thirdparty**: images shared with you by others.
name	No	String	Image repository name.
limit	No	String	Number of returned records. Ensure that the **offset** and **limit** parameters are used together.
offset	No	String	Start index. Ensure that the **offset** and **limit** parameters are used together.

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/shared-repositories?filter=center::self|name::busybox
```

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
[Array element]	Array of **objects**	Viewing the list of shared images.

Table 4-24 *[Array element]* parameters description

Parameter	Type	Description
name	String	Image repository name.
category	String	Image repository type. The value can be **app_server**, **linux**, **framework_app**, **database**, **lang**, **other**, **windows**, or **arm**.
description	String	Image repository description.
size	Integer	Total size of the images in an image repository.
is_public	Boolean	Whether the image is a public image. The value can be **true** or **false**.
num_images	Integer	Number of image tags in an image repository.
num_download	Integer	Download times.
created_at	String	Time when an image repository is created. It is the UTC standard time.
updated_at	String	Time when an image repository is updated. It is the UTC standard time.
logo	String	(Reserved field) Image repository logo address.

Parameter	Type	Description
url	String	(Reserved field) URL of the image repository logo.
path	String	External Docker pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}.
internal_path	String	Internal Docker pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}.
domain_name	String	Account name.
namespace	String	Organization name.
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 **false**, indicating that no images are shared.
total_range	Integer	Total number of records.

- Example response

```
[
  {
     "name": "busybox",
     "category": "other",
     "description": "",
     "size": 115645792,
     "is_public": false,
     "num_images": 1,
     "num_download": 0,
     "created_at": "2021-06-02T09:23:01.751202Z",
     "updated_at": "2021-06-10T07:24:35.204517Z",
     "logo": "",
     "url": "",
     "path": "{Repository address}/group/busybox",
     "internal_path": "{Repository address}/group/busybox",
     "domain_name": "domain",
     "namespace": "namespace",
     "tags": [
        "v1",
        "v2"
     ],
     "status": true,
     "total_range": 1
  }
]
```

Status Code

Status Code	Description
200	Request successful.
400	Request error. Error information is returned.
401	Authentication failed.
500	Internal error. Error information is returned.

4.3 Image Tag Management

4.3.1 Querying the List of Image Repository Tags

Function

Query all image tags of an image repository.

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 **updated_at** (sorting by update time).
order_type	No	String	Sorting type. You can set this parameter to **desc** (descending sort) and **asc** (ascending sort).

Parameter	Mandatory	Type	Description
tag	No	String	Image tag name.

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/tags
```

Response

- Response parameters

Table 4-26 Response body parameter description

Parameter	Type	Description.
[Array element]	Array of **objects**	Image repository tag list.

Table 4-27 *[Array element]* parameters description

Parameter	Type	Description
id	Integer	ID
repo_id	Integer	Repository ID.
Tag	String	Image tag name.
image_id	String	Image ID.
manifest	String	Image manifest.
digest	String	SHA value of an image.
schema	Integer	Docker protocol version. The version can be **1** or **2**.
path	String	External image pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}:{*Tag name*}.

Parameter	Type	Description
internal_path	String	Internal image pull address. The format is {*Repository address*}/{*Namespace name*}/{*Repository name*}:{*Tag name*}.
size	Integer	Image size. Value range: 0 to 9223372036854775807 Unit: byte
is_trusted	Boolean	By default, the value is **false**.
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.
domain_id	String	Account ID.
tag_type	Integer	0: manifest. 1: manifest list.

- Example response

```
[
   {
       "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\":
```

```
\"sha256:77ddbf3a9fe11e81761a0f9df43a28e3e6f29bbb53c0c8cf71cd7efa69729aed\"}]}",
        "digest": "sha256:57b605845a6367c34bfb6ea6477f16852f59aa1861a2b51d10ab77ae0a1dc9c3",
        "schema": 2,
        "path": "{Repository address}/group/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,
        "domain_id": "0503dda878000fed0f75c0096d70a960",
        "tag_type": 0
    }
]
```

Status Code

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

4.3.2 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	Image tag name

Request

- Request parameters

  N/A

- Example request

  ```
  DELETE https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/tags/v1
  ```

Response

N/A

Status Code

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

4.4 Shared Account Management

4.4.1 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 4-29**.

Table 4-29 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 4-30 Request body parameter description

Parameter	Mandatory	Type	Description
access_domain	Yes	String	Name of the account used for image sharing.
permit	Yes	String	Currently, only the read permission is supported.
deadline	Yes	String	Valid until (UTC). If the sharing is permanent, the value is **forever**. Otherwise, the sharing is valid until 00:00:00 of the next day.
description	No	String	Description.

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/repositories/busybox/access-domains
```

Body:

```
{
    "access_domain": "domain_name",
    "permit": "read",
    "deadline": "2021-10-01T16:00:00.000Z",
    "description": "description"
}
```

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

Status Code	Description
201	Creation successful.
400	Request error.

Status Code	Description
401	Authentication failed.
409	The repository has been shared.
500	Internal error.

4.4.2 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 4-31**.

Table 4-31 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.
repository	Yes	String	Image repository name.
access_domain	Yes	String	Name of the account used for image sharing.

Request

- Request parameters

  N/A

- Example request

```
DELETE https://{Endpoint}/v2/manage/namespaces/group/repositories/busybox/access-domains/domain_name
```

Response

N/A

Status Code

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

4.4.3 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 4-32**.

Table 4-32 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
access_domain	Yes	String	Name of the account used for image sharing

Request

- Request parameters

Table 4-33 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 **forever**. Otherwise, the sharing is valid until 00:00:00 of the next day.
description	No	String	Description. This parameter is left blank by default.

- Example request

