Key Management Service

API Reference（ME-AbuDhabi）

Issue 03

Date 2022-11-28

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Before You Start 1
1.1 Overview 1
1.2 API Calling 1
1.3 Endpoints 1
1.4 Constraints 1
1.5 Concepts 1
2 Calling APIs 3
2.1 Making an API Request 3
2.2 Authentication 6
2.3 Returned Values 7
3 API Overview 9
4 APIs 11
4.1 Creating a CMK 11
4.2 Enabling a CMK 14
4.3 Disabling a CMK 16
4.4 Scheduling the Deletion of a CMK 18
4.5 Canceling the Scheduled Deletion of a CMK 20
4.6 Querying the List of CMKs 22
4.7 Querying the Information About a CMK 26
4.8 Creating a Random Number 29
4.9 Creating a DEK 31
4.10 Creating a Plaintext-Free DEK 33
4.11 Encrypting a DEK 36
4.12 Decrypting a DEK 39
4.13 Querying the Number of Instances 41
4.14 Querying the Quota of a User 43
4.15 Changing the Alias of a CMK 45
4.16 Changing the Description of a CMK 47
4.17 Encrypting Data 49
4.18 Decrypting Data 52
4.19 Obtaining CMK Import Parameters 54
4.20 Importing CMK Material 57

4.21 Deleting CMK Material....59
4.22 Querying CMK Instances....61
4.23 Querying CMK Tags....65
4.24 Querying Project Tags....67
4.25 Adding or Deleting CMK Tags in Batches....69
4.26 Adding a CMK Tag....71
4.27 Deleting a CMK Tag....73
5 Permissions Policies and Supported Actions....76
5.1 Introduction....76
5.2 Encryption Key Management....77
A Appendix....81
A.1 Status Codes....81
A.2 Error Code....81
A.3 Obtaining a Project ID....87
A.4 API Permissions....88
A.4.1 Encryption Key Management....88
B Change History....90

1 Before You Start

1.1 Overview

Key Management Service (KMS) is a secure, reliable, and easy-to-use service for managing your keys on the cloud. It helps you easily create, manage, and protect keys.

You can use the APIs described in this document to perform operations on keys, such as creating, querying, and deleting keys. For details about all supported operations, see **API Overview**.

1.2 API Calling

KMS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS requests. For details about API calling, see **Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see **Regions and Endpoints**.

1.4 Constraints

For more constraints, see the descriptions of specific APIs.

1.5 Concepts

- Account

  An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used to perform routine management. For security

purposes, create IAM users under the account and grant them permissions for routine management.

- User

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

  The account name, 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.

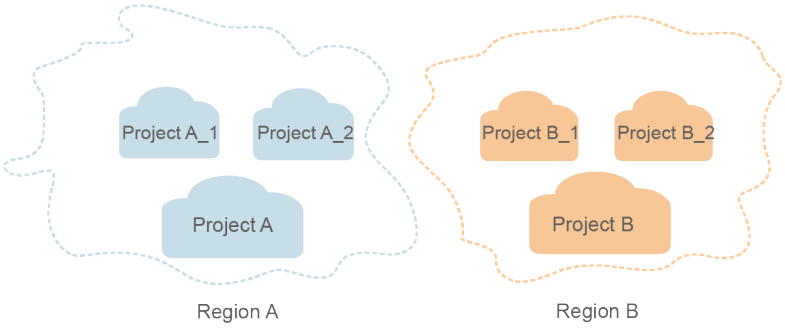
- Availability Zone (AZ)

  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 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**.

  For example, the endpoint of IAM in the **ae-ad-1** region is **iam.ae-ad-1.myhuaweicloud.com**.
- **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**:

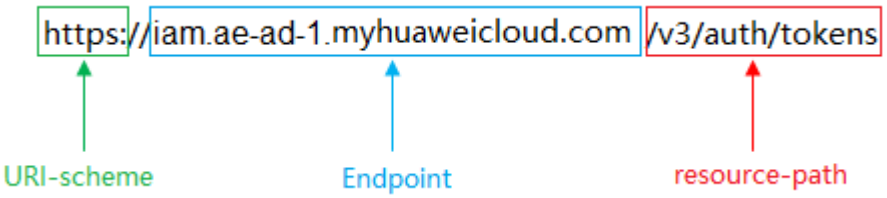
  Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **ae-ad-1** region, obtain the endpoint of IAM (**iam.ae-ad-1.myhuaweicloud.com**) for this region and the **resource-path**

(**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 2-1 Example URI



NOTE

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

Request Methods

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

- **GET**: requests the server to return specified resources.
- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

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

```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

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

Common request header fields are as follows:

- **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://iam.ae-ad-1.myhuaweicloud.com/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 ***xxxxxxxxxxxxxxxxx*** 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://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

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

```
    }
  }
}
```

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 **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:


```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

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

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

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

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or 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 Returned Values

Status Code

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

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

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 2-2 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 2-2 Header of the response to the request for obtaining a user token

(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 API Overview

You can use all the functions of Key Management Service (KMS) by calling its APIs.

Key Management APIs

API	Description
Creating a CMK	Creates a CMK.
Enabling a CMK	Enables a CMK. Only an enabled CMK can be used.
Disabling a CMK	Disables a CMK. A disabled CMK cannot be used.
Scheduling the Deletion of a CMK	Schedules the deletion of a specific key. The deletion can be scheduled 7 to 1096 days in advance. After a key is deleted, the data encrypted using the key cannot be decrypted.
Canceling the Scheduled Deletion of a CMK	Cancels a scheduled deletion of a key. Once the deletion is cancelled, the key can be used.
Querying the List of CMKs	Queries the list of all CMKs.
Querying the Information About a CMK	Queries details of a specified key.
Creating a Random Number	Generates a multiple of 8 that is 8 bits to 8192 bits long.
Creating a DEK	Creates a DEK. A returned result includes the plaintext and the ciphertext of a DEK.
Creating a Plaintext-Free DEK	Creates a plaintext-free DEK, that is, the returned result of this API includes only the ciphertext of the DEK.
Encrypting a DEK	Uses a specified CMK to encrypt a DEK.

