Data Encryption Workshop

API Reference

Issue	01
Date	2026-03-19

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud 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

Data Encryption Workshop (DEW) is a comprehensive cloud data encryption service. It provides Key Management Service (KMS), Cloud Secret Management Service (CSMS), and Key Pair Service (KPS), helping you secure your data and keys and simplify key management. DEW uses hardware security modules (HSMs) to protect your keys. It can be integrated with other cloud services to address data security, key security, and key management issues. Additionally, DEW allows you to develop customized encryption applications.

Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. Obtain the regions and endpoints from the enterprise administrator.

Concepts

- Account

  An account has full access permissions for all the resources and cloud services under it. It can reset user passwords and grant users permissions. The account is a payment entity and should not be used to perform routine management. For security purposes, create IAM users and grant them permissions for routine management.

- User

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

  The username, and password will be required for API authentication.

- 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 as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.

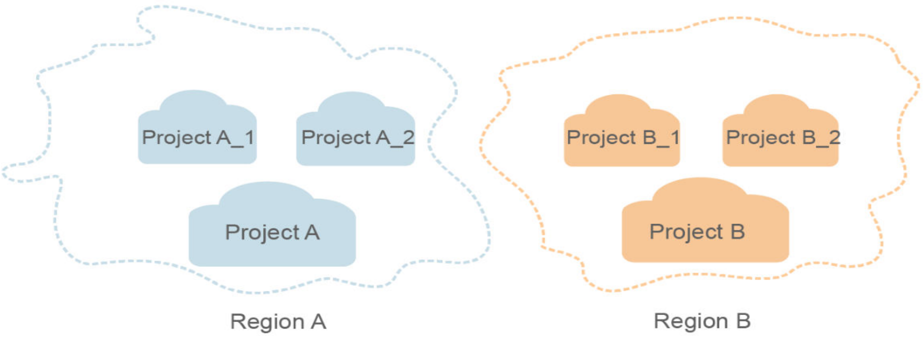
- Availability zone (AZ)

  An AZ comprises one or multiple physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Compute, 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 support cross-AZ high-availability systems.

- Project

  Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

  Figure 1-1 Project isolation model



- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated from each other. An enterprise project can contain resources in multiple regions, and resources can be directly transferred between enterprise projects.

2 Calling APIs

2.1 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.

- **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**.

- **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**.

- **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.

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:

- **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://{{endpoint}}/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:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies a user token only for token-based API authentication. The user token is a response to the API used to obtain a user token. This API is the only one that does not require authentication.

NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For more information, see **AK/SK-based 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://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
```

Request Body

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. Set ***username*** to the name of a user, ***domainname*** to the name of the account that the user belongs to, ************** to the user's login password, and ***xxxxxxxxxxxxxxxxxx*** to the project name. You can learn more information about projects 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://{{endpoint}}/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.

2.2 Authentication

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

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. This method is recommended because it provides higher security than token-based authentication.

Token-based Authentication

NOTE

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

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

The token can be obtained by calling the required API. For more information, see "Obtaining a User Token". A project-level token is required for calling this API, that is, **auth.scope** must be set to **project** in the request body. Example:

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

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 **ABCDEFG....**, add **X-Auth-Token: ABCDEFG....** to a request as follows:

```

GET https://{{endpoint}}/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFG....
```

AK/SK-based Authentication

NOTE

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

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

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

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see **API Signature Guide**.

NOTICE

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

2.3 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**.

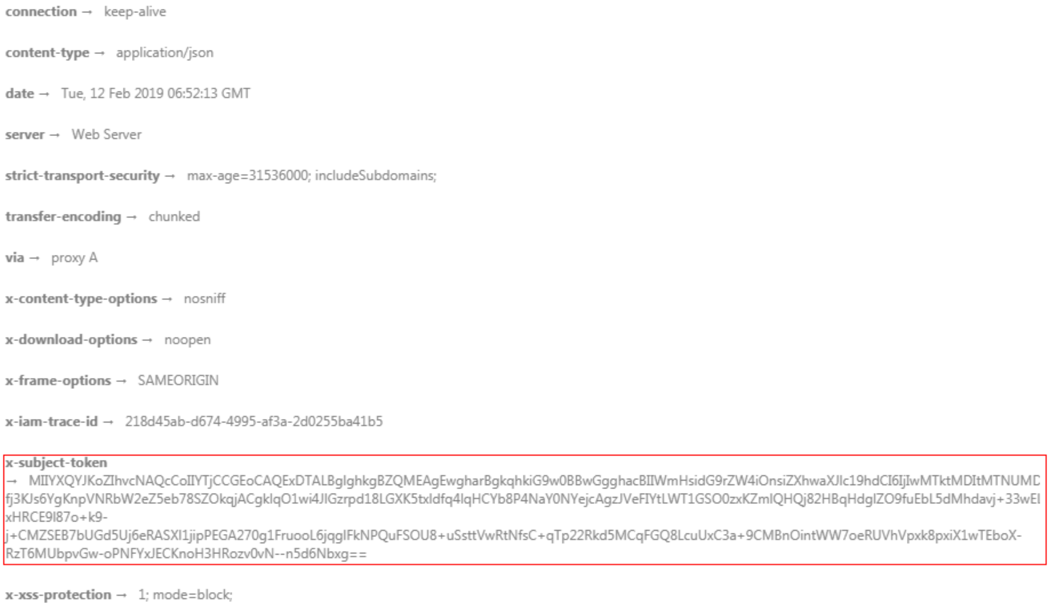
For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

A response header corresponds to a request header, for example, **Content-Type**.

Figure 2-1 shows the response header for the API of obtaining a user token, in which **x-subject-token** is the desired user token. Then, you can use the token to authenticate the calling of other APIs.

Figure 2-1 Header 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
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5txldfq4lqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1GSO0zxKZmIQHQj82HBqHdgIZO9fuEbL5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

x-xss-protection → 1; mode=block;
```

(Optional) Response Body

A response body is generally returned in a structured format, corresponding to the **Content-Type** in the response header, and is used to transfer content other than the response header.

The following shows part of the response body for the API to obtain a user token. For the sake of space, only part of the content is displayed here.

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

If an error occurs during API calling, the system returns an error code and a message to you. The following shows the format of an error response body:

```
{
    "error": {
        "message": "The request you have made requires authentication.",
        "title": "Unauthorized"
    }
}
```

In the preceding information, **error_code** is an error code, and **error_msg** describes the error.

3 APIs

3.1 CMK Management

3.1.1 Creating a CMK

Function

This API is used to create customer master keys (CMKs) used to encrypt data encryption keys (DEKs).

NOTE

Default Master Keys are created by services integrated with KMS. Names of Default Master Keys end with **/default**. Therefore, in naming your CMKs, do not choose those ending with **/default**.

Enterprise project users' Default Master Keys belong to their default enterprise projects. The keys and cannot be moved to other enterprise projects, but can be used for cloud-based encryption in non-default enterprise projects to meet compliance requirements.

URI

- URI format

  POST /v1.0/{project_id}/kms/create-key
- Parameter description

Table 3-1 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 3-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-3 Request parameters

Parameter	Mandatory	Type	Description
key_alias	Yes	String	Alias of a non-default master key (The alias's length ranges from 1 to 255 characters and matches the regular expression **^[a-zA-Z0-9:/_-]{1,255}$**. In addition, it must be different from the alias of a Default Master Key created by the system.)
key_spec	No	String	Key generation algorithm. The default value is **AES_256**. The value can be: ● **AES_256** ● **RSA_2048** ● **RSA_3072** ● **RSA_4096** ● **EC_P256** ● **EC_P384**
key_usage	No	String	Key usage. The default value is **ENCRYPT_DECRYPT** for a symmetric key and **SIGN_VERIFY** for an asymmetric key. Possible values are as follows: ● For AES_256 symmetric keys, the default value is **ENCRYPT_DECRYPT**. ● For RSA asymmetric keys, select **ENCRYPT_DECRYPT** or **SIGN_VERIFY**. The default value is **SIGN_VERIFY**. ● For ECC asymmetric keys, the default value is **SIGN_VERIFY**.

Parameter	Mandatory	Type	Description
enterprise_project_id	No	String	Enterprise project ID. ● If the enterprise project function is not enabled, you do not need to set this parameter. ● If the enterprise project function is enabled, you can set this parameter when creating a resource. If this parameter is not specified, the resource you create will be put under the default enterprise project (whose project ID is **0**). If you do not have the permission to create resources under the default enterprise project, an error will be reported.
key_description	No	String	CMK description (The value ranges from 0 to 255 characters.)
origin	No	String	Origin of a CMK. The default value is **kms**. The following values are enumerated: ● **kms** indicates that the CMK material is generated by KMS. ● **external** indicates that the CMK material is imported.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Responses

Table 3-4 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 3-5**.

Table 3-5 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID

Parameter	Mandatory	Type	Description
domain_id	Yes	String	User domain ID

Examples

The following example describes how to create a CMK with an alias of **test**.

- Example request

```
{
    "key_alias": "test"
}
```

- Example response

```
{
    "key_info": {
        "key_id": "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
        "domain_id": "b168fe00ff56492495a7d22974df2d0b"
    }
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-6 lists the normal status code returned by the response.

Table 3-6 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.2 Enabling a CMK

Function

This API allows you to enable a CMK. Only an enabled CMK can be used.

NOTE

Only a disabled CMK can be enabled.

URI

- URI format
  POST /v1.0/{project_id}/kms/enable-key
- Parameter description

Table 3-7 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-8 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-9 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte key ID that matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**. For example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-10 Response parameter

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 3-11**.

Table 3-11 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
key_state	Yes	String	CMK status: ● **2** indicates that the CMK is enabled. ● **3** indicates that the CMK is disabled. ● **4** indicates that the CMK is scheduled for deletion.

Example

The following example describes how to enable a CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

- Example request

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f"
}
```

- Example response

```
{
    "key_info": {
        "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
        "key_state": "2"
    }
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-12 lists the normal status code returned by the response.

Table 3-12 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.3 Disabling a CMK

Function

This API allows you to disable a CMK. A disabled CMK cannot be used.

NOTE

Only an enabled CMK can be disabled.

URI

- URI format

  POST /v1.0/{project_id}/kms/disable-key
- Parameter description

Table 3-13 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-14 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-15 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte key ID that matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**. For example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-16 Response parameter

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 3-17**.

Table 3-17 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
key_state	Yes	String	CMK status: ● **2** indicates that the CMK is enabled. ● **3** indicates that the CMK is disabled. ● **4** indicates that the CMK is scheduled for deletion.

Example

The following example describes how to disable a CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

- Example request
  ```
  {
      "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f"
  }
  ```
- Example response
  ```
  {
      "key_info": {
          "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
          "key_state": "3"
      }
  }
  ```
  or
  ```
  {
      "error": {
          "error_code": "KMS.XXXX",
          "error_msg": "XXX"
      }
  }
  ```

Status Codes

Table 3-18 lists the normal status code returned by the response.

Table 3-18 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.4 Scheduling the Deletion of a CMK

Function

This API enables you to schedule the deletion of a CMK. A CMK can be scheduled to be deleted after 7 to 1,096 days.

URI

- URI format

  POST /v1.0/{project_id}/kms/schedule-key-deletion
- Parameter description

Table 3-19 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-21 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte key ID that matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**. For example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**
pending_days	Yes	String	Number of days after which a CMK is scheduled to be deleted (The value ranges from **7** to **1,096**.)
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-22 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID

Parameter	Mandatory	Type	Description
key_state	Yes	String	CMK status: ● **2** indicates that the CMK is enabled. ● **3** indicates that the CMK is disabled. ● **4** indicates that the CMK is scheduled for deletion.

Example

The following example describes how to schedule deletion of a CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

- Example request
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "pending_days": "7"
}
```
- Example response
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "key_state": "4"
}
```
  or
```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-23 lists the normal status code returned by the response.

Table 3-23 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.5 Canceling the Scheduled Deletion of a CMK

Function

This API enables you to cancel the scheduled deletion of a CMK.

NOTE

You can cancel the scheduled deletion for a CMK only when the CMK's status is **Scheduled deletion**.

URI

- URI format
  POST /v1.0/{project_id}/kms/cancel-key-deletion
- Parameter description

Table 3-24 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-25 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-26 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte key ID that matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**. For example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-27 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
key_state	Yes	String	CMK status: ● **2** indicates that the CMK is enabled. ● **3** indicates that the CMK is disabled. ● **4** indicates that the CMK is scheduled for deletion.

