Data Encryption Workshop

API Reference

Issue 21

Date 2024-08-02

HUAWEI

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.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Before You Start

1.1 Overview

Welcome to the *Data Encryption Workshop API Reference*. DEW is a comprehensive data encryption service in the cloud. It provides Key Management Service (KMS), Dedicated Hardware Security Module (Dedicated HSM), and Key Pair Service (KPS). DEW uses HSMs to protect the security of your keys, and can be integrated with other Huawei cloud services to address data security, key security, and key management issues. Additionally, DEW enables you to develop customized encryption applications.

Before calling DEW APIs, ensure that you have understood the concepts related to DEW. For more information, see **What Is DEW?**

1.2 API Calling

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

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

The number of keys that you can create is determined by your quota. For details, see **Service Quota**.

In KMS, TPS (the number of API operations that can be performed by a user per second) is set to **20**.

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 directly to perform routine management. For security purposes, create users and grant them permissions for routine management.

- User

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

  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.

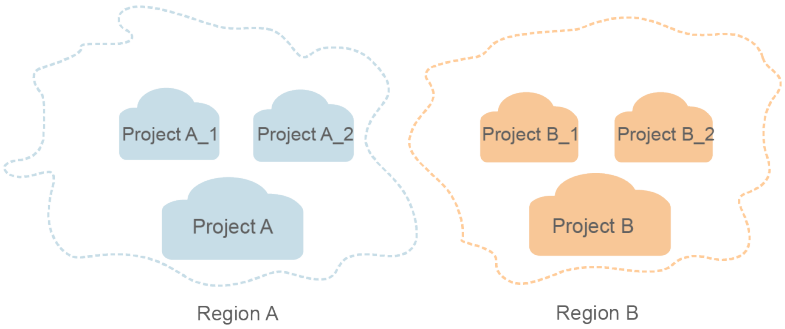
- Availability Zone (AZ)

  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.

  For details about how to obtain enterprise project IDs and features, see **Applicable Scenarios**.

1.6 Selecting an API Type

For SSH key pairs, V3, V2.1, and V2 APIs are available. V3 APIs are recommended.

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 region **CN-Hong Kong** is **iam.ap-southeast-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**:

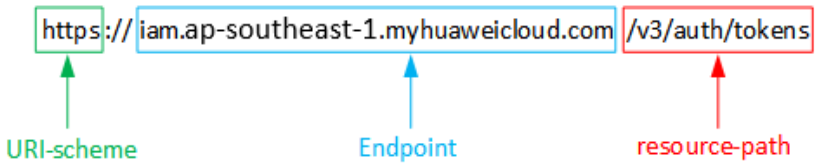
  Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in region **CN North-Hong Kong**, obtain the endpoint of IAM (**iam.cn-ap-southeast-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.ap-southeast-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.ap-southeast-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.ap-southeast-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. Replace *username*, *domainname*, ********* (login password), and *xxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain the value of region 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.ap-southeast-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. You can use this token 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 authentication: Requests are encrypted using AK/SK pairs. 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.

You can **obtain a token** by calling an API. A project-level token is required for calling DEW APIs. When calling an API to obtain a user token, set **project** in **auth.scope** in the request body, as shown in the following 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.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK-based Authentication

NOTE

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

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

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

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

NOTICE

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

2.3 Response

Status Codes

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

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

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

Response Header

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

Figure 2-2 shows the response header for the API of **obtaining a user token**, in which **x-subject-token** is the desired user token. You can use this 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 functions of DEW by using the APIs provided by DEW and the performance parameters corresponding to the APIs.

Performance parameter types include shared traffic control and basic traffic control.

- Shared traffic control: APIs share the traffic.
- Basic traffic control: Only the current APIs can use the traffic.

NOTE

Generally, APIs share the traffic unless labeled as basic traffic control. For example, secret version creation APIs share the traffic.

Type	Description
Key Management APIs	Create, query, modify, and delete keys.
Secret management APIs	Create, query, modify, and delete secrets.
Key Pair Management APIs	(Latest API version) Create, query, modify, and delete key pairs.
Historical Global	(V2.1 and V2 API versions) Create, query, modify, and delete key pairs.