API	Description
Decrypting a DEK	Uses a specified CMK to decrypt a DEK.
Querying the Number of Instances	Obtains the number of created CMKs, excluding the default master keys.
Querying the Quota of a User	Queries the total quota of CMKs available and the usage information, excluding the default master keys.
Changing the Alias of a CMK	Changes the alias of a CMK.
Changing the Description of a CMK	Changes the description of a CMK.
Encrypting Data	Uses a specified CMK to encrypt data.
Decrypting Data	Decrypts data.
Obtaining CMK Import Parameters	Obtains necessary parameters to import a key, including an import token and an encryption public key.
Importing CMK Material	Imports the key material of a specified key.
Querying CMK Instances	Uses the tag filtering function to query the detailed information of a CMK.
Querying CMK Tags	Queries tags of a CMK.
Querying Project Tags	Queries all tag sets of a project.
Adding or Deleting CMK Tags in Batches	Adds or deletes CMK tags in a batch.
Adding a CMK Tag	Adds a tag to a CMK.

4 APIs

4.1 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 4-1 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-2 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.)
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	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-3 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 4-4**.

Table 4-4 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
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 4-5 lists the normal status code returned by the response.

Table 4-5 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.2 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 4-6 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-7 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-8 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 4-9**.

Table 4-9 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.

Examples

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 4-10 lists the normal status code returned by the response.

Table 4-10 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.3 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 4-11 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-12 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

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

Responses

Table 4-13 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 4-14**.

Table 4-14 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.

Examples

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 4-15 lists the normal status code returned by the response.

Table 4-15 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.4 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 1096 days.

URI

- URI format

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

  Table 4-16 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-17 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** 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 **1096**.)
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-18 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.

Examples

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 4-19 lists the normal status code returned by the response.

Table 4-19 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.5 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 4-20 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-21 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-22 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.

Examples

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 4-23 lists the normal status code returned by the response.

Table 4-23 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.6 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 4-24 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-25 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**.
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}$**.

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

Responses

Table 4-26 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 4-31**.
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.

Parameter	Mandatory	Type	Description
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 4-27 lists the normal status code returned by the response.

Table 4-27 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.7 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 4-28 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-29 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-30 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 4-31**.

Table 4-31 key_info field description

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
domain_id	Yes	String	User domain ID
key_alias	Yes	String	Alias of a CMK
realm	Yes	String	Region where a CMK resides
key_description	Yes	String	Description of a CMK
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.

Parameter	Mandatory	Type	Description
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.
key_type	Yes	String	Type of a CMK
origin	Yes	String	Origin of a CMK. The default value is **kms**. The following values are enumerated:
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.

Examples

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 4-32 lists the normal status code returned by the response.

Table 4-32 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.8 Creating a Random Number

Function

This API generates a random number that is 8 bits to 8192 bits long.

URI

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

Table 4-33 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-34 Request parameters

Parameter	Mandatory	Type	Description
random_data_length	Yes	String	The value is a multiple of 8, in the range 8 to 8192.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-35 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 4-36 lists the normal status code returned by the response.

Table 4-36 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.9 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 4-37 Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-38 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK that matches the regular expression **^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

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"}
datakey_length	No	String	Bit length of a key The value is a multiple of 8, in the range 8 to 8192.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-39 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.

Examples

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 4-40 lists the normal status code returned by the response.

Table 4-40 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.10 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 4-41 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-42 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
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	Bit length of a key The value is a multiple of 8, in the range 8 to 8192.
key_spec	No	String	Bit length of the generated key Valid values: **AES_256** and **AES_128**. ● **AES_256**: a 256-bit symmetric key ● **AES_128**: a 128-bit symmetric key **NOTE** Set either **datakey_length** or **key_spec**. ● If neither of them is specified, a 256-bit key is generated by default. ● If both of them are specified, only **datakey_length** takes effect.

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

Responses

Table 4-43 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.

Examples

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":
"020098005CDC28E29EC3230AA42E8985FBABA095037D6474C64519C9B564AB28B15739C88E7E887500D1094973C2DC16353DB7ED3946C73339517AB1E983D521F9E9D700DC5D9C42F557EBF3F608E3CBBEE0BC68136EE7D2A49117E00332BAC4AE4ED805EB6068FA900C5A8019BFE2C2651BE3E130643034363662302D653732372D346439632D623335642D6638346262343734613337660000000F160727EBDB83400C21D80D713B49D3A2C37F24AE160E7BB3DAC025ADC0C45E3"
}
```

or

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

Status Codes

Table 4-44 lists the normal status code returned by the response.

Table 4-44 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.11 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 4-45 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-46 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

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"}
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 **Examples**.
datakey_plain_length	Yes	String	Number of bytes of a DEK in plaintext. The value range is 1 to 1024.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-47 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

Examples

In the following example, the 512-bit plaintext DEK (**7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6c**

cab48f8e5efe74e4b18ff0512525e527b10331100f357bf42125d8d5ced94f) 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 **7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4b18ff0512525e527b10331100f357bf42125d8d5ced94ffbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0a9ee0305fb9540f797**.

- 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 4-48 lists the normal status code returned by the response.

Table 4-48 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.12 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 4-49 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-50 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

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	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-51 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

Examples

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

020098005273E14E6E8E95F5463BECDC27E80AF820B9FC086CB47861899149F67CF07DAFF2810B7D27BDF19AB7632488E0926A48DB2FC85BEA905119411B46244C5E6B8036C60A0B0B4842FFE6994518E89C19B1C1D688D9043BCD6053EA7BA0652642CE59F2543C80669139F4F71ABB9BD9A24330643034363662302D653732372D346439632D623335642D663834626234373461333766000000000D34457984F9730D57F228C210FD22CA6017913964B21D4ECE45D81092BB9112E; length: **64** bits).

- Example request