  ```
  PATCH https://{Endpoint}/v2/manage/namespaces/group/repositories/busybox/access-domains/domain_name
  ```

  Body:

  ```
  {
     "permit": "read",
     "deadline": "forever",
     "description": "description"
  }
  ```

Response

N/A

Status Code

Status Code	Description
201	Modified successfully.
400	Request error.
401	Authentication failed.
500	Internal error.

4.4.4 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 4-34**.

Table 4-34 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name
repository	Yes	String	Image repository name
access_domain	Yes	String	Name of the account used for image sharing.

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repositories/busybox/access-domains/
domain_name
```

Response

- Response parameters

Table 4-35 Response body parameter description

Parameter	Type	Description
exist	Boolean	**true**: Exist. **false**: Not exist.
namespace	String	Organization name.
repository	String	Image repository name.
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.
created	String	Time when an image is created. It is the UTC standard time.
updated	String	Time when an image is updated. It is the UTC standard time.

Parameter	Type	Description
status	Boolean	Status. true: valid. false: expired.

- Example response

```
{
    "exist": true,
    "namespace": "group",
    "repository": "busybox",
    "access_domain": "xxx",
    "permit": "read",
    "deadline": "2021-10-01T16:00:00Z",
    "description": "description",
    "creator_id": "0504186e6a8010e01f3ec009a7279baa",
    "creator_name": "xxx",
    "created": "2021-06-10T08:14:42.56632Z",
    "updated": "2021-06-10T08:14:42.566325Z",
    "status": true
}
```

Status Code

Status Code	Description
200	Query succeeded.
400	Request error.
401	Authentication failed.
500	Internal error.

4.4.5 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 4-36**.

Table 4-36 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name

Parameter	Mandatory	Type	Description
repository	Yes	String	Image repository name

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repositories/busybox/access-domains
```

Response

- Response parameters

Table 4-37 Response body parameter description

Parameter	Type	Description
[Array element]	Array of **objects**	Shared accounts.

Table 4-38 *[Array element]* parameter description

Parameter	Type	Description
namespace	String	Organization name.
repository	String	Image repository name.
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.
created	String	Time when an image is created. It is the UTC standard time.
updated	String	Time when an image is updated. It is the UTC standard time.

Parameter	Type	Description
status	Boolean	Status. true: valid. false: expired.

- Example response

```
[
   {
      "namespace": "group",
      "repository": "busybox",
      "access_domain": "swr",
      "permit": "read",
      "deadline": "2021-10-01T16:00:00Z",
      "description": "description",
      "creator_id": "fb3f175c1fd146ab8cdae3272be6107b",
      "creator_name": "group",
      "created": "2021-06-10T08:14:42.56632Z",
      "updated": "2021-06-10T08:14:42.566325Z",
      "status": true
   }
]
```

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.
500	Internal error. Error information is returned.

4.5 API Versions

4.5.1 Querying All API Versions

Function

Query all available API versions of SWR.

URI

GET /

Request

N/A

Response

- Response parameters

Table 4-39 Response body parameter description

Parameter	Type	Description
versions	Array of **objects**	A list of objects related to the version.

Table 4-40 versions parameter description

Parameter	Type	Description
id	String	Version ID (version number), for example, v2.
links	**Object**	API URL.
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: ● **CURRENT**: The version is the primary version. ● **SUPPORTED**: The version is an old version, but it is still supported. ● **DEPRECATED**: The version is a deprecated version, which may be deleted later.
updated	String	Version release time, which must be the UTC time. For example, the release time of v2 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 4-41 links parameter description

Parameter	Type	Description
href	String	Link

Parameter	Type	Description
rel	String	Description

- Example response

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

Status Code

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

4.5.2 Querying a Specific API Version

Function

Query a specific API version of SWR.

URI

GET /{*api_version*}

For details about parameters, see **Table 4-42**.

Table 4-42 Parameter description

Parameter	Mandatory	Type	Description
api_version	Yes	String	API version

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2
```

Response

- Response parameters

Table 4-43 Response body parameter description

Parameter	Type	Description
version	**Object**	A list of objects related to the version.

Table 4-44 version parameter description

Parameter	Type	Description
id	String	Version ID (version number), for example, v2.
links	**Object**	API URL.
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: ● **CURRENT**: The version is the primary version. ● **SUPPORTED**: The version is an old version, but it is still supported. ● **DEPRECATED**: The version is a deprecated version, which may be deleted later.
updated	String	Version release time, which must be the UTC time. For example, the release time of v2 is **2018-06-28T12:20:21Z**.
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 4-45 links parameter description

Parameter	Type	Description
href	String	Link
rel	String	Description

- Example response

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

Status Code

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

4.6 Organization Permission Management

4.6.1 Creating Organization Permissions

Function

Grant permissions of an organization to certain users so that they can manage, edit, and read images from the organization.

URI

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

Table 4-46 describes the parameters.

Table 4-46 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

Request

- Request parameters

Table 4-47 Request body description

Parameter	Mandatory	Type	Description
[Array element]	Yes	Array of **objects**	Information required for creating organization permissions.