Example

The following example describes how to cancel the scheduled deletion of a CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

- Example request
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f"
}
```
- Example response
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "key_state": "3"
}
```
or
```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-28 lists the normal status code returned by the response.

Table 3-28 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.6 Querying the List of CMKs

Function

This API allows you to query the list of all CMKs.

URI

- URI format

  POST /v1.0/{project_id}/kms/list-keys
- Parameter description

Table 3-29 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 3-30 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-31 Request parameters

Parameter	Mandatory	Type	Description
limit	No	String	This parameter specifies the number of entries returned. If the specified number is smaller than the actual number of existing entries, **true** will be returned for the response parameter **truncated**, indicating that the query results will be displayed in separate pages. The value is within the range of the maximum number of CMKs, for example, **100**.

Parameter	Mandatory	Type	Description
marker	No	String	This parameter marks the starting location in a pagination query. If the **truncated** value is **true**, you can send consecutive requests to obtain more record entries. The **marker** value must be set to the **next_marker** value in the response, for example, **10**.
enterprise_project_id	No	String	Enterprise project ID. ● If the enterprise project function is not enabled, you do not need to set this parameter. ● If the enterprise project function is enabled, you can set this parameter when querying a resource. If this parameter is not specified, the system searches for the required resource in all the enterprise projects that you have permissions for. In this case, the value of **enterprise_project_id** is **all**. The parameter value must meet one of the following requirements: – Is **all** – Is **0** – Matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**.
key_state	No	String	State of a CMK that matches the regular expression **^[1-5]{1}$**. The following values are enumerated: ● **1** indicates that the CMK is waiting to be activated. ● **2** indicates that the CMK is enabled. ● **3** indicates that the CMK is disabled. ● **4** indicates that the CMK is scheduled for deletion. ● **5** indicates that the CMK is waiting to be imported.

Parameter	Mandatory	Type	Description
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 3-32 Response parameters

Parameter	Mandatory	Type	Description
keys	Yes	Array of strings	List of CMK IDs
key_details	Yes	Array of objects	Key details list. For details, see **Table 3-38**.
next_marker	Yes	String	This parameter indicates the **marker** value required for obtaining the next page of query results. If the **truncated** value is **false**, the **next_marker** parameter is left blank.
total	Yes	Integer	Total number of keys.
truncated	Yes	String	This parameter indicates whether there are more results displayed in another page. ● If the value is **true**, there are more results. ● If the value is **false**, the current page is the last page.

Examples

The following shows an example when **limit** is set to **2** and **marker** is set to **1**.

- Example request

```
{
   "limit": "2",
   "marker": "1"
}
```

- Example response

```
{
   "keys": [
      "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
      "2e258389-bb1e-4568-a1d5-e1f50adf70ea"
   ],
   "key_details": [
      {
      "key_id":"0d0466b0-e727-4d9c-b35d-f84bb474a37f",
      "domain_id":"00074811d5c27c4f8d48bb91e4a1dcfd",
      "key_alias":"caseuirpr",
      "realm":"aaaa",
      "key_description":"123",
      "creation_date":"1502799822000",
      "scheduled_deletion_date":"",
      "key_state":"2",
      "default_key_flag":"0",
      "key_type":"1",
      "expiration_time":"1501578672000",
      "origin":"kms"
},
      {
      "key_id":"2e258389-bb1e-4568-a1d5-e1f50adf70ea",
      "domain_id":"00074811d5c27c4f8d48bb91e4a1dcfd",
      "key_alias":"casehvniz",
      "realm":"aaaa",
      "key_description":"234",
      "creation_date":"1502799820000",
      "scheduled_deletion_date":"",
      "key_state":"2",
      "default_key_flag":"0",
      "key_type":"1",
      "expiration_time":"1501578673000",
      "origin":"kms"
}
     ],
   "next_marker": "",
   "truncated": "false",
   "total":2
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-33 lists the normal status code returned by the response.

Table 3-33 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.7 Querying the Information About a CMK

Function

This API allows you to query the details about a CMK.

URI

- URI format
  POST /v1.0/{project_id}/kms/describe-key
- Parameter description

Table 3-34 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-35 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-36 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-37 Response parameter

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 3-38**.

Table 3-38 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
domain_id	Yes	String	User domain ID

Parameter	Mandatory	Type	Description
key_alias	Yes	String	Alias of a CMK
realm	Yes	String	Region where a CMK resides
key_description	Yes	String	Description of a CMK
key_spec	Yes	String	Key generation algorithm. Possible values are as follows: ● **AES_256** ● **RSA_2048** ● **RSA_3072** ● **RSA_4096** ● **EC_P256** ● **EC_P384** ● **SM2**
key_usage	Yes	String	Key usage. Possible values are as follows: ● **ENCRYPT_DECRYPT** ● **SIGN_VERIFY** ● **GENERATE_VERIFY_MAC**
creation_date	Yes	String	Time when a key is created. The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
scheduled_deletion_date	Yes	String	Time when a key will be deleted as scheduled. The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
key_state	Yes	String	State of a CMK: ● **1** indicates that the CMK is waiting to be activated. ● **2** indicates that the CMK is enabled. ● **3** indicates that the CMK is disabled. ● **4** indicates that the CMK is scheduled for deletion. ● **5** indicates that the CMK is waiting to be imported.
default_key_flag	Yes	String	Identification of a Master Key. The value **1** indicates a Default Master Key, and the value **0** indicates a CMK.

Parameter	Mandatory	Type	Description
key_type	Yes	String	Type of a CMK
expiration_time	Yes	String	Expiration time of the key material. It is expressed in the form of a time stamp, the total number of seconds since January 1, 1970.
origin	Yes	String	Origin of a CMK. The default value is **kms**. The following values are enumerated: ● **kms** indicates that the CMK material is generated by KMS. ● **external** indicates that the CMK material is imported.
sys_enterprise_project_id	Yes	String	Enterprise project ID. Its default value is **0**. For users who have enabled the enterprise project function, this value indicates that resources are in the default enterprise project. For users who have not enabled the enterprise project function, this value indicates that resources are not in the default enterprise project.

Example

The following example describes how to query the information of a CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

- Example request
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f"
}
```
- Example response
```
{
    "key_info": {
        "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
        "domain_id": "b168fe00ff56492495a7d22974df2d0b",
        "key_alias": "kms_test",
        "realm": "aaa",
        "key_description": "",
        "creation_date": "1472442386000",
        "scheduled_deletion_date": "",
        "key_state": "2",
        "default_key_flag": "0",
        "key_type": "1",
        "expiration_time":"1501578672000",
        "origin":"kms"
        ,
        "sys_enterprise_project_id ": "0",
    }
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-39 lists the normal status code returned by the response.

Table 3-39 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.8 Creating a Random Number

Function

This API generates a 512-bit random number.

URI

- URI format
  POST /v1.0/{project_id}/kms/gen-random
- Parameter description

Table 3-40 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 3-41 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-42 Request parameters

Parameter	Mandatory	Type	Description
random_data_length	Yes	String	Number of bits of a random number. The value is **512**.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 3-43 Response parameters

Parameter	Mandatory	Type	Description
random_data	Yes	String	Random numbers are expressed in hexadecimal format. Two characters indicate one byte. Length of a random number must be consistent with the **random_data_length** value entered by a user.

Examples

The following example describes how to create a random number with the length of **512** bits.

- Example request

```
{
    "random_data_length": "512"
}
```

- Example response

```
{
    "random_data":
"5791C223E87124AB9FC29B5A8AC60BE4B98D168F47A58BB2A88833E40D6ED32D57E2AAB5410492EB
25096873F9CE3D45E0D22F820A5AB4EEADC33A1A6AE780F1"
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-44 lists the normal status code returned by the response.

Table 3-44 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.9 Creating a DEK

Function

This API allows you to create a DEK. A returned result includes the plaintext and the ciphertext of a DEK.

URI

- URI format

  POST /v1.0/{project_id}/kms/create-datakey

- Parameter description

Table 3-45 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-46 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-47 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.
encryption_context	No	Object	Key-value pairs with a maximum length of 8192 characters. This parameter is used to record resource context information, excluding sensitive information, to ensure data integrity. If this parameter is specified during encryption, it is also required for decryption. Example: {"**Key1**":"**Value1**","**Key2**":"**Value2**"}

Parameter	Mandatory	Type	Description
datakey_length	No	String	Number of bits of a key. The value is **512**.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-48 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
plain_text	Yes	String	The plaintext of a DEK is expressed in hexadecimal format, and two characters indicate one byte.
cipher_text	Yes	String	The ciphertext of a DEK is expressed in hexadecimal format, and two characters indicate one byte.

Example

The following example describes how to create a DEK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and length is **512** bits.

- Example request

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "datakey_length": "512"
}
```

- Example response

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "plain_text": 
"8151014275E426C72EE7D44267EF11590DCE0089E19863BA8CC832187B156A72A5A17F17B5EF0D525
872C59ECEB72948AF85E18427F8BE0D46545C979306C08D",
    "cipher_text": 
"020098009EEAFCE122CAA5927D2E020086F9548BA1675FDB022E4ECC01B96F2189CF4B85E78357E73
E1CEB518DAF7A4960E7C7DE8885ED3FB2F1471ABF400119CC1B20BD3C4A9B80AF590EFD0AEDABFDB
B0E2B689DA7B6C9E7D3C5645FCD9274802586BE63779471F9156F2CDF07CD8412FFBE923064303436
3662302D653732372D346439632D623335642D663834626234373461333766000000045B05321483B
D9F9561865EE7DFE9BE267A42EB104E98C16589CE46940B18E52"
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-49 lists the normal status code returned by the response.

Table 3-49 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.10 Creating a Plaintext-Free DEK

Function

This API allows you to create a plaintext-free DEK, that is, the returned result of this API includes only the ciphertext of the DEK.

URI

- URI format

  POST /v1.0/{project_id}/kms/create-datakey-without-plaintext
- Parameter description

Table 3-50 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-51 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	application/json

Table 3-52 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.
encryption_context	No	Object	Key-value pairs with a maximum length of 8192 characters. This parameter is used to record resource context information, excluding sensitive information, to ensure data integrity. If this parameter is specified during encryption, it is also required for decryption. Example: {"**Key1**":"**Value1**","**Key2**":"**Value2**"}
datakey_length	No	String	Number of bits of a key. The value is **512**.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-53 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
cipher_text	Yes	String	The ciphertext of a DEK is expressed in hexadecimal format, and two characters indicate one byte.

Example

The following example describes how to create a plaintext free DEK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

- Example request
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "datakey_length": "512"
}
```
- Example response
```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "cipher_text":
"020098005CDC28E29EC3230AA42E8985FBABA095037D6474C64519C9B564AB28B15739C88E7E88750
0D1094973C2DC16353DB7ED3946C73339517AB1E983D521F9E9D700DC5D9C42F557EBF3F608E3CBB
EE0BC68136EE7D2A49117E00332BAC4AE4ED805EB6068FA900C5A8019BFE2C2651BE3E130643034363
662302D653732372D346439632D623335642D66383462623437346133376600000000F160727EBDB83
400C21D80D713B49D3A2C37F24AE160E7BB3DAC025ADC0C45E3"
}
```
  or
```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-54 lists the normal status code returned by the response.

Table 3-54 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.11 Encrypting a DEK

Function

This API enables you to encrypt a DEK using a specified CMK.

URI

- URI format

  POST /v1.0/{project_id}/kms/encrypt-datakey
- Parameter description

Table 3-55 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-56 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-57 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.
encryption_context	No	Object	Key-value pairs with a maximum length of 8192 characters. This parameter is used to record resource context information, excluding sensitive information, to ensure data integrity. If this parameter is specified during encryption, it is also required for decryption. Example: {"**Key1**":"**Value1**","**Key2**":"**Value2**"}
plain_text	Yes	String	Hexadecimal character string concatenated from plaintext of a DEK and the plaintext digest (32-byte character string generated using SHA256) For details, see **Example**.
datakey_plain_length	Yes	String	Number of bytes of a DEK in plaintext. The value range is 1 to 1024.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-58 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
cipher_text	Yes	String	The ciphertext of a DEK is expressed in hexadecimal format, and two characters indicate one byte.
datakey_length	Yes	String	Number of bytes in the length of a DEK

Example

In the following example, the 512-bit plaintext DEK (**7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4b18ff0512525e527b10331100f357bf42125d8d5ced94f**) generated from the customer master key whose key ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** can be obtained through the API in **Creating a DEK**.