```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "datakey_cipher_length": "64",
    "cipher_text": 
"020098005273E14E6E8E95F5463BECDC27E80AF820B9FC086CB47861899149F67CF07DAFF2810B7D27BDF19AB7632488E0926A48DB2FC85BEA905119411B46244C5E6B8036C60A0B0B4842FFE6994518E89C19B1C1D688D9043BCD6053EA7BA0652642CE59F2543C80669139F4F71ABB9BD9A24330643034363662302D653732372D346439632D623335642D663834626234373461333766000000000D34457984F9730D57F228C210FD22CA6017913964B21D4ECE45D81092BB9112E"
}
```


- Example response


```
{
    "data_key": 
"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
    "datakey_length": "64",
    "datakey_dgst": "F5A5FD42D16A20302798EF6ED309979B43003D2320D9F0E8EA9831A92759FB4B"
}
```


or


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


Status Codes

Table 4-52 lists the normal status code returned by the response.

Table 4-52 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.13 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 4-53 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

None

Responses

Table 4-54 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 4-55 lists the normal status code returned by the response.

Table 4-55 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.14 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 4-56 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

None

Responses

Table 4-57 Response parameters

Parameter	Mandatory	Type	Description
quotas	Yes	Object	Quota list. For details, see **Table 4-58**.

Table 4-58 quotas field description

Parameter	Mandatory	Type	Description
resources	Yes	Array of objects	Resource quota list. For details, see **Table 4-59**.

Table 4-59 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 4-60 lists the normal status code returned by the response.

Table 4-60 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.15 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 4-61 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-62 Request parameters

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

Responses

Table 4-63 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 4-64**.

Table 4-64 key_info field description

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

Examples

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 4-65 lists the normal status code returned by the response.

Table 4-65 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.16 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 4-66 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-67 Request parameters

Parameter	Type	Mandatory	Description
key_id	String	Yes	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
key_description	String	Yes	CMK description (The value ranges from 0 to 255 characters.)
sequence	String	No	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-68 Response parameters

Parameter	Mandatory	Type	Description
key_info	Yes	Array of objects	Information about keys. For details, see **Table 4-69**.

Table 4-69 key_info field description

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

Examples

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 4-70 lists the normal status code returned by the response.

Table 4-70 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.17 Encrypting Data

Function

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

URI

- URI format

  POST /v1.0/{project_id}/kms/encrypt-data

- Parameter description

Table 4-71 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-72 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
encryption_context	No	Object	Key-value pairs with a maximum length of 8192 characters. This parameter is used to record resource context information, excluding sensitive information, to ensure data integrity. If this parameter is specified during encryption, it is also required for decryption. Example: {"Key1":"Value1","Key2":"Value2"}
plain_text	Yes	String	Plaintext data which is 1 to 4096 bytes in length and matches the regular expression **^.{1,4096}$**. After being converted into a byte array, it is still 1 to 4096 bytes in length.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-73 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
cipher_text	Yes	String	Ciphertext data in Base64 format

Examples

The following example describes how to use a CMK (ID: **0d0466b0-e727-4d9c-b35d-f84bb474a37f**) to encrypt data (plaintext: **12345678**).

- Example request

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "plain_text": "12345678"
}
```

- Example response

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "cipher_text": "AgDoAG7EsEc2OHpQxz4gDFDH54CqwaelpTdEl
+RFPjbKn5klPTvOywYIeZX60kPbFsYOpXJwkL32HUM50MY22Eb1fOSpZK7WJpYjx66EWOkJvO
+Ey3r1dLdNAjrZrYzQlxRwNS05CaNKoX5rr3NoDnmv+UNobaiS25muLLiqOt6UrStaWow9AUyOHSzl
+BrX2Vu0whv74djK
+3COO6cXT2CBO6WajTJsOgYdxMfv24KWSKw0TqvHe8XDKASQGKdgfI74hzI1YWJlNjlmLWFlMTAtNDRjZ
C1iYzg3LTFiZGExZGUzYjdkNwAAAACdcfNpLXwDUPH3023MvZK8RPHe129k6VdNIi3zNb0eFQ==" 
}
```

  or

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

Status Codes

Table 4-74 lists the normal status code returned by the response.

Table 4-74 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.18 Decrypting Data

Function

This API enables you to decrypt data.

URI

- URI format
  POST /v1.0/{project_id}/kms/decrypt-data
- Parameter description

  Table 4-75 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-76 Request parameters

Parameter	Mandatory	Type	Description
cipher_text	Yes	String	Ciphertext of encrypted data. The value is the **cipher_text** value in the data encryption result that matches the regular expression **^[0-9a-zA-Z +/=]{188,5648}$**.
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
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-77 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	CMK ID
plain_text	Yes	String	Plaintext

Examples

The following example describes how to decrypt data (ciphertext: **AgDoAG7EsEc2OHpQxz4gDFDH54CqwaelpTdEl+RFPjbKn5klPTvOywYIeZX60kPbFsYOpXJwkL32HUM50MY22Eb1fOSpZK7WJpYjx66EWOkJvO+Ey3r1dLdNAjrZrYzQlxRwNS05CaNKoX5rr3NoDnmv+UNobaiS25muLLiqOt6UrStaWow9AUyOHSzl+BrX2Vu0whv74djK+3COO6cXT2CBO6WajTJsOgYdxMfv24KWSKw0TqvHe8XDKASQGKdgfI74hzI1YWJlNjlmLWFlMTAtNDRjZC1iYzg3LTFiZGExZGUzYjdkNwAAAACdcfNpLXwDUPH3023MvZK8RPHe129k6VdNIi3zNb0eFQ==**).

- Example request