Table 4-48 [Array element]

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.
user_name	Yes	String	Username.
auth	Yes	Integer	User permission ● 7: Manage ● 3: Write ● 1: Read

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/access
```

Body:

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

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

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

4.6.2 Deleting Organization Permissions

Function

Delete the organization operation permissions of certain users.

URI

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

Table 4-49 describes the parameters.

Table 4-49 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

Request

- Request parameters

Table 4-50 Request body parameter description

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

- Example request

```
DELETE https://{Endpoint}/v2/manage/namespaces/group/access
```

Body:

```
["fb3f175c1fd146ab8cdae3272be6107b"]
```

Response

N/A

Status Code

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

4.6.3 Updating Organization Permissions

Function

Update the organization operation permissions for certain users.

URI

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

For details about parameters, see **Table 4-51**.

Table 4-51 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

Request

- Request parameters

Table 4-52 Request body parameter description

Parameter	Mandatory	Type	Description
[Array element]	Yes	Array of **objects**	Information required for updating organization permissions

Table 4-53 [Array element]

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.
user_name	Yes	String	Username.
auth	Yes	Integer	User permission ● 7: Manage ● 3: Write ● 1: Read

- Example request

```
PATCH https://{Endpoint}/v2/manage/namespaces/group/access
```

  Body:

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

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

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

4.6.4 Querying Organization Permissions

Function

Query organization permissions.

URI

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

For details about parameters, see **Table 4-54**.

Table 4-54 Parameter description

Parameter	Mandatory	Type	Description
namespace	Yes	String	Organization name.

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/access
```

Response

- Response parameters

Table 4-55 Response body parameters

Parameter	Type	Description
id	Integer	Permission ID.
name	String	Organization name.
creator_name	String	Organization creator.
self_auth	**Object**	Permissions of the current user.
others_auths	Array of **objects**	Permissions of other users.

Table 4-56 self_auth parameter description

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

Table 4-57 others_auths parameter description

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

- Example response

```
{
    "id": 1334795,
    "name": "group",
    "creator_name": "user",
    "self_auth": {
        "user_id": "050b12577f00269a1fcfc01f65239697",
        "user_name": "user",
        "auth": 7
    },
    "others_auths": [
        {
            "user_id": "06d89c3b6d800f2d1f28c01f0d882285",
            "user_name": "user_01",
            "auth": 1
        },
        {
            "user_id": "050b12577f00269a1fcfc01f65239697",
            "user_name": "user",
            "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.

4.7 Image Permission Management

4.7.1 Creating Image Permissions

Function

Grant permissions of an image to certain users so that they can manage, edit, and read the image.

URI

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

For details about parameters, see **Table 4-58**.

Table 4-58 Parameter description

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

Request

- Request parameters

Table 4-59 Request body parameter description

Parameter	Mandatory	Type	Description.
[Array element]	Yes	Array of **objects**	User list

Table 4-60 *[Array element]* parameters decription

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.
user_name	Yes	String	Username.
auth	Yes	Integer	User permission that is configured. ● 7: Manage ● 3: Write ● 1: Read

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/access
```

Body:

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

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

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

4.7.2 Deleting Image Permissions

Function

Delete the image operation permissions of certain users.

URI

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

For details about parameters, see **Table 4-61**.

Table 4-61 Parameter description

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

Request

- Request parameters

Table 4-62 Request body parameter description

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

- Example request

```
DELETE https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/access
```

Body:

```
["fb3f175c1fd146ab8cdae3272be6107b"]
```

Response

N/A

Status Code

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

4.7.3 Updating Image Permissions

Function

Update the image operation permissions for certain users.

URI

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

For details about parameters, see **Table 4-63**.

Table 4-63 Parameter description

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

Request

- Request parameters

Table 4-64 Request body parameter description

Parameter	Mandatory	Type	Description
[Array element]	Yes	Array of **objects**	Information about the image permission to be updated.

Table 4-65 *[Array element]* parameter description

Parameter	Mandatory	Type	Description
user_id	Yes	String	User ID.
user_name	Yes	String	Username.
auth	Yes	Integer	User permission that is configured. ● 7: Manage ● 3: Write ● 1: Read

- Example request

```
PATCH https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/access
```

Body:

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

Response

- Response parameters

  N/A

- Example response

```
{}
```

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.

4.7.4 Querying Image Permissions

Function

Query image permissions.

URI

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

For details about parameters, see **Table 4-66**.

Table 4-66 Parameter description

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

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/access
```

Response

- Response parameters

Table 4-67 Response body parameter description

Parameter	Type	Description
id	Integer	Permission ID.
name	String	Image name.
self_auth	**Object**	Permissions of the current user.
others_auths	Array of **objects**	Permissions of other users.

Table 4-68 self_auth parameter description

Parameter	Type	Description
user_id	String	User ID
user_name	String	Username
auth	Integer	User permission ● 7: Manage ● 3: Write ● 1: Read

Table 4-69 others_auths parameter description

Parameter	Type	Description
user_id	String	User ID
user_name	String	Username
auth	Integer	User permission ● 7: Manage ● 3: Write ● 1: Read

- Example response

```
{
    "id": 1422,
    "name": "busybox",
    "self_auth": {
        "user_id": "3059e6b5562241fda3fa441cca6f228b",
        "user_name": "admin",
        "auth": 7
```

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

Status Code

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

4.8 Automatic Image Synchronization Task Management

4.8.1 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 4-70 describes the parameters.

Table 4-70 Parameter description

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

Parameter	Mandatory	Type	Description
limit	Yes	String	Number of returned records. Ensure that the **limit** and **offset** parameters are used together.
offset	Yes	String	Start index. Ensure that the **limit** and **offset** parameters are used together.
order	No	String	Sorting by creation time. The value can be **desc** (descending order) or **asc** (ascending order).

Request

- Request parameters

  N/A
- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/sync_job?filter=limit::10|offset::0
```

Response

- Response parameters

Table 4-71 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 4-72 Response body parameter description

Parameter	Type	Description.
[Array element]	Array of **objects**	Obtaining the list of image synchronization tasks.