The digest of the plaintext DEK is **fbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0a9ee0305fb9540f797**. The method for calculating the digest is as follows:

```
//Digest calculation
public static byte[] sha256(byte[] cmkData) {
    byte[] digest = new byte[0];
   try {
       MessageDigest md = MessageDigest.getInstance("SHA-256");
         md.update(cmkData);
        digest = md.digest();
     } catch (Exception e) {
        System.out.println("calculate digest failure, exception is " + e.toString());
    }
     return digest;
}
//Convert the obtained digest into a hexadecimal character string.
public static String bytesToHexString(byte[] digest) {
        ...
}
```

The value of **plain_text** (a hexadecimal character string concatenated from plaintext of the DEK and the plaintext digest) is **7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4b18ff0512525e527b10331100f357bf42125d8d5ced94f fbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0a9ee0305fb9540f797**.

- Example request

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "plain_text":
"7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4b18ff
0512525e527b10331100f357bf42125d8d5ced94f
fbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0a9ee0305fb9540f797",
    "datakey_plain_length": "64"
}
```

- Example response

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "cipher_text":
"020098005273E14E6E8E95F5463BECDC27E80AF820B9FC086CB47861899149F67CF07DAFF2810B7D2
7BDF19AB7632488E0926A48DB2FC85BEA905119411B46244C5E6B8036C60A0B0B4842FFE6994518E89
C19B1C1D688D9043BCD6053EA7BA0652642CE59F2543C80669139F4F71ABB9BD9A243306430343636
62302D653732372D346439632D623335642D663834626234373461333766000000000D34457984F9730
D57F228C210FD22CA6017913964B21D4ECE45D81092BB9112E",
    "datakey_length": "64"
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-59 lists the normal status code returned by the response.

Table 3-59 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.12 Decrypting a DEK

Function

This API enables you to decrypt a DEK using a specified CMK.

NOTE

Data encryption results are used for decryption.

URI

- URI format

  POST /v1.0/{project_id}/kms/decrypt-datakey

- Parameter description

Table 3-60 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-61 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-62 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.

Parameter	Mandatory	Type	Description
encryption_context	No	Object	Key-value pairs with a maximum length of 8192 characters. This parameter is used to record resource context information, excluding sensitive information, to ensure data integrity. If this parameter is specified during encryption, it is also required for decryption. Example: {"**Key1**":"**Value1**","**Key2**":"**Value2**"}
cipher_text	Yes	String	This parameter indicates the hexadecimal character string of the DEK ciphertext and the metadata. The value is the **cipher_text** value in the encryption result of a DEK.
datakey_cipher_length	Yes	String	Number of bytes of a key. The value range is 1 to 1024.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-63 Response parameters

Parameter	Mandatory	Type	Description
data_key	Yes	String	Hexadecimal character string of the plaintext of a DEK
datakey_length	Yes	String	Number of bytes in the length of the plaintext of a DEK
datakey_dgst	Yes	String	Hexadecimal character string corresponding to the SHA-256 hash value of the plaintext of a DEK

Example

The following is an example about how to use a CMK (ID: **0d0466b0-e727-4d9c-b35d-f84bb474a37f**) to decrypt a DEK (ciphertext:

020098005273E14E6E8E95F5463BECDC27E80AF820B9FC086CB47861899149F67CF07DAFF2810B7D27BDF19AB7632488E0926A48DB2FC85BEA905119411B46244C5E6B8036C60A0B0B4842FFE6994518E89C19B1C1D688D9043BCD6053EA7BA0652642CE59F2543C80669139F4F71ABB9BD9A24330643034363662302D653732372D346439632D623335642D66383462623437346133376600000000D34457984F9730D57F228C210FD22CA6017913964B21D4ECE45D81092BB9112E; length: **64** bits).

- Example request

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "datakey_cipher_length": "64",
    "cipher_text": 
"020098005273E14E6E8E95F5463BECDC27E80AF820B9FC086CB47861899149F67CF07DAFF2810B7D2
7BDF19AB7632488E0926A48DB2FC85BEA905119411B46244C5E6B8036C60A0B0B4842FFE6994518E89
C19B1C1D688D9043BCD6053EA7BA0652642CE59F2543C80669139F4F71ABB9BD9A243306430343636
62302D653732372D346439632D623335642D66383462623437346133376600000000D34457984F9730
D57F228C210FD22CA6017913964B21D4ECE45D81092BB9112E"
}
```

- Example response

```
{
    "data_key": 
"00000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000",
    "datakey_length": "64",
    "datakey_dgst": "F5A5FD42D16A20302798EF6ED309979B43003D2320D9F0E8EA9831A92759FB4B"
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-64 lists the normal status code returned by the response.

Table 3-64 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.13 Signing Data

Function

- This API is used to use the private key of an asymmetric key to digitally sign a message or digest.

Constraints

- Only the asymmetric key whose **key_usage** is **SIGN_VERIFY** can be used for signature.

URI

POST /v1.0/{project_id}/kms/sign

Table 3-65 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 3-66 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)
Content-Type	Yes	String	application/json

Table 3-67 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.
message	Yes	String	Message digest or message to be signed. The message must be encoded using Base64 and be less than 4096 bytes.
signing_algorithm	Yes	String	Signature algorithm. Possible values are as follows: ● **RSASSA_PSS_SHA_256** ● **RSASSA_PSS_SHA_384** ● **RSASSA_PSS_SHA_512** ● **RSASSA_PKCS1_V1_5_SHA_256** ● **RSASSA_PKCS1_V1_5_SHA_384** ● **RSASSA_PKCS1_V1_5_SHA_512** ● **ECDSA_SHA_256** ● **ECDSA_SHA_384** ● **ECDSA_SHA_512**

Parameter	Mandatory	Type	Description
message_type	No	String	Message type. The default value is **DIGEST**. Possible values are as follows: ● **DIGEST** (message digest) ● **RAW** (original message)
sequence	No	String	A 36-byte serial number of a request message. Example: 919c82d4-8046-4722-9094-35c3c6524cff

Response Parameters

Status code: 200

Table 3-68 Response body parameters

Parameter	Type	Description
key_id	String	CMK ID
signature	String	Signature value, which is encoded using Base64

Status code: 400

Table 3-69 Response body parameter

Parameter	Type	Description
error	Object	Error message.

Table 3-70 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-71 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-72 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-73 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-74 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-75 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-76 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-77 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-78 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-79 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-80 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-81 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-82 ErrorDetail

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error information

Example Request

The following uses the RSASSA_PKCS1_V1_5_SHA_256 signature algorithm to sign the raw message.

```
{
 "key_id": "968d6cf0-feb6-42c6-bb30-d69f74f2d5f9",
 "message": "aGVsbG8g",
 "signing_algorithm": "RSASSA_PSS_SHA_256",
 "message_type": "RAW"
}
```

The following uses the RSASSA_PKCS1_V1_5_SHA_256 signature algorithm to sign the digest message.

```
{
 "key_id": "968d6cf0-feb6-42c6-bb30-d69f74f2d5f9",
 "message": "iNQmb9TmM40TuEX88olXnSCciXgjuSF9o+Fhk28DFYK=",
 "signing_algorithm": "RSASSA_PSS_SHA_256",
 "message_type": "DIGEST"
}
```

Example Response

Status code: 200

The following shows that the request for signing the raw message using the RSASSA_PKCS1_V1_5_SHA_256 signature algorithm is successful.

```
{
 "key_id": "968d6cf0-feb6-42c6-bb30-d69f74f2d5f9",
 "signature": "BqhL4PFPMNIXyEld3qviF7uqqnqlm9TcVCUN9FTRCr6KGreHIvwE4YuAc
+eLWVSCGRd3bQHhDOQ9GlWjixGengwBix1RPP0qxtn2p7kQxkC2j76VjKCwqAsAy4MyxjN8RNOdnVCpOObD
GoLxPHxUwNvSqZ6GxQKZ4cHPXVH0r/jH9csgk6IUr6ATyto+IcNWSvD03LfaNRQ
+Rvc5tOzNFpFrMnVl319UG9ANscq1ne67VW2uQIf74Osg9DYzbJTf/xqW5GFi3ZoeQUu
+gMxwgQp3pkuYhygjw6a8Qy9ZNMHmWnY199SzHrxgIq3ymQzUU5zrikKMColX2goPXf5fxQ=="
}
```

The following shows that the request for signing the digest message using the RSASSA_PKCS1_V1_5_SHA_256 signature algorithm is successful.

```
{
 "key_id": "968d6cf0-feb6-42c6-bb30-d69f74f2d5f9",
 "signature": "M8Gqrm7EyyCPckMs90D7IOlUPCMHhoBh+nz9ySvdbOi7JMrl0ei+2lb+CQ2ZJN+pu7mftotq7/
sHt0wWsDl8IOywYSBtWEmLW6AHnEPMykG/A9/Dp3kRuuKFoouCzWXeZyhIrzRUunAK5j5njcY/yTf6T8M
+zBy1nAApb8WcHUen9/j7+X348iOnsSuWNVfXxy3NX41v9kLn6x115UDA/798VLSoMbsjcXKgdf/
3GoZRYjcHxiX6s71/RWsQYme68qQN2B0q8Y9lk6rQxrw/AXHFoeaphYb7PriURRx0GxhOEEHb/9Tcr39Zlh3bbl/
2aF3ytJORWIqatLtqgJ4uEA=="
}
```

Status Codes

Status Code	Description
200	The request has succeeded.
400	Invalid request parameters.
401	Username and password are required to access the page requested.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

3.1.14 Verifying a Signature

Function

- This API uses the private key of an asymmetric key to verify a signature.

Constraints

- Only the asymmetric key whose **key_usage** is **SIGN_VERIFY** can be used for signature verification.

URI

POST /v1.0/{project_id}/kms/verify

Table 3-83 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 3-84 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)
Content-Type	Yes	String	application/json

Table 3-85 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	The value can be a key ID, alias (**key_alias**), or URN. ● Key ID: A 36-byte string that matches the **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$** regular expression, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f** ● Alias: An identifier of a key. The value starts with **alias/**, for example, **alias/4555**. ● URN: Each alias automatically matches a unique URN, for example, **kms:eu-de-ring0:3ba44455500dd43127:alias:4555**. **NOTE** The **alias_urn** generated during key alias creation is the URN.
message	Yes	String	Message digest or message to be signed. The message must be encoded using Base64 and be less than 4,096 bytes.
signature	Yes	String	Signature value to be verified, which is encoded using Base64.

Parameter	Mandatory	Type	Description
signing_algorithm	Yes	String	Signature algorithm. Possible values are as follows: • **RSASSA_PSS_SHA_256** • **RSASSA_PSS_SHA_384** • **RSASSA_PSS_SHA_512** • **RSASSA_PKCS1_V1_5_SHA_256** • **RSASSA_PKCS1_V1_5_SHA_384** • **RSASSA_PKCS1_V1_5_SHA_512** • **ECDSA_SHA_256** • **ECDSA_SHA_384** • **ECDSA_SHA_512**
message_type	No	String	Message type. The default value is **DIGEST**. Possible values are as follows: • **DIGEST** (message digest) • **RAW** (original message)
sequence	No	String	A 36-byte serial number of a request message. Example: **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 3-86 Response body parameters

Parameter	Type	Description
key_id	String	CMK ID
signature_valid	String	Whether the signature is valid. Its value can be **true** (valid) or **false** (invalid).

Status code: 400

Table 3-87 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-88 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-89 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-90 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-91 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-92 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-93 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-94 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-95 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-96 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-97 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-98 ErrorDetail

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error information

Status code: 504

Table 3-99 Response body parameter

Parameter	Type	Description
error	Object	Error message

Table 3-100 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "signing_algorithm" : "RSASSA_PKCS1_V1_5_SHA_256",
  "signature" : "jFUqQESGBc0j6k9BozzrP9YL4qk8/W9DZRvK6XXX...",
  "message" : "MmFiZWE0ZjI3ZGIxYTkzY2RmYmEzM2YwMTA1YmJjYw=="
}
```

Example Response

Status code: 200

The request has succeeded.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "signature_valid" : "true"
}
```

Status Codes

Status Code	Description
200	The request has succeeded.
400	Invalid request parameters.
401	Username and password are required to access the page requested.
403	Authentication failed.

Status Code	Description
404	The requested resource does not exist or is not found.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

3.1.15 Querying the Number of Instances

Function

This API is used to query the number of instances, that is, the number of CMKs created.

NOTE

Default Master Keys are automatically created by services and are not included in this query.

URI

- URI format

  GET /v1.0/{project_id}/kms/user-instances

- Parameter description

Table 3-101 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 3-102 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling an IAM API. The value of **X-Subject-Token** in the response header is the user token.

Responses

Table 3-103 Response parameters

Parameter	Mandatory	Type	Description
instance_num	Yes	Integer	Number of non-default CMKs

Examples

- Example request

  None

- Example response

```
{
    "instance_num": 15
}
```

  or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-104 lists the normal status code returned by the response.

Table 3-104 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.16 Querying the Quota of a User

Function

This API is used to query the quota of a user, that is, the allocated total number of CMKs that can be created by a user and the number of CMKs that has been created by the user.

NOTE

The quota does not include Default Master Keys.

URI

- URI format
  GET /v1.0/{project_id}/kms/user-quotas
- Parameter description

Table 3-105 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 3-106 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling an IAM API. The value of **X-Subject-Token** in the response header is the user token.

Responses

Table 3-107 Response parameters

Parameter	Mandatory	Type	Description
quotas	Yes	Object	Quota list. For details, see **Table 3-108**.

Table 3-108 quotas field description

Parameter	Mandatory	Type	Description
resources	Yes	Array of objects	Resource quota list. For details, see **Table 3-109**.

Table 3-109 resources field description

Parameter	Mandatory	Type	Description
type	Yes	String	Quota type. Enumerated values: ● **CMK** indicates a Customer Master Key. ● **grant_per_CMK** indicates the number of grants that can be created on a CMK.
used	Yes	Integer	Used quota
quota	Yes	Integer	Total quota

Examples

- Example request

  None

- Example response

```
{
    "quotas": {
        "resources": [
            {
                "type": "CMK",
                "used": 15,
                "quota": 20
            },
            {
                "type": "grant_per_CMK",
                "used": 15,
                "quota": 100
            }

        ]
    }
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-110 lists the normal status code returned by the response.