  ```
  {
      "cipher_text": "AgDoAG7EsEc2OHpQxz4gDFDH54CqwaelpTdEl
  +RFPjbKn5klPTvOywYIeZX60kPbFsYOpXJwkL32HUM50MY22Eb1fOSpZK7WJpYjx66EWOkJvO
  +Ey3r1dLdNAjrZrYzQlxRwNS05CaNKoX5rr3NoDnmv+UNobaiS25muLLiqOt6UrStaWow9AUyOHSzl
  +BrX2Vu0whv74djK
  +3COO6cXT2CBO6WajTJsOgYdxMfv24KWSKw0TqvHe8XDKASQGKdgfI74hzI1YWJlNjlmLWFlMTAtNDRjZ
  C1iYzg3LTFiZGExZGUzYjdkNwAAAACdcfNpLXwDUPH3023MvZK8RPHe129k6VdNIi3zNb0eFQ=="
  }
  ```

- Example response

  ```
  {
     "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
     "plain_text": "12345678"
  }
  ```

  or

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


Status Codes

Table 4-78 lists the normal status code returned by the response.

Table 4-78 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.19 Obtaining CMK Import Parameters

Function

This API enables you to obtain necessary parameters to import a CMK, including a CMK import token and a CMK encryption public key.

NOTE

The returned public key type is RSA_2048 by default.

URI

- URI format

  POST /v1.0/{project_id}/kms/get-parameters-for-import
- Parameter description

  Table 4-79 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-80 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
wrapping_algorithm	Yes	String	Encryption algorithm for CMK material. The following values are enumerated: ● RSAES_PKCS1_V1_5 ● RSAES_OAEP_SHA_1 ● RSAES_OAEP_SHA_256
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-81 Response parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	ID of a CMK in Base64 format
import_token	Yes	String	CMK import token
expiration_time	Yes	String	Expiration time of the import parameter. The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
public_key	Yes	String	Public key (in Base64 format) used to encrypt CMK material

Examples

The following example describes how to obtain the imported parameter of a CMK (ID: **bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e**; encryption algorithm: **RSAES_OAEP_SHA_1**).

- Example request