Table 4-73 *[Array element]* parameters description

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

- Example response

```
[
  {
    "createdAt": "2020-03-12T08:11:36.518144Z",
    "domainID": "0657**************************f60",
    "domainName": "h**********4",
    "id": 172928,
    "namespace": "group",
    "override": false,
    "remoteNamespace": "remtens",
    "remoteRegionId": "region-1",
    "repoName": "busybox",
    "status": "success",
    "syncOperatorId": "065***********************10",
    "syncOperatorName": "h**********4",
    "tag": "8",
    "updatedAt": "2020-03-12T08:11:41.505499Z"
  }
]
```

Status Code

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

4.8.2 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 4-74 describes the parameters.

Table 4-74 Parameter description

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

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/sync_repo
```

Response

- Response parameters

Table 4-75 Response body parameter description

Parameter	Type	Description.
[Array element]	Array of **objects**	Obtaining the list of automatic image synchronization tasks.

Table 4-76 *[Array element]* parameters description

Parameter	Type	Description
createdAt	String	Creation time.
domainID	String	Account ID.
domainName	String	Account name.
id	Integer	Task ID.
namespace	String	Organization name.
override	Boolean	Whether to overwrite.
remoteNamespace	String	Target organization.
remoteRegionId	String	Target region.
repoName	String	Image repository name.
syncAuto	Boolean	Whether automatic synchronization is enabled.
updatedAt	String	Update time.

- Example response

```
[
  {
    "createdAt": "2020-04-17T02:11:03.53183Z",
    "domainID": "8b1e81************************633fb",
    "domainName": "p*********59",
    "id": 1364,
    "namespace": "group",
    "override": true,
    "remoteNamespace": "sdsad",
    "remoteRegionId": "region-1",
    "repoName": "busybox",
    "syncAuto": true,
    "updatedAt": "2020-04-17T02:11:03.531831Z"
  }
]
```

Status Code

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

4.8.3 Creating an Automatic Image Synchronization Task

Function

Create an automatic image synchronization task.

URI

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

Table 4-77 describes the parameters.

Table 4-77 Parameter description

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

Request

- Request parameters

Table 4-78 Request body parameter description

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

Parameter	Mandatory	Type	Description
syncAuto	No	Boolean	(Reserved field) Automatic synchronization. You are recommended to retain the default value **false**.
override	No	Boolean	Whether to overwrite. The default value is **false**, which indicates not to overwrite any nonidentical image that has the same name in the target organization.

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/sync_repo
```

Body:

```
{
    "remoteRegionId": "region-2",
    "remoteNamespace": "sdsad",
    "syncAuto": true,
    "override": true
}
```

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

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

4.8.4 Deleting an Automatic Image Synchronization Task

Function

Delete an automatic image synchronization task.

URI

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

Table 4-79 describes the parameters.

Table 4-79 Parameter description

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

Request

- Request parameters

Table 4-80 Request body parameter description

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

- Example request

```
DELETE https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/sync_repo
```

Body:

```
{
    "remoteRegionId": "region-2",
    "remoteNamespace": "sdsad"
}
```

Response

- Response parameters

N/A

- Example response

```
{}
```

Status Code

Status Code	Description
200	Deleted successfully.

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.

4.8.5 Manually Synchronizing an Image

Function

Manually synchronize an image.

URI

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

Table 4-81 describes the parameters.

Table 4-81 Parameter description

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

Request

- Request parameters

Table 4-82 Request body parameter description

Parameter	Mandatory	Type	Description
imageTag	Yes	Array of String	Image tag list.
override	No	Boolean	Whether to overwrite. The default value is **false**, which indicates not to overwrite any nonidentical image that has the same name in the target organization.

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

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/sync_images
```

Body:

```
{
    "imageTag": [
        "v1"
    ],
    "override": false,
    "remoteNamespace": "sdsad",
    "remoteRegionId": "region-2"
}
```

Response

- Response parameters

Table 4-83 Response body parameter description

Parameter	Type	Description
[Array element]	Array of strings	Image tags that fail to be synchronized.

- Example response

```
[]
```

Status Code

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

4.9 Trigger Management

4.9.1 Creating a Trigger

Function

Create a trigger.

URI

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

Table 4-84 describes the parameters.

Table 4-84 Parameter description

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

Request

- Request parameters

Table 4-85 Request body parameter description

Parameter	Mandatory	Type	Description
action	Yes	String	Action to trigger. The value is **update**.
app_type	Yes	String	Application type. The value can be **deployments** or **statefulsets**.
application	Yes	String	Application name.
cluster_id	No	String	Cluster ID. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_name	No	String	Cluster name. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_ns	Yes	String	Namespace where the application is located.

Parameter	Mandatory	Type	Description
condition	Yes	String	Trigger condition. - When **trigger_type** is set to **all**, set this parameter to **.***. - When **trigger_type** is set to **tag**, set this parameter to specific image tags separated by semicolons (;). - When **trigger_type** is set to **regular**, 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 **cce** or **cci**. The default value is **cce**.
trigger_type	Yes	String	Trigger condition. The value can be **all** (all tags), **tag** (specific tag), or **regular** (tags matching regular expression).

- Example request

  ```
  POST https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/triggers
  ```

  Body:

  ```
  {
     "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

- Response parameters

  N/A

- Example response

  ```
  {}
  ```

Status Code

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

4.9.2 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 4-86 describes the parameters.