Table 3-110 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.17 Changing the Alias of a CMK

Function

This API enables you to change the alias of a CMK.

NOTE

- A Default Master Key (the alias suffix of which is **/default**) does not allow alias changes.
- A CMK in **Scheduled deletion** status does not allow alias changes.

URI

- URI format

  POST /v1.0/{project_id}/kms/update-key-alias
- Parameter description

Table 3-111 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-112 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-113 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte key ID that matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**. For example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**
key_alias	Yes	String	Alias of a CMK whose length is 1 to 255 characters and which matches the regular expression **^[a-zA-Z0-9:/_-]{1,255}$**. Suffix of the alias cannot be **/default**.
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-114 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 3-115**.

Table 3-115 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
key_alias	Yes	String	Alias of a CMK

Example

The following is an example about how to modify a CMK whose alias ID is **bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e** and alias is **test**.

- Example request

```
{
    "key_alias": "test",
    "key_id": "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e"
}
```

- Example response

```
{
    "key_info": {
        "key_id": "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
        "key_alias": "test"
     }
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-116 lists the normal status code returned by the response.

Table 3-116 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.1.18 Changing the Description of a CMK

Function

This API enables you to change the description of a CMK.

NOTE

- A Default Master Key (the alias suffix of which is **/default**) does not allow alias changes.
- A CMK in **Scheduled deletion** status does not allow description changes.

URI

- URI format

  POST /v1.0/{project_id}/kms/update-key-description

- Parameter description

Table 3-117 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Message

Table 3-118 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).
Content-Type	Yes	String	application/json

Table 3-119 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte key ID that matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**. For example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**
key_description	Yes	String	CMK description (The value ranges from 0 to 255 characters.)
sequence	No	String	A 36-byte serial number of a request message. For example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Message

Table 3-120 Response parameter

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 3-121**.

Table 3-121 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
key_description	Yes	String	Description of a CMK

Example

The following is an example about how to modify a CMK whose alias ID is **bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e** and description is **test**.

- Example request

```
{
    "key_id": "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "key_description": "test"
}
```

- Example response

```
{
    "key_info": {
        "key_id": "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
        "key_description": "test"
     }
}
```

or

```
{
    "error": {
        "error_code": "KMS.XXXX",
        "error_msg": "XXX"
    }
}
```

Status Codes

Table 3-122 lists the normal status code returned by the response.

Table 3-122 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

Exception status code. For details, see **Status Codes**.

3.2 Cloud Secret Management Service

3.2.1 Creating a Secret

Function

Create a secret and store the secret value in the initial secret version.

Secret values are encrypted and stored in secret versions. A version can have multiple statuses. Versions without any statuses are regarded as deprecated versions and can be automatically deleted by CSMS.

The initial version is marked by the **SYSCURRENT** status tag.

Constraints

You can use a symmetric customer master key (CMK) to encrypt a secret. If the **kms_key_id** parameter is not specified, the default master key **csms/default** will be used to encrypt secrets. The default key is automatically created by CSMS.

To use a user-defined key to encrypt secrets, you need to have the kms:dek:create permission for the key.

URI

POST /v1/{project_id}/secrets

Table 3-123 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameter

Table 3-124 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-125 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Secret name. Constraint: The value can contain 1 to 64 characters and must match the regular expression **^[a-zA-Z0-9._-]{1,64}$**.
kms_key_id	No	String	ID of the KMS CMK used to encrypt secrets. If this parameter is not specified, the default master key **csms/default** will be used. The default key is automatically created by CSMS.
description	No	String	Description of a secret. Constraints: The value contains 2048 bytes.
secret_binary	No	String	Plaintext of a binary secret in Base64 format. CSMS encrypts it and stores it in the initial version of the secret. Type: binary data object in Base64 format Constraints: You must configure one and only one of **secret_binary** and **secret_string**. The maximum size is 32 KB.

Parameter	Mandatory	Type	Description
secret_string	No	String	Plaintext of a binary secret in text format. CSMS encrypts it and stores it in the initial version of the secret. Constraints: You must configure one and only one of **secret_binary** and **secret_string**. The maximum size is 32 KB.

Response Parameters

Status code: 200

Table 3-126 Response body parameter

Parameter	Type	Description
secret	**Secret** object	Secret

Table 3-127 Secret parameters

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when a secret was last updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).

Parameter	Type	Description
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970). If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Status code: 400

Table 3-128 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-129 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-130 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-131 Response body parameters

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error description

Status code: 500

Table 3-132 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-133 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-134 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

Create a secret named **demo**. Encrypt the value of secret **this is a demo secret string** using the KMS key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
{
  "name" : "demo",
  "kms_key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "secret_string" : "this is a demo secret string"
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.2 Querying the Secret List

Function

Query all the secrets created by the user in the current project.

Constraints

The information returned via this API is the metadata of the secret and does not contain the secret value.

URI

GET /v1/{project_id}/secrets

Table 3-135 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 3-136 Query parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of results returned on each page Default value: 50
marker	No	String	Starting secret name of pagination query. If the parameter is left blank, only the secrets on the first page are queried.

Request Parameter

Table 3-137 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-138 Response body parameters

Parameter	Type	Description
secrets	Array of **Secret** objects	Secret list
page_info	**PageInfo** object	Pagination information

Table 3-139 Secret parameters

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when a secret was last updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970). If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Table 3-140 PageInfo

Parameter	Type	Description
next_marker	String	Query address of the next page (secret name at the end of the current page and the start of the next page)
previous_marker	String	Secret name at the start of the current page and the end of the last page
current_count	Integer	Number of records returned on the current page

Status code: 400

Table 3-141 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-142 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-143 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-144 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-145 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-146 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-147 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "secrets" : [ {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "secret-name-test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  } ],
  "page_info" : {
    "next_marker" : "secret-name-test",
    "previous_marker" : "secret-name-test",
    "current_count" : 1
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.

Status Code	Description
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.3 Querying a Secret

Function

Query a specified secret.

Constraints

The information returned via this API is the metadata of the secret and does not contain the secret value.

URI

GET /v1/{project_id}/secrets/{secret_name}

Table 3-148 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Request Parameter

Table 3-149 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-150 Response body parameter

Parameter	Type	Description
secret	**Secret** object	Secret

Table 3-151 Secret parameters

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when a secret was last updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).

Parameter	Type	Description
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970). If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Status code: 400

Table 3-152 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-153 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-154 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-155 Response body parameters

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error description

Status code: 500

Table 3-156 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-157 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-158 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.4 Updating a Secret

Function

Update the metadata of a specified secret.

Constraints

This API can be used to modify only the secret metadata, not the secret value.

URI

PUT /v1/{project_id}/secrets/{secret_name}

Table 3-159 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Parameter	Mandatory	Type	Description
secret_name	Yes	String	Secret name

Request Parameter

Table 3-160 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-161 Request body parameters

Parameter	Mandatory	Type	Description
kms_key_id	No	String	ID of the KMS CMK used to encrypt secret values. The CMK of a secret can be updated. Secret versions after the update use the new CMK for encryption. Secret versions before the update use the old CMK for decryption.
description	No	String	Description of a secret Constraints: The value contains 2048 bytes.

Response Parameters

Status code: 200

Table 3-162 Response body parameter

Parameter	Type	Description
secret	**Secret** object	Secret

Table 3-163 Secret parameters

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when a secret was last updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970). If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Status code: 400

Table 3-164 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-165 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-166 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-167 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-168 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-169 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-170 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

Update the ID of the secret KMS key to **test** and description to **update description**.

```
{
  "kms_key_id" : "test",
  "description" : "update description"
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.

Status Code	Description
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.5 Deleting a Secret Immediately

Function

Delete a specified secret immediately. The deleted secret cannot be restored.

Constraints

Secrets deleted via this API cannot be restored.

URI

DELETE /v1/{project_id}/secrets/{secret_name}

Table 3-171 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Request Parameter

Table 3-172 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 400

Table 3-173 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-174 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-175 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-176 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-177 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-178 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-179 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

None

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.6 Restoring a Secret

Function

Restore a secret by uploading the secret backup file.

Constraints

The information returned via this API is the metadata of the secret and does not contain the secret value.

URI

POST /v1/{project_id}/secrets/restore

Table 3-180 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 3-181 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Table 3-182 Request body parameter

Parameter	Mandatory	Type	Description
secret_blob	Yes	String	Backup file of a secret. The file contains information about all the versions of the secret. The backup file is encrypted and encoded, and cannot be directly read.

Response Parameters

Status code: 200

Table 3-183 Response body parameter

Parameter	Type	Description
secret	**Secret** object	Secret

Table 3-184 Secret

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp which indicates how many seconds it has been since January 1, 1970.
update_time	Long	Time when a secret was last updated. The value is a timestamp which indicates how many seconds it has been since January 1, 1970
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The value is a timestamp which indicates how many seconds it has been since January 1, 1970 If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Status code: 400

Table 3-185 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-186 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-187 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-188 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-189 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-190 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-191 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

Upload the secret backup file.

```
{
  "secret_blob" :
")CloudSecretManagementBackupV1.cloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTU
wNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKz
NaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-
Cbmf6qpJF61kXKHX4TBe6-
oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI
8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-
Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--
jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-
DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxU
XpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-
fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7Gug
cNi7jTKuvxu4l2Q2v7wnk"
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.7 Downloading a Secret Backup

Function

Download the backup file of a specified secret.

Constraints

This API returns a string indicating the secret backup file. The content is encrypted and cannot be read.

URI

POST /v1/{project_id}/secrets/{secret_name}/backup

Table 3-192 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Request Parameter

Table 3-193 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. The token can be obtained by calling the IAM API. (The token is the value of **X-Subject-Token** in the response header.)

Response Parameters

Status code: 200

Table 3-194 Response body parameters

Parameter	Type	Description
secret_blob	String	Backup file of a secret. The file contains information about all the versions of the secret. The backup file is encrypted and encoded, and cannot be directly read.

Status code: 400

Table 3-195 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-196 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-197 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-198 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-199 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-200 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-201 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "secret_blob" :
")CloudSecretManagementBackupV1.cloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTU
wNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKz
NaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-
Cbmf6qpJF61kXKHX4TBe6-
oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI
8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-
Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--
jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-
DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxU
XpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-
fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7Gug
cNi7jTKuvxu4l2Q2v7wnk"
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.8 Creating a Scheduled Secret Deletion Task

Function

Create a scheduled task to delete a secret in 7 to 30 days.

Constraints

If a secret is in **Pending deletion** state, its metadata cannot be updated and its value cannot be viewed.

URI

POST /v1/{project_id}/secrets/{secret_name}/scheduled-deleted-tasks/create

Table 3-202 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Request Parameter

Table 3-203 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-204 Request body parameter

Parameter	Mandatory	Type	Description
recovery_window_in_days	Yes	Integer	Create a scheduled secret deletion task and specify the waiting period before deletion. Constraints: The waiting period can be 7 to 30 days. Default value: 30

Response Parameters

Status code: 200

Table 3-205 Response body parameter

Parameter	Type	Description
secret	**Secret** object	Secret

Table 3-206 Secret

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when a secret was last updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970). If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Status code: 400

Table 3-207 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-208 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-209 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-210 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-211 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-212 Response body parameters

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error description

Status code: 504

Table 3-213 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

Create a scheduled secret deletion task and delete the secret 15 days later.