```
{
   "key_id": "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "wrapping_algorithm":"RSAES_OAEP_SHA_1"
}
```

- Example response

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

"import_token":"AACIBjY2ZTQxYjBmLTY3ZWItNDU4Ny04OTIxLWVhZTVhZjg5NDZmYQAAuihvPN7Hly3
uhp7cWw4cfuwDIem9mGwall7/HTx10+8ENsRR4FB7DCR+zG1s7UIZMAZRLx7LD1lkXY
+rfN5ibDOOHZkoIiVSh+9u7xtC5m/
mNpIFeyqumxHei2I8CNdsNuJtjLV5bDU3tQrIkj72HCWpC0k9yf1ZSvi3yCwD4wyULXBsYwUa76bTK85MIZ
NGGtqfOyV6w74MT6m70gLhog8r7oWe6Gbof58uyYfFMbc
+s0OpkzMjvlvl1HApyOTijled26VgbgoGbPm9QvgjxC7mQEJpzQeg1/uNiziAG0YKo7wuD2mojwMBnr
+XGJrrFgmdO0pUaK+53KtDr8dtpGrVfJ+0zvebA45c4A4VfvaQQDCI5nJvB2Zz3LM4oiullVt
+0xrwDJYn9KRNZto2/zsGzrc/iBVASKE2UpIH7IikALJuDNrla8MVP5lzxdE0I
+905U2O7HLOsIwlDKMXx3CFao+4qLTb2O+Mq6xMQUwR2pwLcQA1cw
+BypJe4XE3z4fqFejO6VzjX5yd5pDVQ19eAzr9RgvSCi/IuyefFUci
+aX7xB4jx5MNwej3aePsOC9afsXBulhFyGgS/dZoPQ9kyG5TE2ELqAN6obERYoiZcyvq8RW9w/
uItLS99nGjwVe3U1yW4P6ColV+u7ygWxXm/Zs
+QTJHUDwl2ysbrebnN9PLNJSpHbBmuLjiMX02xtDAIt1meB2hGLqw+Mj/
n1jF5rnt5eXrNiG94pHZEvbp2BEDawJrRpaGj15C984WVw8ja/ZrTYfWklcNKW84cLvJXl9vxsuUp3/
ZYKh32M/ORUT46o6KtB/
xEltkaDJiSBBK4utuxQ8wO5UXW6FRkmAuV2naxhF6Obk7kEKYnuj4jxWAODtL9GcoNwq04ylSXj/
ZzaYbqXo1O34fjyz3QG5ZChXGgg52+wPj2LBDjUvlriuARg7cATgdqq9c6aifrGQAj0QgVp9Gv/
8c7PRzjzfH2vRwOZqpLSuCD5sIWFSgc/RLxf1YNtNx98Jo
+PjRTWbyuZNiH2xOrpG0oKyk1giFITqOTuQ6UL768HgVJPRP4CgkgF7v65QpYaYgPvkJvwOb7j2VMr5Voy
kTipt7R2Xvh2LMy6wBW+HA0rw8V7ebc8/
KaH3CkGTdYL2MIfbOlxjyNplUeBKu8zYshFWfp7BUQsflAFMQyp2FhO7PGMygvqY0LLzDphVvBjpFCO4V
qHZ/iOSDzL8vuEA
+OX8XLhZp9Kb7JPIJflfEz2lx3K8YvOJeRxUfOgvvBhpKu7KUDvnarW1R9rDX4adD4EC3mgP42SumAMYvF
BKb6BgOkGAlTgHgLrKKsDw4DW56ANua30ZjeKJ1ZVftnyU0UJ34jsY0uJPi6QujBHqUzFbCp019Jx8Mi
+LtkN3e8Sl+4pvIfj7t+t9Xu03oDhD0J65qhHlpNP/NFrvP3KLmXFyXTWpGeczXxZvDp7Wmu5TnDSozN/
AbzBuyWASYZpLvgsf1xwevMmM1Gw/UX/
WVPQdN5lzWjhT1Dcy4ar8OozYtQeQ2ItSH1UaPJx0hW3BA1GYjW42+Vjy0VSLkliK/n6lN9KwTTTGAbW
+BvftImzGnfFM7fTCMJ3Jnx9nTn6+fbnhoXXfGHjOgPZ208VEIlG5YHS+HN/
JYyAkkj8G2+bSZmKfX9VMbYRGNTPrrghjAEY/Hh8V+/
ZhUSR3pPnblhr30SePGYgQPUGmnoTRHulCHRFOMcvu9nQ1P855DNpoE7fYi
+7N9xu1wFTB3DHtgUW8yuwtt
+q6LJZQMuGfmJLhBBf05FKlSxpR49IaJ0uQc7fsVYCpECL2aH8ueBqVGvQtEebWG6q0XTIrhqmaPtlQx9rVP
8oevPZ99yfB+8TZCT0B9WNqCotxijWqH3eyePY0Hb/AAXB34GjH1gni4NjwEI6LVX
+jSGb2ATy4Bd6ckonhGO9uwvW3WaPX214+GZvPdmv0pN60XfQ9B4Il/
RLIek6h6+2WEmB4i8qsvjgWfDD7DEhq6YN1Q/44NqUdDjrVCozBxXyDOab5tdsWCvfGXruGa/
wq7I1kH7K76s7TeL0a3pc0H5zt8qU/UT7uoLv0G7H+vVulGmqcl5pbsHYxTqNtSu2w9OBQ6PC8g+MCS/
fnXIcAhS7Lmvy8TFK4x0N+MhZqVbozVW37apCXFg6m1I9N0Sa4=",
   "expiration_time":1501578672,

"public_key":"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAnJQqE9GorZ16XMlOQngJfU0Sg
kMKJpL9W+byIebeKgmDt2I6oVSPckk9y3JiaGjXKYlepawob9b61IRR97Bcr4Sf2p3J6J3gpiYGp1Ai3495rYF
+FSZAxW+VDOzbN3vig6SVxcP1PXtaKzQbtNfnlIh+rvSMJpVI3MFHh5lWjEn8L/
XpprLy1FqHSSvgB99qwiPw1ZGTL5XGSrIpCV3/ah8u+5VGoIUJZTtiZk6OQDkFH9fxwIahYvLI8/
yjrWFLtJuApr7aIrhRN0iDBINxddNh8M0A9sIFoS3D5RNKITjIKIMl/GVz+mHaPjK+91M/
b7JrNvinFCMQDGrb/1qoGQIDAQAB"
}
```

or

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



```
    }
}
```

Status Codes

Table 4-82 lists the normal status code returned by the response.

Table 4-82 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.20 Importing CMK Material

Function

This API allows you to import CMK material.

URI

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

  Table 4-83 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-84 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

Parameter	Mandatory	Type	Description
import_token	Yes	String	CMK import token in Base64 format that matches the regular expression **^[0-9a-zA-Z+/=]{200,6144}$**
encrypted_key_material	Yes	String	Encrypted CMK material in Base64 format that matches the regular expression ^[0-9a-zA-Z+/=]{344,360}$
expiration_time	No	String	Expiration time of the key material. The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970. KMS will delete the key material within 24 hours after the expiration. Example: 1550291833
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

None

Examples

The following example describes how to import the CMK material and the import-token to the CMK whose ID is **bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e**, and set the expiration time of the CMK material to **1521578672**.