Table 4-86 Parameter description

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

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/triggers
```

Response

- Response parameters

Table 4-87 Response body parameter description

Parameter	Type	Description
[Array element]	Array of **objects**	Trigger list

Table 4-88 [Array element]

Parameter	Type	Description
action	String	Action to trigger. The value is **update**.
app_type	String	Application type. The value can be **deployments** or **statefulsets**.
application	String	Application name.
cluster_id	String	Cluster ID. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_name	String	Cluster name. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_ns	String	Namespace where the application is located.
condition	String	Trigger condition. ● When **trigger_type** is set to **all**, set this parameter to **.***. ● When **trigger_type** is set to **tag**, set this parameter to specific image tags separated by semicolons (;). ● When **trigger_type** is set to **regular**, 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 **objects**	(Reserved field) This parameter is not in use.
trigger_mode	String	Trigger type. The value can be **cce** or **cci**.

Parameter	Type	Description
trigger_type	String	Trigger condition. The value can be **all** (all tags), **tag** (specific tag), or **regular** (tags matching regular expression).

Table 4-89 trigger_history parameter description

Parameter	Type	Description
action	String	Action to trigger. The value is **update**.
app_type	String	Application type. The value can be **deployments** or **statefulsets**.
application	String	Application name.
cluster_id	String	Cluster ID. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_name	String	Cluster name. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_ns	String	Namespace where the application is located.
condition	String	Trigger condition. ● When **trigger_type** is set to **all**, set this parameter to **.***. ● When **trigger_type** is set to **tag**, set this parameter to specific image tags separated by semicolons (;). ● When **trigger_type** is set to **regular**, 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 **success** or **failed**.
tag	String	Triggered tag.

- Example response

```
[
  {
    "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

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

4.9.3 Deleting a Trigger

Function

Delete a trigger.

URI

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

Table 4-90 describes the parameters.

Table 4-90 Parameter description

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

Request

- Request parameters

  N/A

- Example request

  ```
  DELETE https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/triggers/test1
  ```

Response

NA

Status Code

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

4.9.4 Obtaining Information About a Trigger

Function

Obtain information about a trigger.

URI

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

Table 4-91 describes the parameters.

Table 4-91 Parameter description

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

Request

- Request parameters

  N/A

- Example request

  ```
  GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/triggers/test
  ```

Response

- Response parameters

Table 4-92 Response body parameter description

Parameter	Type	Description
action	String	Action to trigger. The value can be **update**.
app_type	String	Application type. The value can be **deployments** or **statefulsets**.
application	String	Application name.
cluster_id	String	Cluster ID. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_name	String	Cluster name. If **trigger_mode** is set to **cci**, leave this parameter blank.
cluster_ns	String	Namespace where the application is located.
condition	String	Trigger condition. ● When **trigger_type** is set to **all**, set this parameter to **.***. ● When **trigger_type** is set to **tag**, set this parameter to specific image tags separated by semicolons (;). ● When **trigger_type** is set to **regular**, 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 **objects**	Trigger history.
trigger_mode	String	Trigger type. The value can be **cce** or **cci**.

Parameter	Type	Description
trigger_type	String	Trigger condition. The value can be **all**, **tag**, or **regular**.

Table 4-93 Parameter description

Parameter	Type	Description
action	String	Action to trigger. The value can be **update**.
app_type	String	Application type. The value can be **deployments** or **statefulsets**.
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 **trigger_type** is set to **all**, set this parameter to **.***. ● When **trigger_type** is set to **tag**, set this parameter to specific image tags separated by semicolons (;). ● When **trigger_type** is set to **regular**, 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 **success** or **failed**.
tag	String	Image tag that triggers the update.

- Example response

```
{
    "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": "xxxxxxxxxxxxxx",
        "result": "failed",
        "tag": "test"
      }
    ],
    "trigger_mode": "cci",
    "trigger_type": "all"
}
```

Status Code

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

4.9.5 Updating Trigger Settings

Function

Update the status of a trigger.

URI

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

Table 4-94 describes the parameters.

Table 4-94 Parameter description

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

Request

- Request parameters

Table 4-95 Request body parameter description

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

- Example request

```
PATCH https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/triggers/test
```

Body:

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

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

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

4.10 Image Retention Policy Management

4.10.1 Adding an Image Retention Policy

Function

Add an image retention policy.

URI

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

Table 4-96 describes the parameters.

Table 4-96 Parameter description

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

Request

- Request parameters

Table 4-97 Request body parameter description

Parameter	Mandatory	Type	Description
algorithm	Yes	String	Retention policy matching rule. The value is **or**.
rules	Yes	Array of **objects**	Image retention policy.

Table 4-98 rules parameter description

Parameter	Mandatory	Type	Description
template	Yes	String	Retention policy type. The value can be **date_rule** (number of days) or **tag_rule** (number of tags).

Parameter	Mandatory	Type	Description
params	Yes	Object	- If **template** is set to **date_rule**, set **params** to **{"days": "*xxx*"}**. - If **template** is set to **tag_rule**, set **params** to **{"num": "*xxx*"}**.
tag_selectors	Yes	Array of **objects**	Image tags that are not counted in the retention policy.

Table 4-99 tag_selectors parameter description

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

- Example request

```
POST https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/retentions
```

Body:

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

Response

- Response parameters

Table 4-100 Response body parameter description

Parameter	Type	Description
id	Integer	Rule ID.

- Example response

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

Status Code

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

4.10.2 Obtaining the List of Image Retention Policy

Function

Obtain the list of image retention policy.

URI

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

Table 4-101 describes the parameters.