```
{
  "recovery_window_in_days" : 15
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.

Status Code	Description
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.9 Canceling a Scheduled Secret Deletion Task

Function

Cancel the scheduled deletion task of a secret. The secret will be available.

Constraints

This API can be used only if a secret is in the **Pending deletion** state.

URI

POST /v1/{project_id}/secrets/{secret_name}/scheduled-deleted-tasks/cancel

Table 3-214 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Request Parameter

Table 3-215 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-216 Response body parameter

Parameter	Type	Description
secret	**Secret** object	Secret

Table 3-217 Secret

Parameter	Type	Description
id	String	Secret ID
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when a secret was last updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970). If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Status code: 400

Table 3-218 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-219 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-220 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-221 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-222 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-223 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-224 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.

Status Code	Description
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.10 Creating a Secret Version

Function

Create a new version of a secret to encrypt and keep the new value of the secret. By default, The latest secret version in **SYSCURRENT** state. The previous version is in the **SYSPREVIOUS** state. You can configure the **VersionStage** to overwrite the default settings.

Constraints

- The CSMS console only uses the **secret_string** field. To add a binary secret to the **secret_binary** field, must use an SDK or API.
- A secret can have up to 20 versions.
- You can only add versions to enabled secrets.
- Secret versions are numbered v1, v2, v3, and so on based on their creation time.

URI

POST /v1/{project_id}/secrets/{secret_name}/versions

Table 3-225 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Request Parameters

Table 3-226 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-227 Request body parameters

Parameter	Mandatory	Type	Description
secret_binary	No	String	Value of a new secret. It will be encrypted and stored in the initial version of the secret. Type: binary data object in Base64 format Constraints: You must configure one and only one of **secret_binary** and **secret_string**. The maximum size is 32 KB.
secret_string	No	String	Value of a new secret. It will be encrypted and stored in the initial version of the secret. Constraints: You must configure one and only one of **secret_binary** and **secret_string**. The maximum size is 32 KB.
version_stages	No	Array of strings	Status of a new secret version. If this parameter is not specified, the default version **SYSCURRENT** will be used. Constraint: The array can contain 1 to 12 items. The stage length is 1 to 64 bytes.

Response Parameters

Status code: 200

Table 3-228 Response body parameter

Parameter	Type	Description
version_metadata	**VersionMetadata** object	Status of a secret version.

Table 3-229 VersionMetadata

Parameter	Type	Description
id	String	ID of a secret version. A secret cannot have duplicate version IDs.
create_time	Long	Secret version creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
kms_key_id	String	ID of the KMS CMK used to encrypt secret values.
secret_name	String	Secret name
version_stages	Array of strings	Status of a secret version. A status tag can be used for only one version of each secret. For example, if you add the status tag used by version A to version B, the tag will be moved from version A to version B. If the **version_stage** parameter is not specified, the status of the latest version will be **SYSCURRENT** by default.

Status code: 400

Table 3-230 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-231 Response body parameters

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error description

Status code: 403

Table 3-232 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-233 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-234 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-235 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-236 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

Create a secret version. The secret value is **secret_string**.

```
{
  "secret_string" : "secret_string"
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "version_metadata" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "create_time" : 1581507580000,
    "secret_name" : "secret-name-demo",
    "version_stages" : [ "pending", "used" ]
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.11 Querying the Secret Version List

Function

Query the version list of a specific secret.

Constraints

The information returned via this API is the metadata of the secret version and does not contain the secret value.

URI

GET /v1/{project_id}/secrets/{secret_name}/versions

Table 3-237 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name

Table 3-238 Query parameters

Parameter	Mandatory	Type	Description
marker	No	String	Version number of the last item in the previous page.
limit	No	Integer	Number of items displayed per page. The default value is **50**.

Request Parameter

Table 3-239 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-240 Response body parameters

Parameter	Type	Description
version_metadatas	Array of **VersionMetadata** objects	Version metadata
page_info	**PageInfo** object	Pagination information

Table 3-241 VersionMetadata

Parameter	Type	Description
id	String	ID of a secret version. A secret cannot have duplicate version IDs.
create_time	Long	Secret creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
kms_key_id	String	ID of the KMS CMK used to encrypt secret values.
secret_name	String	Secret name
version_stages	Array of strings	Status of a secret version. A status tag can be used for only one version of each secret. For example, if you add the status tag used by version A to version B, the tag will be moved from version A to version B. If the **version_stage** parameter is not specified, the status of the latest version will be **SYSCURRENT** by default.

Table 3-242 PageInfo

Parameter	Type	Description
next_marker	String	Query address of the next page (secret name at the end of the current page and the start of the next page)
previous_marker	String	Secret name at the start of the current page and the end of the last page

Parameter	Type	Description
current_count	Integer	Number of records returned on the current page

Status code: 400

Table 3-243 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-244 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-245 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-246 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-247 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-248 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-249 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "version_metadatas" : [ {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "create_time" : 1581507580000,
    "secret_name" : "secret-name-demo",
    "version_stages" : [ "pending", "used" ]
  } ],
  "page_info" : {
    "next_marker" : "v10",
    "previous_marker" : "v1",
    "current_count" : 10
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.12 Querying the Secret Version and Value

Function

Query a specified secret version and the plaintext secret value in the version. Only enabled secrets can be queried. The value of the latest secret version can be obtained via **/v1/{project_id}/secrets/{secret_name}/versions/latest**. (Set the **{version_id}** in the URL of the current API to **latest**).

URI

GET /v1/{project_id}/secrets/{secret_name}/versions/{version_id}

Table 3-250 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name
version_id	Yes	String	Secret version ID

Request Parameter

Table 3-251 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-252 Response body parameter

Parameter	Type	Description
version	**Version** object	Secret version

Table 3-253 Version

Parameter	Type	Description
version_metadata	**VersionMetadata** object	Status of a secret version
secret_binary	String	Plaintext of a binary secret in Base64 format. CSMS encrypts it and stores it in the initial version of the secret. Type: binary data object in Base64 format
secret_string	String	Plaintext of a binary secret in text format. CSMS encrypts it and stores it in the initial version of the secret.

Table 3-254 VersionMetadata

Parameter	Type	Description
id	String	ID of a secret version. A secret cannot have duplicate version IDs.

Parameter	Type	Description
create_time	Long	Secret version creation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
kms_key_id	String	ID of the KMS CMK used to encrypt secret values.
secret_name	String	Secret name
version_stages	Array of strings	Status of a secret version. A status tag can be used for only one version of each secret. For example, if you add the status tag used by version A to version B, the tag will be moved from version A to version B. If the **version_stage** parameter is not specified, the status of the latest version will be **SYSCURRENT** by default.

Status code: 400

Table 3-255 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-256 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-257 Response body parameters

Parameter	Type	Description
error_code	String	Error code

Parameter	Type	Description
error_msg	String	Error description

Status code: 404

Table 3-258 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-259 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-260 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-261 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "version" : {
    "version_metadata" : {
      "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
      "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
      "create_time" : 1581507580000,
      "secret_name" : "secret-name-demo",
      "version_stages" : [ "pending", "used" ]
    },
    "secret_binary" : "secret_binary",
    "secret_string" : "secret_string"
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.13 Updating the Version Status of a Secret

Function

Update the version status of a secret.

Constraints

- A status tag can be used for only one version of each secret. For example, if you add the status tag used by version A to version B, the tag will be moved from version A to version B. Versions without any statuses are regarded as deprecated versions and can be automatically deleted by CSMS.
- A secret can have up to 12 version statuses. A status can be used for only one version. **SYSCURRENT** and **SYSPREVIOUS** are the preconfigured secret statuses of a service.

URI

PUT /v1/{project_id}/secrets/{secret_name}/stages/{stage_name}

Table 3-262 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name
stage_name	Yes	String	Name of a secret version status which matches the regular expression **'^[a-zA-Z0-9._-]{1,64}$'**.

Request Parameter

Table 3-263 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-264 Request body parameter

Parameter	Mandatory	Type	Description
version_id	Yes	String	Secret version ID

Response Parameters

Status code: 200

Table 3-265 Response body parameter

Parameter	Type	Description
stage	Stage object	Secret status

Table 3-266 Stage

Parameter	Type	Description
name	String	Name of a secret version status. Constraint: 1 to 64 characters long
update_time	Long	Secret version update timestamp. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
secret_name	String	Secret name
version_id	String	Secret version ID

Status code: 400

Table 3-267 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-268 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-269 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-270 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-271 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-272 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-273 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

Update the version status of a secret. The version is **version_id**.

```
{
  "version_id" : "version_id"
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "stage" : {
    "name" : "name",
    "version_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "update_time" : 1581507580000,
    "secret_name" : "secret-name-demo"
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.14 Querying the Status of a Secret Version

Function

Query the status of a specified secret version.

URI

GET /v1/{project_id}/secrets/{secret_name}/stages/{stage_name}

Table 3-274 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name
stage_name	Yes	String	Name of a secret version status.

Request Parameter

Table 3-275 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-276 Response body parameter

Parameter	Type	Description
stage	**Stage** object	Secret status

Table 3-277 Stage

Parameter	Type	Description
name	String	Name of a secret version status. Constraint: 1 to 64 characters long
update_time	Long	Secret version update timestamp. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).

Parameter	Type	Description
secret_name	String	Secret name
version_id	String	Secret version ID

Status code: 400

Table 3-278 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-279 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-280 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-281 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-282 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-283 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-284 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "stage" : {
    "name" : "name",
    "version_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "update_time" : 1581507580000,
    "secret_name" : "secret-name-demo"
  }
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.15 Deleting the Version Status of a Secret

Function

Delete the status of a specified secret version.

Constraints

The **SYSCURRENT** and **SYSPREVIOUS** are preconfigured statuses and cannot be deleted.

URI

DELETE /v1/{project_id}/secrets/{secret_name}/stages/{stage_name}

Table 3-285 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret ID
stage_name	Yes	String	Name of a secret version status.

Request Parameter

Table 3-286 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 400

Table 3-287 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 401

Table 3-288 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 403

Table 3-289 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 404

Table 3-290 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 500

Table 3-291 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 502

Table 3-292 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Status code: 504

Table 3-293 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

None

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error code

For details, see **Error Codes**.

3.2.16 Querying a Secret Instance

Function

Query a secret instance. Filter user secrets by tag, and return a secret list.

URI

POST /v1/{project_id}/csms/{resource_instances}/action

Table 3-294 URI parameters

Parameter	Mandatory	Type	Description
resource_instances	Yes	String	Resource instance. The value is **resource_instances**.
project_id	Yes	String	Project ID

Request Parameter

Table 3-295 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-296 Request body parameter

Parameter	Mandatory	Type	Description
limit	No	String	Number of records in a query. If **action** is set to **count**, do not set this parameter. If **action** is set to **filter**, the default value of this parameter is **10**. The value ranges from **1** to **1,000**.
offset	No	String	If **action** is set to **count**, do not specify this parameter.
action	No	String	Operation type. Possible values are as follows: ● **filter**: Filter records. ● **count**: Count all the records.
tags	No	Array of Tag objects	Tag list, which is the value pairs of **key** and **value**. ● **key**: Tag key. A secret can contain up to 20 keys. This value cannot be left blank or repeated. The value of a **key** must be unique and contain up to 36 characters. Each pair contains one **key** and one **value**. ● **value**: Tag value. There can be multiple values and each value can contain up to 43 characters.

Parameter	Mandatory	Type	Description
matches	No	Array of **TagItem** objects	Search field. ● **key** is an exact match field. Currently, its value can only be **resource_name**. ● **value** is a fuzzy match field. It can contain a maximum of 255 characters. If this parameter is not specified, an empty value will be returned.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**.

Table 3-297 Tag

Parameter	Mandatory	Type	Description
key	No	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.
values	No	Array of strings	Tag value set

Table 3-298 TagItem

Parameter	Mandatory	Type	Description
key	No	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.

Parameter	Mandatory	Type	Description
value	No	String	Value. Each value can contain up to 43 Unicode characters and can be an empty string. It cannot contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.

Response Parameters

Status code: 200

Table 3-299 Response body parameters

Parameter	Type	Description
resources	Array of **ActionResources** objects	Resource instance list
total_count	Integer	Total number of resources.

Table 3-300 ActionResources

Parameter	Type	Description
resource_id	String	Resource ID
resource_detail	**Secret** object	Secret
resource_name	String	Resource name. This parameter is an empty string by default.
tags	Array of **TagItem** objects	Tag list. If there is no tag in the list, an empty array is returned.
sys_tags	Array of **TagItem** objects	Tag list. If there is no tag in the list, an empty array is returned.