- Example request

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

"import_token":"AACIBjY2ZTQxYjBmLTY3ZWItNDU4Ny04OTIxLWVhZTVhZjg5NDZmYQAAuihvPN7Hly3
uhp7cWw4cfuwDIem9mGwaIl7/HTx10+8ENsRR4FB7DCR+zG1s7UIZMAZRLx7LD1lkXY
+rfN5ibDOOHZkoIiVSh+9u7xtC5m/
mNpIFeyqumxHei2I8CNdsNuJtjLV5bDU3tQrIkj72HCWpC0k9yf1ZSvi3yCwD4wyULXBsYwUa76bTK85MIZ
NGGtqfOyV6w74MT6m70gLhog8r7oWe6Gbof58uyYfFMbc
+s0OpkzMjvlvl1HApyOTijled26VgbgoGbPm9QvgjxC7mQEJpzQeg1/uNiziAG0YKo7wuD2mojwMBnr
+XGJrrFgmdO0pUaK+53KtDr8dtpGrVfJ+0zvebA45c4A4VfvaQQDCI5nJvB2Zz3LM4oiullVt
+0xrwDJYn9KRNZto2/zsGzrc/iBVASKE2UpIH7IikALJuDNrla8MVP5lzxdE0I
+905U2O7HLOsIwlDKMXx3CFao+4qLTb2O+Mq6xMQUwR2pwLcQA1cw
+BypJe4XE3z4fqFejO6VzjX5yd5pDVQ19eAzr9RgvSCi/luyefFUci
+aX7xB4jx5MNwej3aePsOC9afsXBulhFyGgS/dZoPQ9kyG5TE2ELqAN6obERYoiZcyvq8RW9w/
uItLS99nGjwVe3U1yW4P6ColV+u7ygWxXm/Zs
+QTJHUDwl2ysbrebnN9PLNJSpHbBmuLJiMX02xtDAIt1meB2hGLqw+Mj/
n1jF5rnt5eXrNiG94pHZEvbp2BEDawJrRpaGj15C984WVw8ja/ZrTYfWklcNKW84cLvJXl9vxsuUp3/
ZYKh32M/ORUT46o6KtB/
xEltkaDJiSBBK4utuxQ8wO5UXW6FRkmAuV2naxhF6Obk7kEKYnuj4jxWAODtL9GcoNwq04ylSXj/
ZzaYbqXo1O34fjyz3QG5ZChXGgg52+wPj2LBDjUvlriuARg7cATgdqq9c6aifrGQAj0QgVp9Gv/
8c7PRzjzfH2vRwOZqpLSuCD5sIWFSgc/RLxf1YNtNx98Jo
```

+PjRTWbyuZNiH2xOrpG0oKyk1giFITqOTuQ6UL768HgVJPRP4CgkgF7v65QpYaYgPvkJvwOb7j2VMr5Voy
kTipt7R2Xvh2LMy6wBW+HA0rw8V7ebc8/
KaH3CkGTdYL2MIfbOlxjyNplUeBKu8zYshFWfp7BUQsflAFMQyp2FhO7PGMygvqY0LLzDphVvBjpFCO4V
qHZ/iOSDzL8vuEA
+OX8XLhZp9Kb7JPIJflfEz2lx3K8YvOJeRxUfOgvvBhpKu7KUDvnarW1R9rDX4adD4EC3mgP42SumAMYvF
BKb6BgOkGAlTgHgLrKKsDw4DW56ANua30ZjeKJ1ZVftnyU0UJ34jsY0uJPi6QujBHqUzFbCp019Jx8Mi
+LtkN3e8Sl+4pvIfj7t+t9Xu03oDhD0J65qhHlpNP/NFrvP3KLmXFyXTWpGeczXxZvDp7Wmu5TnDSozN/
AbzBuyWASYZpLvgsf1xwevMmM1Gw/UX/
WVPQdN5lzWjhT1Dcy4ar8OozYtQeQ2ItSH1UaPJx0hW3BA1GYjW42+Vjy0VSLkliK/n6lN9KwTTTGAbW
+BvftImzGnfFM7fTCMJ3Jnx9nTn6+fbnhoXXfGHjOgPZ208VEIlG5YHS+HN/
JYyAkkj8G2+bSZmKfX9VMbYRGNTPrrghjAEY/Hh8V+/
ZhUSR3pPnblhr30SePGYgQPUGmnoTRHulCHRFOMcvu9nQ1P855DNpoE7fYi
+7N9xu1wFTB3DHtgUW8yuwtt
+q6LJZQMuGfmJLhBBf05FKlSxpR49IaJ0uQc7fsVYCpECL2aH8ueBqVGvQtEebWG6q0XTIrhqmaPtlQx9rVP
8oevPZ99yfB+8TZCT0B9WNqCotxijWqH3eyePY0Hb/AAXB34GjH1gni4NjwEI6LVX
+jSGb2ATy4Bd6ckonhGO9uwvW3WaPX214+GZvPdmv0pN60XfQ9B4Il/
RLIek6h6+2WEmB4i8qsvjgWfDD7DEhq6YN1Q/44NqUdDjrVCozBxXyDOab5tdsWCvfGXruGa/
wq7I1kH7K76s7TeL0a3pc0H5zt8qU/UT7uoLv0G7H+vVulGmqcl5pbsHYxTqNtSu2w9OBQ6PC8g+MCS/
fnXIcAhS7Lmvy8TFK4x0N+MhZqVbozVW37apCXFg6m1I9N0Sa4=",
"encrypted_key_material":"K+ixymtl90e
+B5Rdan89KjDslBIoOexrIwzkYHGz3odS7FDXDkogqbWwwJg5wQ6zjUbEvsR/+Fi
+A0SSkhhqtijivOKHu4Z86RWjOCBdrr9es+ZhJ0zYBNMN+7Rf2fd9vxbb873Q7VBkJRyH1hi3Wh
+kLmDW4rpWZm4+YGCtWylz7ZKbV1KBlhSNLDtZzT4nxUra0p7Die4HgUUxSjZTOr/0s71yF6o2eysreIzIl
+GbpCft0WpRsxN2Ng++ntgOcwOf2zOC9o/tjraxeAvgGw
+Dwt4cjF4znnFf0LPQ2YvpNUo248LjAGxdFvzUABNzfYSj3RZ0K3wQCNAcXU3HYw==",
"expiration_time":1521578672
}

- Example response