Table 4-101 Parameter description

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

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/retentions
```

Response

- Response parameters

Table 4-102 Response body parameter description

Parameter	**Type**	**Description.**
[Array element]	Array of **objects**	Obtaining the list of image retention policy.

Table 4-103 *[Array element]* parameters description

Parameter	**Type**	**Description**
algorithm	String	Retention policy matching rule. The value is **or**.
id	Integer	Rule ID.
rules	Array of **objects**	Image retention policy.
scope	String	Reserved field.

Table 4-104 rules parameter description

Parameter	**Type**	**Description**
template	String	Retention policy type. The value can be **date_rule** (number of days) or **tag_rule** (number of tags).
params	Object	● If **template** is set to **date_rule**, set **params** to **{"days": "*xxx*"}**. ● If **template** is set to **tag_rule**, set **params** to **{"num": "*xxx*"}**.
tag_selectors	Array of **objects**	Image tags that are not counted in the retention policy.

Table 4-105 tag_selectors parameter description

Parameter	**Type**	**Description**
kind	String	Matching rule. The value can be **label** or **regexp**.

Parameter	Type	Description
pattern	String	- If **kind** is set to **label**, set this parameter to specific image tags. - If **kind** is set to **regexp**, set this parameter to a regular expression.

- Example response

```
[
    {
        "algorithm": "or",
        "id": 1093,
        "rules": [
            {
                "template": "date_rule",
                "params": {
                    "days": "30"
                },
                "tag_selectors": [
                    {
                        "kind": "label",
                        "pattern": "v5"
                    },
                    {
                        "kind": "label",
                        "pattern": "1.0.1"
                    },
                    {
                        "kind": "regexp",
                        "pattern": "^123$"
                    }
                ]
            }
        ],
        "scope": ""
    }
]
```

Status Code

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

4.10.3 Deleting an Image Retention Policy

Function

Delete an image retention policy.

URI

DELETE /v2/manage/namespaces/{namespace}/repos/{repository}/retentions/{retention_id}

Table 4-106 describes the parameters.

Table 4-106 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

  N/A

- Example request

```
DELETE https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/retentions/1093
```

Response

- Response parameters

  N/A

- Example response

```
{}
```

Status Code

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

4.10.4 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 4-107 describes the parameters.

Table 4-107 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

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/retentions/1093
```

Response

- Response parameters

Table 4-108 Response body parameter description

Parameter	Type	Description
algorithm	String	Retention policy matching rule. The value is **or**.
id	Integer	Rule ID.
rules	Array of **objects**	Image retention policy.
scope	String	Reserved field.

Table 4-109 rules parameter description

Parameter	Type	Description
template	String	Retention policy type. The value can be **date_rule** (number of days) or **tag_rule** (number of tags).

Parameter	Type	Description
params	Object	- If **template** is set to **date_rule**, set **params** to **{"days": "*xxx*"}**. - If **template** is set to **tag_rule**, set **params** to **{"num": "*xxx*"}**.
tag_selectors	Array of **objects**	Image tags that are not counted in the retention policy.

Table 4-110 tag_selectors parameter description

Parameter	Type	Description
kind	String	Matching rule. The value can be **label** or **regexp**.
pattern	String	- If **kind** is set to **label**, set this parameter to specific image tags. - If **kind** is set to **regexp**, set this parameter to a regular expression.

- Example response

```
{
    "algorithm": "or",
    "id": 1093,
    "rules": [
        {
            "template": "date_rule",
            "params": {
                "days": "30"
            },
            "tag_selectors": [
                {
                    "kind": "label",
                    "pattern": "v5"
                },
                {
                    "kind": "label",
                    "pattern": "1.0.1"
                },
                {
                    "kind": "regexp",
                    "pattern": "^123$"
                }
            ]
        }
    ],
    "scope": ""
}
```

Status Code

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

4.10.5 Modifying an Image Retention Policy

Function

Modify an image retention policy.

URI

PATCH /v2/manage/namespaces/{namespace}/repos/{repository}/retentions/{retention_id}

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
retention_id	Yes	String	Image retention policy ID

Request

- Request parameters

Table 4-112 Request body parameter description

Parameter	Mandatory	Type	Description
algorithm	Yes	String	Retention policy matching rule. The value is **or**.

Parameter	Mandatory	Type	Description
rules	Yes	Array of **objects**	Image retention policy.

Table 4-113 rules parameter description

Parameter	Mandatory	Type	Description
template	Yes	String	Retention policy type. The value can be set **date_rule** and **tag_rule**.
params	Yes	Object	● If **template** is set to **date_rule**, set **params** to **{"days": "*xxx*"}**. ● If **template** is set to **tag_rule**, set **params** to **{"num": "*xxx*"}**.
tag_selectors	Yes	Array of **objects**	Image tags that are not counted in the retention policy.

Table 4-114 tag_selectors parameter description

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

- Example request

```
PATCH https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/retentions/1093
```

Body:

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

Response

- Response parameters

  N/A

- Example response

  ```
  {}
  ```

Status Code

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

4.10.6 Obtaining Image Retention Records

Function

Obtain image retention records.

URI

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

Table 4-115 describes the parameters.

Table 4-115 Parameter description

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

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset.
limit	No	Integer	Count.

Request

- Request parameters

  N/A

- Example request

```
GET https://{Endpoint}/v2/manage/namespaces/group/repos/busybox/retentions/histories
```

Response

- Response parameters

Table 4-116 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 4-117 Response body parameter description

Parameter	Type	Description
retention_log	Array of **objects**	Image retention log.
total	Integer	Total number.