Table 3-301 Secret

Parameter	Type	Description
id	String	Secret ID

Parameter	Type	Description
name	String	Secret name
state	String	Secret status. Possible values are as follows: **ENABLED** **DISABLED** **PENDING_DELETE** **FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt secret values
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp which indicates how many seconds it has been since January 1, 1970.
update_time	Long	Time when a secret was last updated. The value is a timestamp which indicates how many seconds it has been since January 1, 1970
scheduled_delete_time	Long	Time when a secret will be deleted as scheduled. The value is a timestamp which indicates how many seconds it has been since January 1, 1970 If a secret is not in **Pending deletion** state, the value of this parameter is **null**.

Table 3-302 TagItem

Parameter	Type	Description
key	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.
value	String	Value. Each value can contain up to 43 Unicode characters and can be an empty string. It cannot contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.

Status code: 400

Table 3-303 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-304 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-305 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-306 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-307 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-308 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-309 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-310 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-311 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-312 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-313 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-314 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-315 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-316 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

Filter user secrets based on the tag whose **key** is **key1** and **value** is **val1**, and return the secret list.

```
{
  "action" : "filter",
  "tags" : [ {
    "key" : "key1",
    "values" : [ "val1" ]
  } ]
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "total_count" : 1,
  "resources" : [ {
    "resource_id" : "2d1152f2-290d-4756-a1d2-e12c14992416"
  }, {
    "resource_detail" : {
      "id" : "2d1152f2-290d-4756-a1d2-e12c14992416",
      "name" : "example_name",
      "state" : "ENABLED",
      "description" : "",
      "kms_key_id" : "1213d410-ass1-1254-1a2d-3cca2sa2w554",
      "create_time" : 1581507580000,
      "update_time" : 1581507580000,
      "scheduled_delete_time" : 1581507580000
    }
  }, {
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    }, {
      "key" : "key2",
      "value" : "value2"
    } ]
  }, {
    "sys_tags" : null
  }, {
    "resource_name" : "example_name"
  } ]
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.17 Adding or Deleting Secret Tags in Batches

Function

Add or delete secret tags in batches.

URI

POST /v1/{project_id}/csms/{secret_id}/tags/action

Table 3-317 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_id	Yes	String	Secret ID

Request Parameter

Table 3-318 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-319 Request body parameter

Parameter	Mandatory	Type	Description
tags	No	Array of **TagItem** objects	Tag list, which is the value pairs of **key** and **value**.
action	No	String	Operation. The value can be **create** or **delete**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**.

Table 3-320 TagItem

Parameter	Mandatory	Type	Description
key	No	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.
value	No	String	Value. Each value can contain up to 43 Unicode characters and can be an empty string. It cannot contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.

Response Parameters

Status code: 400

Table 3-321 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-322 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-323 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-324 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-325 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-326 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-327 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-328 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-329 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-330 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-331 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-332 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-333 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-334 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

Add multiple secret tags. For tag 1, the **key** is **key1** and the **value** is **value1**. For tag 2, the **key** is **key2** and the **value** is **value2**.

```
{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

Example Response

None

Status Code

Status Code	Description
204	No Content
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.18 Querying a Secret Tag

Function

Query a secret tag.

URI

GET /v1/{project_id}/csms/{secret_id}/tags

Table 3-335 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_id	Yes	String	Secret ID

Request Parameter

Table 3-336 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-337 Response body parameters

Parameter	Type	Description
tags	Array of **TagItem** objects	Tag list, which is the value pairs of **key** and **value**. ● **key**: Tag key. A secret can contain up to 10 keys. This value cannot be left blank or repeated. The value of a **key** must be unique and contain up to 36 characters. ● **value**: Tag value. Each tag value can contain up to 43 characters. A search result matches all the values.

Parameter	Type	Description
sys_tags	Array of **TagItem** objects	Tag list, which is the value pairs of **key** and **value**. ● **key**: Tag key. A secret can contain up to 10 keys. This value cannot be left blank or repeated. The value of a **key** must be unique and contain up to 36 characters. ● **value**: Tag value. Each tag value can contain up to 43 characters. A search result matches all the values.

Table 3-338 TagItem

Parameter	Type	Description
key	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.
value	String	Value. Each value can contain up to 43 Unicode characters and can be an empty string. It cannot contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.

Status code: 400

Table 3-339 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-340 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-341 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-342 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-343 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-344 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-345 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-346 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-347 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-348 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-349 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-350 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-351 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-352 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ],
  "sys_tags" : null
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.

Status Code	Description
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.19 Adding a Secret Tag

Function

Add a secret tag.

URI

POST /v1/{project_id}/csms/{secret_id}/tags

Table 3-353 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_id	Yes	String	Secret ID

Request Parameter

Table 3-354 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Table 3-355 Request body parameter

Parameter	Mandatory	Type	Description
tag	No	**TagItem** object	-

Table 3-356 TagItem

Parameter	Mandatory	Type	Description
key	No	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.
value	No	String	Value. Each value can contain up to 43 Unicode characters and can be an empty string. It cannot contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.

Response Parameters

Status code: 400

Table 3-357 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-358 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-359 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-360 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-361 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-362 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-363 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-364 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-365 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-366 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-367 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-368 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-369 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-370 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

Add a secret tag whose **key** is **DEV** and **value** is **DEV1**.

```
{
  "tag" : {
    "key" : "DEV",
    "value" : "DEV1"
  }
}
```

Example Response

None

Status Code

Status Code	Description
204	No Content
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.20 Deleting a Secret Tag

Function

Delete a secret tag.

URI

DELETE /v1/{project_id}/csms/{secret_id}/tags/{key}

Table 3-371 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_id	Yes	String	Secret ID
key	Yes	String	Value of a tag key

Request Parameter

Table 3-372 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 400

Table 3-373 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-374 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-375 Response body parameter

Parameter	Type	Description
error	ErrorDetail object	-

Table 3-376 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-377 Response body parameter

Parameter	Type	Description
error	ErrorDetail object	-

Table 3-378 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-379 Response body parameter

Parameter	Type	Description
error	ErrorDetail object	-

Table 3-380 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-381 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-382 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-383 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-384 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-385 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-386 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

None

Example Response

None

Status Code

Status Code	Description
204	No Content
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.2.21 Querying Project Tags

Function

Query all secret tags of a user in a specified project.

URI

GET /v1/{project_id}/csms/tags

Table 3-387 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameter

Table 3-388 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API (value of **X-Subject-Token** in the response header).

Response Parameters

Status code: 200

Table 3-389 Response body parameter

Parameter	Type	Description
tags	Array of **Tag** objects	Tag list, which is the value pairs of **key** and **value**. ● **key**: Tag key. A secret can contain up to 10 keys. This value cannot be left blank or repeated. The value of a **key** must be unique and contain up to 36 characters. ● **value**: Tag value. Each tag value can contain up to 43 characters. A search result matches all the values.

Table 3-390 Tag

Parameter	Type	Description
key	String	Key. The value can contain up to 36 Unicode characters. This parameter cannot be left empty or contain non-printable characters, including ASCII(0-31), *, <, >, \, and =.
values	Array of strings	Tag value set

Status code: 400

Table 3-391 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-392 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 401

Table 3-393 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-394 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 403

Table 3-395 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-396 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 404

Table 3-397 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-398 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 500

Table 3-399 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-400 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 502

Table 3-401 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-402 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Status code: 504

Table 3-403 Response body parameter

Parameter	Type	Description
error	**ErrorDetail** object	-

Table 3-404 ErrorDetail

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error information

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "tags" : [ {
    "key" : "key1",
    "values" : [ "val1" ]
  }, {
    "key" : "key2",
    "values" : [ "val2" ]
  } ]
}
```

Status Code

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.
401	Username and password are required for the requested page.
403	Authentication failed.
404	The requested resource does not exist.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Code

For details, see **Error Codes**.

3.3 Key Pair Service

3.3.1 Creating and Importing an SSH Key Pair

Function

Create and import an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs

Table 3-405 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 3-406 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Table 3-407 Request body parameter

Parameter	Mandatory	Type	Description
keypair	Yes	**CreateKeypairAction** object	Parameter in the request body for creating a key pair

Table 3-408 CreateKeypairAction

Parameter	Mandatory	Type	Description
name	Yes	String	SSH key pair name. ● A new key pair cannot use the same name as an existing one. ● The name can contain at most 255 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
type	No	String	SSH key pair type. The value can be **ssh** or **x509**.

Parameter	Mandatory	Type	Description
public_key	No	String	String of a public key to be imported.
scope	No	String	Tenant-level or user-level. The value can be **domain** or **user**.
user_id	No	String	User to whom an SSH key pair belongs.
key_protection	No	**KeyProtection** object	SSH key pair private key management and protection.

Table 3-409 KeyProtection

Parameter	Mandatory	Type	Description
private_key	No	String	Private key of the imported SSH key pair.
encryption	Yes	**Encryption** object	Encryption method for the private key.

Table 3-410 Encryption

Parameter	Mandatory	Type	Description
type	Yes	String	The value can be **kms** or **default**. - **default**: default encryption mode, which applies to sites where the KMS service is not deployed. - **kms**: KMS encryption mode. If KMS is unavailable, set this parameter to **default**.
kms_key_name	No	String	KMS key name. - If **type** is set to **kms**, you must enter the KMS key name or ID.
kms_key_id	No	String	KMS key ID. - If **type** is set to **kms**, you must enter the KMS key name or ID.

Response Parameters

Status code: 200

Table 3-411 Response body parameter

Parameter	Type	Description
keypair	**CreateKeypairResp** object	SSH key pair details

Table 3-412 CreateKeypairResp

Parameter	Type	Description
name	String	SSH key pair name
type	String	SSH key pair type. The value can be **ssh** or **x509**.
public_key	String	Public key information about an SSH key pair
private_key	String	Private key information about an SSH key pair ● The information about the private key is contained in the response for creating an SSH key pair. ● The information about the private key is not contained in the response for importing an SSH key pair.
fingerprint	String	Fingerprint information about an SSH key pair
user_id	String	User to whom an SSH key pair belongs

Status code: 400

Table 3-413 Response body parameter

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

```
{
  "keypair" : {
    "name" : "demo2"
  }
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "keypair" : {
    "name" : "demo",
    "type" : "ssh",
    "public_key" : "ssh-rsa AAAAB3NzaC1yc2EAAAADAQAB...",
    "private_key" : "-----BEGIN RSA PRIVATE KEY-----...",
    "fingerprint" : "49:ef:73:2b:9b:7f:2e:0c:58:d3:e3:42:8e:28:04:3b",
    "user_id" : "e4f380899b1248918f3d37098dc63746"
  }
}
```

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

For details, see **Error Codes**.

3.3.2 Clearing a Private Key

Function

Delete the private key of an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/keypairs/{keypair_name}/private-key

Table 3-414 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Key pair name

Request Parameters

Table 3-415 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Response Parameters

Status code: 404

Table 3-416 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

None

Status Codes

Status Code	Description
200	Request succeeded.
404	The requested resource does not exist.

Error Codes

For details, see **Error Codes**.

3.3.3 Obtaining SSH Key Pairs

Function

Obtain SSH key pairs.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/keypairs

Table 3-417 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 3-418 Query parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of results returned on each page. Default value: **50**
marker	No	String	Resource ID of pagination query. If the parameter is left blank, only resources on the first page are queried.

Request Parameters

Table 3-419 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Response Parameters

Status code: 200

Table 3-420 Response body parameters

Parameter	Type	Description
keypairs	Array of **Keypairs** objects	SSH key pair list
page_info	**PageInfo** object	Pagination information

Table 3-421 Keypairs

Parameter	Type	Description
keypair	**Keypair** object	Key pair information

Table 3-422 Keypair

Parameter	Type	Description
name	String	SSH key pair name
type	String	Type of the SSH key pair. The value can be **ssh** or **x509**.
scope	String	Tenant-level or user-level. The value can be **domain** or **user**.
public_key	String	Public key information about an SSH key pair
fingerprint	String	Fingerprint information about an SSH key pair
is_key_protection	Boolean	Whether to host keys.

Parameter	Type	Description
frozen_state	String	Whether the key pair is frozen. - **0**: normal - **1**: frozen due to common causes - **2**: frozen by the public security bureau - **3**: frozen due to common causes and by the public security bureau - **4**: frozen due to violations - **5**: frozen due to common causes and violations - **6**: frozen by the public security bureau and due to violations - **7**: frozen by the public security bureau and due to common causes and violations - **8**: frozen due to lack of real-name authentication - **9**: frozen due to common causes and lack of real-name authentication - **10**: frozen by the public security bureau and due to lack of real-name authentication

Table 3-423 PageInfo

Parameter	Type	Description
next_marker	String	Address of the next page.
previous_marker	String	Address of the previous page.
current_count	Integer	Number of returned records.

Status code: 400

Table 3-424 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "keypairs" : [ {
    "keypair" : {
      "name" : "1hprr3TI",
      "type" : "ssh",
      "scope" : "user",
      "public_key" : "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABjV8GvwpSs.....",
      "fingerprint" : "65:ca:87:0a:16:86:59:ea:57:ea:18:37:58:e2:04:b0",
      "is_key_protection" : false,
      "frozen_state" : 0
    }
  } ],
  "page_info" : {
    "next_marker" : "KeyPair-dxxx",
    "previous_marker" : "KeyPair-xxxx",
    "current_count" : 49
  }
}
```

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

For details, see **Error Codes**.