```
{
}
```

or

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

Status Codes

Table 4-85 lists the normal status code returned by the response.

Table 4-85 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.21 Deleting CMK Material

Function

This API allows you to delete CMK material.

URI

- URI format
  POST /v1.0/{project_id}/kms/delete-imported-key-material
- Parameter description

  Table 4-86 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-87 Request parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

None

Examples

The following example describes how to delete the material of a CMK (ID: **0d0466b0-e727-4d9c-b35d-f84bb474a37f**).

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


Status Codes

Table 4-88 lists the normal status code returned by the response.

Table 4-88 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.22 Querying CMK Instances

Function

This API allows you to query CMK instances.

You can use the tag filtering function to query the detailed information about a specified CMK.

URI

- URI format

  POST /v1.0/{project_id}/kms/resource_instances/action
- Parameter description

  Table 4-89 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 4-90 Request parameters

Parameter	Mandatory	Type	Description
tags	No	Array of objects	list of tags, including tag keys and tag values. • **key** indicates the tag key. A CMK can have a maximum of 10 keys, and each of them is unique and cannot be empty. A key cannot have duplicate values. The value of **key** contains a maximum of 36 characters. • **value** indicates the tag value. Each tag value can contain a maximum of 43 characters. The relationship between values is **AND**.
limit	No	String	Number of queried records. If **action** is set to **count**, this parameter does not need to be set. If **action** is set to **filter**, the default value is **10**. The value ranges from 1 to 1000.
offset	No	String	Index location. The query starts from the next piece of data indexed by this parameter. When data on the first page is queried, the value of this parameter queried on previous page is contained. If **action** is **count**, this parameter does not need to be set. If **action** is set to **filter**, the default value is **0**. The value must be a numeral and cannot be a negative number.
action	Yes	String	Operation ID, which can be set to **filter** or **count**. • **filter**: indicates filtering. • **count**: indicates the number of queried records.

Parameter	Mandatory	Type	Description
matches	No	Array of objects	Search field. • **key** indicates the field to be matched, for example, **resource_name**. • **value** indicates the value to be matched, which contains a maximum of 255 characters and cannot be empty.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

Table 4-91 Response parameters

Parameter	Mandatory	Type	Description
resources	Yes	Array of objects	Resource instance list. For details, see **Table 4-92**.
total_count	Yes	Integer	Total number of records

Table 4-92 resource field description

Parameter	Mandatory	Type	Description
resource_id	Yes	String	Resource ID
resource_detail	Yes	Object	Resource details. For details, see **Table 4-31**.
tags	Yes	Array of objects	Lists of tags. If there is no tag, the array is empty by default.
resource_name	Yes	String	Resource name. This parameter is an empty string by default.

Examples

The following example describes how to query key instances.

- Example request

```
{
      "offset": "100",
      "limit": "100",
      "action": "filter",
      "matches":[
      {
            "key": "resource_name",
            "value": "resource1"
       }
      ],
      "tags": [
          {
            "key": "key1",
            "values": [
                  "value1",
                  "value2"
            ]
          }
      ]
  }
```

- Example response

```
{
 "resources": [{
      "resource_id": "90c03e67-5534-4ed0-acfa-89780e47a535",
      "resource_detail": {
            "key_id": "90c03e67-5534-4ed0-acfa-89780e47a535",
            "domain_id": "4B688Fb77412Aee5570E7ecdbeB5afdc",
            "key_alias": "tagTest_xmdmi",
            "key_description": "123",
            "creation_date": 1521449277000,
            "scheduled_deletion_date": "",
            "key_state": 2,
            "default_key_flag": 0,
            "key_type": 1
      },
      "resource_name": "tagTest_xmdmi",
      "tags": [{
            "key": "$",
            "value": "testValue!"
      }, {
            "key": "1",
            "value": "ccwZ"
      }, {
            "key": "1&",
            "value": "testValue!"
      }, {
            "key": "abcd",
            "value": "1&"
      }, {
            "key": "efg",
            "value": "1&"
      }, {
            "key": "faregbqer",
            "value": "AAaa00-99"
      }, {
            "key": "fcwefwq",
            "value": "$"
      }, {
            "key": "fwqegqwrg",
            "value": "1&"
      }, {
            "key": "haha",
            "value": "qzzahnzgoqbkabppdehnbrrgbrkvlxkkfoosqyhdylq"
```



```
    }, {
        "key": "quapxpysduboguiluwargcgmvcgxinianbhl",
        "value": "testValue!"
    }]
}]
"total_count": "1"}
```

or

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

Status Codes

Table 4-93 lists the normal status code returned by the response.

Table 4-93 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.23 Querying CMK Tags

Function

This API allows you to query tags of a specified CMK.

TMS may use this API to query all tags of a specified CMK.

URI

- URI format
  GET /v1.0/{project_id}/kms/{key_id}/tags
- Parameter description

  Table 4-94 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Parameter	Mandatory	Type	Description
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

Requests

None

Responses

Table 4-95 Response parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of objects	list of tags, including tag keys and tag values. • **key** indicates the tag key. A CMK can have a maximum of 10 keys, and each of them is unique and cannot be empty. A key cannot have duplicate values. The value of **key** contains a maximum of 36 characters. • **value** indicates the tag value. Each tag value can contain a maximum of 43 characters. The relationship between values is **AND**.
existTagNum	Yes	Integer	Number of key tags.

Examples

The following example describes how to query CMK tags.

- Example request

  None

- Example response

```
{       "tags": [
         {
          "key": "key1",
          "value": "value1"
         },
```