Table 4-118 retention_log parameter description

Parameter	Type	Description
created_at	String	Creation time
id	Integer	ID.
namespace	String	Organization name.

Parameter	Type	Description
repo	String	Image repository name.
retention_id	Integer	Image retention policy ID.
rule_type	String	Policy.
tag	String	Image tag.

- Example response

```
{
    "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

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

4.11 Temporary Login Command

4.11.1 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

For details about parameters, see **Table 4-119**.

Table 4-119 Parameter description

Parameter	Mandatory	Type	Description
projectname	No	String	Project name. The default value is the region name.

Request Parameters

Table 4-120 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format). The value can be **application/json;charset=utf-8** or **application/json**. The default value is **application/json**.
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API used to obtain a user token. The value of **X-Subject-Token** in the response header is the user token.

Response Parameters

Status code: 200

Table 4-121 Response body parameters

Parameter	Type	Description
auths	Map<String,**AuthInfo**>	Authentication information.

Table 4-122 AuthInfo

Parameter	Type	Description
auth	String	Base64-encoded authentication information.

Example Request

POST https://{endpoint}/v2/manage/utils/secret

```
POST https://{endpoint}/v2/manage/utils/secret
```

Example Response

Status code: 200

Creation succeeded.

```
{
  "auths" : {
    "swr.xxx" : {
      "auth" : "Y24tbm9ydGg******hhMTgzMGVmN2RhZjJm"
    }
  }
}
```

Status Codes

Status Code	Description
200	Creation succeeded.
400	Request error.
401	Authentication failed.
500	Internal error.

4.12 Quota Management

4.12.1 Obtaining Quota Information

Description

Obtain quota information.

Constraints

N/A

URI

GET /v2/manage/projects/{project_id}/quotas

Table 4-123 Path parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

Table 4-124 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	Message body type (format). The value can be **application/json;charset=utf-8 application/json** The default value is **application/json**.
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API used to obtain a user token. The value of **X-Subject-Token** in the response header is the user token.

Response Parameters

Status code: 200

Table 4-125 Response body parameter

Parameter	Type	Description
quotas	Array of **ShowQuota** objects	List of quotas.

Table 4-126 ShowQuota

Parameter	Type	Description
quota_limit	Integer	Quota limit.
quota_key	String	Quota type.
unit	String	Unit.

Parameter	Type	Description
used	Integer	Used quota.

Example Request

```
GET https://{endpoint}/v2/manage/projects/{project_id}/quotas
```

Example Response

Status code: 200

Obtaining the quota succeeded.

```
{
  "quotas" : [ {
    "quota_limit" : 15,
    "quota_key" : "namespace",
    "unit" : "",
    "used" : 1
  } ]
}
```

Status Codes

Status Codes	Description
200	Obtaining the quota succeeded.
400	Request error.
401	Authentication failed.
500	Internal error.

Error Codes

For details, see section **Error Codes** in *API Reference*.

5 Example Applications

5.1 Creating an Image Repository

Scenario

This section describes how to create and delete an image repository using APIs. For details on how to call APIs, see **Calling APIs**.

Involved APIs

In this example, the following APIs are used:

- **Creating an Organization**
- **Querying the List of Organizations**
- **Creating an Image Repository**
- **Querying the Brief Information on an Image Repository**

Procedure

1. Create an organization.
   - URI format

     POST /v2/manage/namespaces

     For details, see **Creating an Organization**.
   - Example request

     POST https://{*endpoint*}/v2/manage/namespaces

     Body:
     ```
     {
         "namespace": "group"
     }
     ```
   - Example response
     ```
     {}
     ```
2. Query the list of organizations.
   - URI format

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

For details, see **Querying the List of Organizations**.

- Example request

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

- Example response

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

3. Create an image repository.
   - URI format

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

     For details, see **Creating an Image Repository**.

   - Example request

     POST https://{*endpoint*}/v2/manage/namespaces/{*namespace*}/repos?repository=test&category=linux&description=test&is_public=false

   - Example response

```
{}
```

4. Query the brief information on an image repository.
   - URI format

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

     For details, see **Querying the Brief Information on an Image Repository**.

   - Example request

     GET https://{*endpoint*}/v2/manage/namespaces/{*namespace*}/repos/{*repository*}

   - Example response

```
{
   "category": "other",
   "created": "2020-10-30T11:35:02.939134Z",
   "creator_id": "ff4f44473a1d4ab3847906410af2d152",
   "creator_name": "test",
   "description": "",
   "domain_id": "8b1e8153e5c4414fa57c356319b633fb",
   "id": 379367,
   "internal_path": ,
   "is_public": true,
   "name": "test",
   "ns_id": 7,
   "num_download": 3,
   "num_images": 1,
   "path": ,
   "priority": 0,
   "size": 1304902,
   "updated": "2020-10-30T11:47:46.084808Z",
   "url": ""
}
```

5.2 Adding Image Permissions

Scenario

This section describes how to add image permissions using APIs. For details on how to call APIs, see **Calling APIs**.

Involved APIs

In this example, the following APIs are used:

- **Creating Image Permissions**
- **Querying Image Permissions**

Prerequisites

An image repository has been created. For details on how to create an image repository, see **Creating an Image Repository**.

Procedure

1. Add image permissions.
   - URI format

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

     For details, see **Creating Image Permissions**.
   - Example request

     POST https://{*endpoint*}/v2/manage/namespaces/group/repos/test/access

     Body:

     ```
     [
         {
             "user_id": "fb3f175c1fd146ab8cdae3272be*****",
             "user_name": "test",
             "auth": 7
         }
     ]
     ```
   - Example response

     ```
     {}
     ```
2. Query image permissions.
   - URI format

     GET https://{*endpoint*}/v2/manage/namespaces/{*namespace*}/repos/{*repository*}/access

     For details, see **Querying Image Permissions**.
   - Example request

     GET /v2/manage/namespaces/group/repos/test/access
   - Example response

     ```
     {
         "id": 385550,
         "name": "test",
         "others_auths": [
     ```