3.3.4 Obtaining Details About an SSH Key Pair

Function

Obtain details about an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/keypairs/{keypair_name}

Table 3-425 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Key pair name

Request Parameters

Table 3-426 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Response Parameters

Status code: 200

Table 3-427 Response body parameter

Parameter	Type	Description
keypair	**KeypairDetail** object	Key pair details

Table 3-428 KeypairDetail

Parameter	Type	Description
name	String	SSH key pair name
id	Long	SSH key pair ID
type	String	SSH key pair type. The value can be **ssh** or **x509**.
scope	String	Tenant-level or user-level. The value can be **domain** or **user**.

Parameter	Type	Description
public_key	String	Public key information about an SSH key pair
fingerprint	String	Fingerprint information about an SSH key pair
is_key_protection	Boolean	Whether to host keys.
deleted	Boolean	Tag that indicates an SSH key pair is deleted
description	String	Description of an SSH key pair
user_id	String	User to whom an SSH key pair belongs
create_time	Long	Time when the SSH key pair was created. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
delete_time	Long	Time when the SSH key pair was deleted. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when the SSH key pair was updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
frozen_state	Integer	Whether the key pair is frozen. ● **0**: normal ● **1**: frozen due to common causes ● **2**: frozen by the public security bureau ● **3**: frozen due to common causes and by the public security bureau ● **4**: frozen due to violations ● **5**: frozen due to common causes and violations ● **6**: frozen by the public security bureau and due to violations ● **7**: frozen by the public security bureau and due to common causes and violations ● **8**: frozen due to lack of real-name authentication ● **9**: frozen due to common causes and lack of real-name authentication ● **10**: frozen by the public security bureau and due to lack of real-name authentication
key_id	String	Key ID
algorithm	String	Algorithm

Status code: 400

Table 3-429 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 200

Request succeeded.

```
{
  "keypair" : {
    "name" : "1hprr3TI",
    "id" : 116248,
    "type" : "ssh",
    "scope" : "user",
    "public_key" : "ssh-rsa AAAGenerated-by-Nova",
    "fingerprint" : "65:ca:87:0a:16:86:59:ea:57:ea:18:37:58:e2:04:b0",
    "is_key_protection" : false,
    "deleted" : false,
    "description" : "12345",
    "user_id" : "6c2a33b1b8474d0dbac0a24297127525",
    "create_time" : 1581507580000,
    "delete_time" : null,
    "update_time" : null,
    "frozen_state" : 0
  }
}
```

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

For details, see **Error Codes**.

3.3.5 Deleting an SSH Key Pair

Function

Delete an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/keypairs/{keypair_name}

Table 3-430 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Key pair name

Request Parameters

Table 3-431 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Response Parameters

Status code: 400

Table 3-432 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

None

Example Response

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
204	OK
400	Error response

Error Codes

For details, see **Error Codes**.

3.3.6 Updating the Description About an SSH Key Pair

Function

Update the description about an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/keypairs/{keypair_name}

Table 3-433 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Key pair name

Request Parameters

Table 3-434 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Table 3-435 Request body parameter

Parameter	Mandatory	Type	Description
keypair	Yes	**UpdateKeypairDescriptionReq** object	Message body for updating the SSH key pair description

Table 3-436 UpdateKeypairDescriptionReq

Parameter	Mandatory	Type	Description
description	Yes	String	Description

Response Parameters

Status code: 400

Table 3-437 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

```
{
  "keypair" : {
    "description" : "description"
  }
}
```

Example Response

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

For details, see **Error Codes**.

3.3.7 Importing a Private Key

Function

Import a private key to a specified key pair.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs/private-key/import

Table 3-438 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 3-439 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Table 3-440 Request body parameter

Parameter	Mandatory	Type	Description
keypair	Yes	**ImportPrivateKeyKeypairBean** object	Information about the key pair to be imported

Table 3-441 ImportPrivateKeyKeypairBean

Parameter	Mandatory	Type	Description
name	Yes	String	SSH key pair name. A new key pair cannot use the same name as an existing one. The name can contain at most 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
user_id	No	String	User to whom an SSH key pair belongs
key_protection	Yes	**ImportPrivateKeyProtection** object	SSH key pair private key management and protection.

Table 3-442 ImportPrivateKeyProtection

Parameter	Mandatory	Type	Description
private_key	Yes	String	Private key of the imported SSH key pair.
encryption	Yes	**Encryption** object	Encryption method for the private key.

Table 3-443 Encryption

Parameter	Mandatory	Type	Description
type	Yes	String	The value can be **kms** or **default**. - **default**: default encryption mode, which applies to sites where the KMS service is not deployed. - **kms**: KMS encryption mode. If KMS is unavailable, set this parameter to **default**.
kms_key_name	No	String	KMS key name. - If **type** is set to **kms**, you must enter the KMS key name or ID.
kms_key_id	No	String	KMS key ID. - If **type** is set to **kms**, you must enter the KMS key name or ID.

Response Parameters

Status code: 200

Table 3-444 Response body parameter

Parameter	Type	Description
keypair	**ImportPrivateKeyKeypairBean** object	N/A

Table 3-445 ImportPrivateKeyKeypairBean

Parameter	Type	Description
name	String	SSH key pair name. A new key pair cannot use the same name as an existing one. The name can contain at most 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
user_id	String	User to whom an SSH key pair belongs

Parameter	Type	Description
key_protection	**ImportPrivateKeyProtection** object	SSH key pair private key management and protection.

Table 3-446 ImportPrivateKeyProtection

Parameter	Type	Description
private_key	String	Private key of the imported SSH key pair.
encryption	**Encryption** object	Encryption method for the private key.

Table 3-447 Encryption

Parameter	Type	Description
type	String	The value can be **kms** or **default**. ● **default**: default encryption mode, which applies to sites where the KMS service is not deployed. ● **kms**: KMS encryption mode. If KMS is unavailable, set this parameter to **default**.
kms_key_name	String	KMS key name. ● If **type** is set to **kms**, you must enter the KMS key name or ID.
kms_key_id	String	KMS key ID. ● If **type** is set to **kms**, you must enter the KMS key name or ID.

Status code: 400

Table 3-448 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

```
{
  "keypair" : {
    "name" : "demo2",
    "key_protection" : {
      "private_key" : "-----BEGIN RSA PRIVATE KEY-----...",
      "encryption" : {
        "type" : "kms",
        "kms_key_name" : "kps/default"
      }
    }
  }
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "keypair" : {
    "name" : "demo2"
  }
}
```

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

For details, see **Error Codes**.

3.3.8 Exporting a Private Key

Function

Export the private key of a specified key pair.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs/private-key/export

Table 3-449 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 3-450 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It can be obtained by calling the IAM API. The value of **X-Subject-Token** in the response header is the user token.

Table 3-451 Request body parameter

Parameter	Mandatory	Type	Description
keypair	Yes	**KeypairBean** object	Information about the key pair whose private key is to be exported.

Table 3-452 KeypairBean

Parameter	Mandatory	Type	Description
name	Yes	String	SSH key pair name

Response Parameters

Status code: 200

Table 3-453 Response body parameters

Parameter	Type	Description
keypair	**ExportPrivateKeyKeypairBean** object	Information about the exported private key

Table 3-454 ExportPrivateKeyKeypairBean

Parameter	Type	Description
name	String	SSH key pair name
private_key	String	Private key of the SSH key pair

Status code: 400

Table 3-455 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_msg	String	Error description

Example Request

```
{
  "keypair" : {
    "name" : "demo2"
  }
}
```

Example Response

Status code: 200

Request succeeded.

```
{
  "keypair" : {
    "name" : "demo2",
    "private_key" : "-----BEGIN RSA PRIVATE KEY-----..."
  }
}
```

Status code: 400

Error response

```
{
  "error_code" : "KPS.XXX",
  "error_msg" : "XXX"
}
```

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

For details, see **Error Codes**.

4 Permissions Policies and Supported Actions

4.1 Introduction

This chapter describes fine-grained permissions management for your DEW. If your account does not need individual IAM users, you may skip over this chapter.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

You can grant permissions to users by using roles and policies. Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

An account has all of the permissions required to call all APIs, but IAM users must have the required permissions specifically assigned. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully.

Supported Actions

You can use system-defined policies provided in IAM, or create custom policies to supplement the system-defined policies, implementing refined access control. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permission: A statement in a policy that allows or denies certain operations.
- APIs: REST APIs that can be called in a custom policy.

- Actions: Added to a custom policy to control permissions for specific operations.
- Dependent actions: When assigning an action to users, you also need to assign dependent permissions for that action to take effect.
- IAM projects/Enterprise projects: the authorization scope of a custom policy. A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect in IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Project.

NOTE

√: supported; x: not supported

DEW supports the following actions that can be defined in custom policies:

Manage keys, such as creating keys, querying keys, and creating grants.

4.2 Encryption Key Management

Permission	API	Action	Dependent Permission	IAM Project (Project)	Enterprise Project (Enterprise Project)
Creating a CMK	POST /v1.0/{project_id}/kms/create-key	kms:cmk:create	-	√	√
Enabling a CMK	POST /v1.0/{project_id}/kms/enable-key	kms:cmk:enable	-	√	√
Disabling a CMK	POST /v1.0/{project_id}/kms/disable-key	kms:cmk:disable	-	√	√
Scheduling the deletion of a CMK	POST /v1.0/{project_id}/kms/schedule-key-deletion	kms:cmk:update	-	√	√
Canceling the scheduled deletion of a CMK	POST /v1.0/{project_id}/kms/cancel-key-deletion	kms:cmk:update	-	√	√

Permission	API	Action	Dependent Permission	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying the list of CMKs	POST /v1.0/{project_id}/kms/list-keys	kms:cmk:list	-	√	√
Queries the CMK information.	POST /v1.0/{project_id}/kms/describe-key	kms:cmk:get	-	√	√
Generating a random number	POST /v1.0/{project_id}/kms/gen-random	kms:cmk:generate	-	√	√
Creating a DEK	POST /v1.0/{project_id}/kms/create-datakey	kms:dek:create	-	√	√
Creating a plaintext-free DEK	POST /v1.0/{project_id}/kms/create-datakey-without-plaintext	kms:dek:create	-	√	√
Encrypting a DEK	POST /v1.0/{project_id}/kms/encrypt-datakey	kms:dek:crypto	-	√	√
Decrypting a DEK	POST /v1.0/{project_id}/kms/decrypt-datakey	kms:dek:crypto	-	√	√
Querying the number of instances	GET /v1.0/{project_id}/kms/user-instances	kms:cmk:getInstance	-	√	√
Querying the user quota	GET /v1.0/{project_id}/kms/user-quotas	kms:cmk:getQuota	-	√	√
Modifying the CMK alias	POST /v1.0/{project_id}/kms/update-key-alias	kms:cmk:update	-	√	√

Permission	API	Action	Dependent Permission	IAM Project (Project)	Enterprise Project (Enterprise Project)
Modifying the description of a CMK	POST /v1.0/{project_id}/kms/update-key-description	kms:cmk:update	-	√	√
Encrypting data	POST /v1.0/{project_id}/kms/encrypt-data	kms:cmk:crypto	-	√	√
Decrypting data	POST /v1.0/{project_id}/kms/decrypt-data	kms:cmk:crypto	-	√	√
Obtaining parameters for importing a key	POST /v1.0/{project_id}/kms/get-parameters-for-import	kms:cmk:getMaterial	-	√	√
Importing key material	POST /v1.0/{project_id}/kms/import-key-material	kms:cmk:importMaterial	-	√	√
Deleting key material	POST /v1.0/{project_id}/kms/delete-imported-key-material	kms:cmk:deleteMaterial	-	√	√
Querying key resource instances	POST /v1.0/{project_id}/kms/resource_instances/action	kms:cmkTag:listInstance	-	√	√
Querying tags of a key	GET /v1.0/{project_id}/kms/{key_id}/tags	kms:cmkTag:list	-	√	√
Querying the project tags	GET /v1.0/{project_id}/kms/tags	kms:cmkTag:list	-	√	√
Adding or deleting key tags in batches	POST /v1.0/{project_id}/kms/{key_id}/tags/action	kms:cmkTag:batch	-	√	√

Permission	API	Action	Dependent Permission	IAM Project (Project)	Enterprise Project (Enterprise Project)
Adding tags to a key	POST /v1.0/{project_id}/kms/{key_id}/tags	kms:cmkTag:create	-	√	√
Deleting tags of a key	POST /v1.0/{project_id}/kms/{ key_id }/tags/{key}	kms:cmkTag:delete	-	√	√

A Appendix

A.1 Status Codes

Status Code	Status	Description
200	OK	Request processed successfully.
400	Bad Request	The request parameter is incorrect.
403	Forbidden	The server understood the request, but is refusing to fulfill it.
404	Not Found	The requested resource does not exist or not found.
500	Internal Server Error	Internal service error.