```
        {
         "key": "key2",
         "value": "value3"
        }
        ],
        "existTagsNum":2
}
```

or

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

Status Codes

Table 4-96 lists the normal status code returned by the response.

Table 4-96 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.24 Querying Project Tags

Function

This API enables you to query all tag sets of a specified project.

URI

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

Table 4-97 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

None

Responses

Table 4-98 Response parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of objects	list of tags, including tag keys and tag values. • **key** indicates the tag key. A CMK can have a maximum of 10 keys, and each of them is unique and cannot be empty. A key cannot have duplicate values. The value of **key** contains a maximum of 36 characters. • **value** indicates the tag value. Each tag value can contain a maximum of 43 characters. The relationship between values is **AND**.

Examples

The following example describes how to query project tags.

- Example request

  None

- Example response

```
{          "tags": [
          {
            "key": "key1",
            "values": [
               "value1",
               "value2"
             ]
          },
          {
            "key": "key2",
            "values": [
               "value1",
               "value2"
             ]
     }
]
}
```

  or

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

Status Codes

Table 4-99 lists the normal status code returned by the response.

Table 4-99 Status codes

Status Code	Status	Description
200	OK	Request processed successfully.

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

4.25 Adding or Deleting CMK Tags in Batches

Function

This API enables you to add or delete CMK tags in batches.

URI

- URI format
  POST /v1.0/{project_id}/kms/{key_id}/tags/action
- Parameter description

Table 4-100 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

Requests

Table 4-101 Request parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of objects	list of tags, including tag keys and tag values. • **key** indicates the tag key. A CMK can have a maximum of 10 keys, and each of them is unique and cannot be empty. A key cannot have duplicate values. The value of **key** contains a maximum of 36 characters. • **value** indicates the tag value. Each tag value can contain a maximum of 43 characters. The relationship between values is **AND**.
action	Yes	String	Operation ID. The value can be **create** or **delete**.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Responses

None

Examples

The following example describes how to add tags, the keys and values of which are **key1**, **key**, **value1**, and **value3** respectively.

- Example request

```
{
   "action": "create",
   "tags": [
      {
         "key": "key1",
         "value": "value1"
      },
      {
         "key": "key",
         "value": "value3"
      }
   ]
}
```

or


```
{
   "action": "delete",
   "tags": [
      {
         "key": "key1",
         "value": "value1"
      },
      {
         "key": "key2",
         "value": "value3"
      }
   ]
}
```

- Example response

```
{
}
```

or

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

Status Codes

Table 4-102 lists the normal status code returned by the response.

Table 4-102 Status codes

Status Code	Status	Description
204	No Content	The request is processed successfully and no content is returned.

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

4.26 Adding a CMK Tag

Function

This API allows you to add a CMK tag.

URI

- URI format

  POST /v1.0/{project_id}/kms/{key_id}/tags
- Parameter description

Table 4-103 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f

Requests

Table 4-104 Request parameters

Parameter	Mandatory	Type	Description
tag	Yes	Array of object	Tag. For details, see **Table 4-105**.
sequence	No	String	36-byte serial number of a request message Example: 919c82d4-8046-4722-9094-35c3c6524cff

Table 4-105 tag field description

Parameter	Mandatory	Type	Description
key	Yes	String	Key. The value contains a maximum of 36 Unicode characters. The value of **key** cannot be empty, and cannot contain the following characters: ASCII (0-31) and ***<>\=**
value	Yes	String	Value. Each value contains a maximum of 43 Unicode characters and can be an empty string. The value cannot contain the following characters: ASCII (0-31) and ***<>\=**

Responses

None

Examples

The following example describes how to add a tag, the key and value of which are **DEV** and **DEV1** respectively.

- Example request

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

- Example response

```
{
}
```

or

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

Status Codes

Table 4-106 lists the normal status code returned by the response.

Table 4-106 Status codes

Status Code	Status	Description
204	No Content	The request is processed successfully and no content is returned.

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

4.27 Deleting a CMK Tag

Function

This API enables you to delete a CMK tag.

URI

- URI format

  DELETE /v1.0/{project_id}/kms/{key_id}/tags/{key}

- Parameter description

Table 4-107 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	36-byte ID of a CMK 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}$** Example: 0d0466b0-e727-4d9c-b35d-f84bb474a37f
key	Yes	String	Tag key

Requests

Table 4-108 Request parameters

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

Responses

None

Examples

Example response

```
{
}
```

or

```
{
    "error": {
```



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

Status Codes

Table 4-109 lists the normal status code returned by the response.

Table 4-109 Status codes

Status Code	Status	Description
204	No Content	The request is processed successfully and no content is returned.

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

5 Permissions Policies and Supported Actions

5.1 Introduction

This chapter describes fine-grained permissions management for your KMS. If your account does not need individual IAM users, then 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 users permissions 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 or enterprise project: Scope of users a permission is granted to. 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

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

Manage keys, such as creating keys and querying keys.

5.2 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 Code

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_descriptio n.	Invalid key_descriptio n.	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 1096 days.	Set the pending deletion period between 7 to 1096 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 Enabled or Disabled state and cannot be deleted.	Ensure that the key is in 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

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 operations:

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 Encryption Key Management

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
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

API	API Function	Permission
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

B Change History

Release Date	Description
2022-11-28	This is the third official release. Added: • Example response in section "Creating a CMK" Modified the following content: • Optimized the content in section "Scheduling the Deletion of a CMK". • Modified the error code format in the "Error Codes" section. • Added the **key_info** parameter and its description in "Creating a CMK". • Added the **key_info** parameter and its description in "Enabling a CMK".
2021-06-03	This is the second official release. Added section "Permissions and Supported Actions".
2021-01-27	This is the first official release.