```
        {
            "auth": 7,
            "user_id": "fb3f175c1fd146ab8cdae3272be*****",
            "user_name": "test"
        }
    ],
    "self_auth": {
        "auth": 7,
        "user_id": "fb3f175c1fd146ab8cdae3272be*****",
        "user_name": "test"
    }
}
```

5.3 Adding an Image Retention Policy

Scenario

This section describes how to add an image retention policy using APIs. For details on how to call APIs, see **Calling APIs**.

Involved APIs

In this example, the following APIs are used:

- **Adding an Image Retention Policy**
- **Deleting an Image Retention Policy**

Procedure

1. Add an image retention policy.
   - URI format

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

     For details, see **Adding an Image Retention Policy**.
   - Example request

     POST https://{*endpoint*}/v2/manage/namespaces/group/repos/test/retentions

     Body:

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

- Example response

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

2. Delete an image retention policy.
   - URI format

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

     For details, see **Deleting an Image Retention Policy**.

   - Example request

     DELETE https://{*endpoint*}/v2/manage/namespaces/group/repos/test/retentions/34

   - Example response

```
{}
```

6 Appendixes

6.1 Status Codes

- Normal

Returned Value	Description
200 OK	The results of GET and PUT operations are returned as expected.
201 Created	The results of the POST operation are returned as expected.
202 Accepted	The request has been accepted for processing.
204 No Content	The results of the DELETE operation are returned as expected.

- Abnormal

Returned Value	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You must enter a username and password to access the requested page.
403 Forbidden	You are forbidden to access the requested page.
404 Not Found	The server cannot find the requested page.
405 Method Not Allowed	You are not allowed to use the method specified in the request.
406 Not Acceptable	The response generated by the server cannot be accepted by the client.

Returned Value	Description
407 Proxy Authentication Required	You must use the proxy server for authentication so that the request can be processed.
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	Failed to complete the request because of a service error.
501 Not Implemented	Failed to complete the request because the server does not support the requested function.
502 Bad Gateway	Failed to complete the request because the request is invalid.
503 Service Unavailable	Failed to complete the request. The service is unavailable.
504 Gateway Timeout	A gateway timeout error occurred.

6.2 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.

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

Status Code	Error Code	Description
401	SVCSTG.SWR.4010000	Authentication failed. Obtain a new token and try again.

Status Code	Error Code	Description
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.
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.

Status Code	Error Code	Description
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.
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.

Status Code	Error Code	Description
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.
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.

Status Code	Error Code	Description
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.
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.

Status Code	Error Code	Description
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.
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.

Status Code	Error Code	Description
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	**Content-Type** must be set to **multipart/form-data**.
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	**Type** must be set to **app**.
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	**is_public** must be set to **true** or **false**.
400	SVCSTG.SWR.4001010	**Type** must be set to **app** or **image**.
400	SVCSTG.SWR.4001011	**Type** must be set to **app**, **image** or left blank.
413	SVCSTG.SWR.4131012	The description exceeds the maximum length allowed.
400	SVCSTG.SWR.4001013	**ak** or **sk** 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 **is_public** and **manifests** must be set.
400	SVCSTG.SWR.4001018	Incorrect request body type.

Status Code	Error Code	Description
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	**Status** must be set to **done** or **error**.
400	SVCSTG.SWR.4001028	Invalid organization name.
400	SVCSTG.SWR.4001029	Invalid image name.
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.

Status Code	Error Code	Description
400	SVCSTG.SWR.4001041	Invalid **domain**.
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 **SWRServer** 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.
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.

Status Code	Error Code	Description
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	Cross-account error. 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.
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.

Status Code	Error Code	Description
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.
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	Failed to create the image. The maximum number of images have been reached.

Status Code	Error Code	Description
403	SVCSTG.SWR.4037020	You do not have the permission to operate the image repository resource. Please improve your organization permissions.

6.3 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 a project ID by calling the API used to **query 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.

6.4 Obtaining the Username and User ID

Scenarios

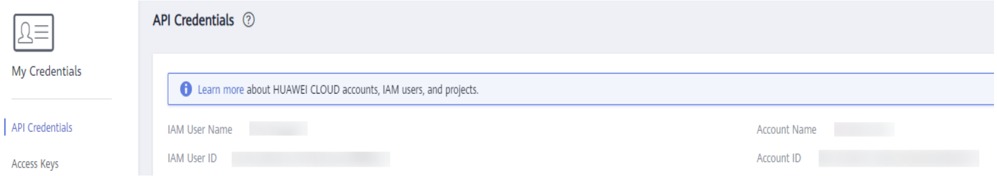
When calling APIs, you need to fill in your domain name and domain ID in certain URLs. To do so, you need to obtain domain name and domain ID on the console first. For details, see **Procedure**.

Procedure

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

   On the displayed **API Credentials** page, view the IAM username and IAM user ID.

Figure 6-1 API credentials



My Credentials

API Credentials

Access Keys

API Credentials

Learn more about HUAWEI CLOUD accounts, IAM users, and projects.

IAM User Name

IAM User ID

Account Name

Account ID

7 Change History

Release Date	Description
2022-9-30	This issue is the first official release.