A.2 Error Codes

Status Code	Error Code	Error Message	Description	Measure
400	KMS.0201	Invalid request URL.	Invalid request URL.	Enter a valid URL.
400	KMS.0202	Invalid JSON format of the request message.	Invalid JSON format of the request message.	Enter a valid message.
400	KMS.0203	Request message too long.	Request message too long.	Enter a valid message.

Status Code	Error Code	Error Message	Description	Measure
400	KMS.0204	Parameters missing in the request message.	Parameters missing in the request message.	Enter a valid message.
400	KMS.0205	Invalid key ID.	Invalid key ID.	Enter a valid key ID.
400	KMS.0206	Invalid sequence number.	Invalid sequence number.	Enter a valid sequence number.
400	KMS.0208	Invalid value of value encryption_context.	Invalid value of value encryption_context.	Enter a valid value of encryption_context.
400	KMS.0209	The key has been disabled.	The key has been disabled.	Enable the key.
400	KMS.0210	The key is in Scheduled deletion state and cannot be used.	The key is in **Pending deletion** state and cannot be used.	Enable the key.
400	KMS.0211	Cannot perform this operation on Default Master Keys.	Cannot perform this operation on default master keys.	Perform this operation on a common CMK.
400	KMS.0308	Invalid parameter.	Invalid parameter.	Enter a valid parameter.
400	KMS.0309	External keys required.	An external key is required.	Use an imported key.
400	KMS.0310	The key is not in Pending import state.	The key is not in Pending import state.	Ensure the key is in Pending import state.
400	KMS.0311	Failed to decrypt data using the RSA private key.	Failed to decrypt data using the RSA private key.	Ensure the input ciphertext is correct and try again, or contact customer service.
400	KMS.0312	External keys cannot be rotated.	External keys cannot be rotated.	Use a common CMK.

Status Code	Error Code	Error Message	Description	Measure
400	KMS.0313	Key rotation is not enabled.	Key rotation is not enabled.	Enable key rotation.
400	KMS.0401	Tag list cannot be empty.	The tag list cannot be empty.	Enter a valid parameter.
400	KMS.0402	Invalid match value.	Invalid match value.	Enter a valid parameter.
400	KMS.0403	Invalid match key.	Invalid match key.	Enter a valid parameter.
400	KMS.0404	Invalid action.	Invalid action.	Enter a valid parameter.
400	KMS.0405	Invalid tag value.	Invalid tag value.	Enter a valid parameter.
400	KMS.0406	Invalid tag key.	Invalid tag key.	Enter a valid parameter.
400	KMS.0407	Invalid tag list size.	Invalid tag list size.	Enter a valid parameter.
400	KMS.0408	Invalid resourceType.	Invalid **resourceType**.	Enter a valid parameter.
400	KMS.0409	Too many tags.	Too many tags.	Delete unnecessary tags and try again.
400	KMS.0410	Invalid tag value length.	Invalid tag value length.	Enter a valid parameter.
400	KMS.0411	Invalid tag key length.	Invalid tag key length.	Enter a valid parameter.
400	KMS.0412	Invalid tag list.	Invalid tag list.	Enter a valid parameter.
400	KMS.0413	Too many tag values.	Too many tag values.	Enter a valid parameter.
400	KMS.0415	Invalid matches.	Invalid matches.	Enter a valid parameter.
400	KMS.0417	Invalid offset.	Invalid offset.	Enter a valid parameter.
400	KMS.1101	Invalid key_alias.	Invalid key_alias.	Enter a valid parameter.

Status Code	Error Code	Error Message	Description	Measure
400	KMS.1102	Invalid realm.	Invalid realm.	Enter a valid parameter.
400	KMS.1103	Invalid key_description.	Invalid key_description.	Enter a valid parameter.
400	KMS.1104	Duplicate key aliases.	Duplicate key aliases.	Use another alias.
400	KMS.1105	Too many keys.	Too many keys.	Increase key quota or delete unnecessary keys.
400	KMS.1201	The key is not disabled.	The key is not disabled.	Disable the key.
400	KMS.1301	The key is not enabled.	The key is not enabled.	Enable the key.
400	KMS.1401	Set the pending deletion period between 7 to 1,096 days.	Set the pending deletion period between 7 to 1,096 days.	Enter a valid parameter.
400	KMS.1402	The key is already in Pending deletion state.	The key is already in **Pending deletion** state.	No further operation required.
400	KMS.1501	The key is not in Pending deletion state.	The key is not in **Pending deletion** state.	Schedule deletion the key.
400	KMS.1601	Invalid limit.	Invalid limit.	Enter a valid parameter.
400	KMS.1602	marker must be greater than or equals 0.	**marker** must be greater than or equals 0.	Enter a valid parameter.
400	KMS.1801	random_data_length must be 512 bits.	random_data_length must be 512 bits.	Enter a valid parameter.

Status Code	Error Code	Error Message	Description	Measure
400	KMS.1901	datakey_length must be in the range 8 bits to 8,192 bits.	datakey_length must be in the range 8 bits to 8,192 bits.	Enter a valid parameter.
400	KMS.2001	datakey_length must be 512 bits.	datakey_length must be 512 bits.	Enter a valid parameter.
400	KMS.2101	Invalid plain_text.	Invalid plain_text.	Enter a valid parameter.
400	KMS.2102	datakey_plain_length must be 64 bytes.	datakey_plain_length must be 64 bytes.	Enter a valid parameter.
400	KMS.2103	Failed to verify the DEK hash.	Failed to verify the DEK hash.	Check whether the DEK is valid.
400	KMS.2201	Invalid cipher_text.	invalid cipher_text.	Enter a valid parameter.
400	KMS.2202	datakey_cipher_length must be 64 bytes.	datakey_cipher_length must be 64 bytes.	Enter a valid parameter.
400	KMS.2203	Failed to verify the DEK hash.	Failed to verify the DEK hash.	Check whether the DEK is valid.
400	KMS.2601	Token expired.	Token expired.	Obtain a new token.
400	KMS.2602	Key expiration time must be later than the current time.	Key expiration time must be later than the current time.	Set a valid key expiration time.
400	KMS.2603	Key IDs in the imported key and token do not match.	Key IDs in the imported key and token do not match.	Ensure the key ID in the imported key matches that in the token.
400	KMS.2604	The external key plaintext length must be 32 bits.	The external key plaintext length must be 32 bits.	Enter a valid parameter.
400	KMS.2605	Token verification failed.	Token verification failed.	Obtain a new token.

Status Code	Error Code	Error Message	Description	Measure
400	KMS.2606	You are importing a deleted key again. The imported plaintext must be the same as the deleted key plaintext.	You are importing a deleted key again. The imported plaintext must be the same as the deleted key plaintext.	Ensure the plaintext of the imported key is the same as that of the deleted key.
400	KMS.2701	Key material is not in Enabled or Disabled state and cannot be deleted.	Key material is not in the **Enabled** or **Disabled** state and cannot be deleted.	Ensure that the key is in the **Enabled** or **Disabled** state.
403	KMS.0301	Invalid or null X-Auth-Token.	Invalid or null X-Auth-Token.	Obtain the token again and ensure the token string is complete.
403	KMS.0302	Invalid X-Auth-Token.	Invalid X-Auth-Token.	Obtain the token again and ensure the token string is complete.
403	KMS.0303	X-Auth-Token expired.	X-Auth-Token expired.	Obtain the token again and ensure the token string is complete.
403	KMS.0304	X-Auth-Token contains the OBT tag and cannot be used to access services.	X-Auth-Token contains the OBT tag and cannot be used to access services.	Obtain the token again and ensure the token string is complete.
403	KMS.0305	Invalid X-Auth-Token project name.	Invalid X-Auth-Token project name.	Obtain the token again and ensure the token string is complete.
403	KMS.0306	No access permissions.	The user has no permission to access the key.	Contact the KMS administrator to grant required permissions.

Status Code	Error Code	Error Message	Description	Measure
403	KMS.0307	No access permissions.	No access permissions.	Contact the administrator to grant required permissions.
500	KMS.0101	KMS error.	KMS error.	Try again.
500	KMS.0102	Abnormal KMS I/O.	Abnormal KMS I/O.	Try again.

A.3 Obtaining a Project ID

Scenario

A project ID is required for some URLs when an API is called. Obtain the required project ID using either of the following methods:

- **Obtaining a Project ID by Calling an API**
- **Obtaining a Project ID from the Console**

Obtaining a Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the 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**.

In the following example, **id** indicates the project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "xxxxxxxx",
            "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"
    }
}
```

Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following steps:

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

   On the **My Credential** page, view project IDs in the project list.

A.4 API Permissions

A.4.1 API Actions

API	API Function	Permission
POST /v1.0/{project_id}/kms/create-key	Creates a CMK.	kms:cmk:create
POST /v1.0/{project_id}/kms/enable-key	Enables a CMK.	kms:cmk:enable
POST /v1.0/{project_id}/kms/disable-key	Disables a CMK.	kms:cmk:disable
POST /v1.0/{project_id}/kms/schedule-key-deletion	Schedules the deletion of a CMK.	kms:cmk:update
POST /v1.0/{project_id}/kms/cancel-key-deletion	Cancels the scheduled deletion of a CMK.	kms:cmk:update
POST /v1.0/{project_id}/kms/list-keys	Queries the list of CMKs.	kms:cmk:list
POST /v1.0/{project_id}/kms/describe-key	Queries the CMK information.	kms:cmk:get
POST /v1.0/{project_id}/kms/gen-random	Generates a random number.	kms:cmk:generate
POST /v1.0/{project_id}/kms/create-datakey	Creates a DEK.	kms:dek:create
POST /v1.0/{project_id}/kms/create-datakey-without-plaintext	Creates a plaintext-free DEK.	kms:dek:create
POST /v1.0/{project_id}/kms/encrypt-datakey	Encrypts a DEK.	kms:dek:crypto
POST /v1.0/{project_id}/kms/decrypt-datakey	Decrypts a DEK.	kms:dek:crypto

API	API Function	Permission
GET /v1.0/{project_id}/kms/user-instances	Queries the number of instances.	kms:cmk:getInstance
GET /v1.0/{project_id}/kms/user-quotas	Queries the user quota.	kms:cmk:getQuota
POST /v1.0/{project_id}/kms/update-key-alias	Modifies the CMK alias.	kms:cmk:update
POST /v1.0/{project_id}/kms/update-key-description	Modifies the description of a CMK.	kms:cmk:update
POST /v1.0/{project_id}/kms/encrypt-data	Encrypts data.	kms:cmk:crypto
POST /v1.0/{project_id}/kms/decrypt-data	Decrypts data.	kms:cmk:crypto
POST /v1.0/{project_id}/kms/get-parameters-for-import	Obtains parameters for importing a key.	kms:cmk:getMaterial
POST /v1.0/{project_id}/kms/import-key-material	Imports key material.	kms:cmk:importMaterial
POST /v1.0/{project_id}/kms/delete-imported-key-material	Deletes key material.	kms:cmk:deleteMaterial
POST /v1.0/{project_id}/kms/resource_instances/action	Queries key resource instances.	kms:cmkTag:listInstance
GET /v1.0/{project_id}/kms/{key_id}/tags	Queries tags of a key.	kms:cmkTag:list
GET /v1.0/{project_id}/kms/tags	Queries the project tags.	kms:cmkTag:list
POST /v1.0/{project_id}/kms/{key_id}/tags/action	Adds or deletes key tags in batches.	kms:cmkTag:batch
POST /v1.0/{project_id}/kms/{key_id}/tags	Adds tags to a key.	kms:cmkTag:create
POST /v1.0/{project_id}/kms/{ key_id }/tags/{key}	Deletes tags of a key.	kms:cmkTag:delete