Key Management APIs

Type	Name	Description	Performance
Global API version query	Query version list	Obtain the API version list.	-
	Query a version	Query a specified API version.	

Type	Name	Description	Performance
Lifecycle management	Create a key	Create a CMK, which can be symmetric or asymmetric.	20 times per second for a single user 100 times per second globally
	Enable a key	Enable a key, which can only be used after being enabled.	
	Disable a key	Disable a CMK. A disabled CMK cannot be used.	
	Schedule the deletion of a key	Schedule a deletion task for a specified key. The deletion can be scheduled 7 to 1,096 days in advance. After a key is deleted, the data encrypted using the key cannot be decrypted.	
	Cancel the scheduled deletion of a key	Cancel a scheduled deletion of a key. Once the deletion is cancelled, the key can be used.	
	Modify a key alias	Change the alias of a CMK.	
	Modify CMK description	Change the description of a CMK.	
Data encryption key (DEK) management	Generate a random number	Generate a random number that is 8 bits to 8,192 bits long.	800 times per second for a single user 10,00 times per second globally
	Create a DEK	Create a DEK. The returned result includes the plaintext and the ciphertext of a DEK.	
	Create a plaintext-free DEK	Create a plaintext-free DEK. The returned result includes only the plaintext of a DEK.	
	Encrypt a DEK	Use a specified CMK to encrypt a DEK.	
	Decrypt a DEK	Use a specified CMK to decrypt a DEK.	
Key import management	Obtain parameters for importing a key	Obtain necessary parameters to import a key, including an import token and a public key.	20 times per second for a single user 100 times per second globally
	Import key materials	Import the material of a key.	

Type	Name	Description	Performance
	Delete key materials	Delete the material of a key.	
Authorization management	**Create a grant**	Grant a user with key operation permissions.	20 times per second for a single user 100 times per second globally
	Revoke a grant	Revoke the key operation permissions granted to a user.	
	Retire a grant	Retire the granted key operation permissions.	
	Query the grant list	Query grants on a CMK.	
	Query grants that can be retired	Query grants that can be retired.	
Small-size data encryption and decryption	**Encrypt data**	Use a specified CMK to encrypt data.	20 times per second for a single user 100 times per second globally
	Decrypt data	Decrypt data.	
Signature and verification	**Signing Data**	Digitally sign a message or message digest using the private key of an asymmetric key.	300 times per second for a single user 500 times per second globally
	Verifying a Signature	Verify the signature of a message or message digest using the public key of an asymmetric key.	
Rotation management	**Enable key rotation**	Enable the rotation of a CMK. Default master keys and imported keys cannot be rotated.	20 times per second for a single user 100 times per second globally
	Disable key rotation	Disable the rotation of a CMK.	
	Modify key rotation interval	Change the rotation interval for a CMK.	
	Query key rotation status	Query the rotation status of a CMK.	

Type	Name	Description	Performance
Tag management	Query key instances	Use tag filtering to query the detailed information of a CMK.	20 times per second for a single user 100 times per second globally
	Query key tags	Query tags of a CMK.	
	Add tags to a key	Query all tag sets of a project.	
	Query project tags	Add or delete CMK tags in batches.	
	Batch add or delete key tags	Add a tag to a CMK.	
	Delete key tags	Delete a tag from a CMK.	
Query	Query the key list	Obtain the list of all CMKs.	160 times per second for a single user 200 times per second globally
	Query key details	Query details of a specified key.	
	Query instance quantity	Obtain the number of created CMKs, excluding the default master keys.	80 times per second for a single user 200 times per second globally
	Query quotas	Query the total quota of CMKs available and the usage information, excluding the default master keys.	

CSMS APIs

Type	Name	Description	Quota
Lifecycle management	Creating a Secret	Create a secret and stores the secret value in the initial secret version.	300 times per minute for a single user 4,800 times per minute globally

Type	Name	Description	Quota
	Querying the Secret List	Query all the secrets created by the current user in the current project.	100 times per second for a single user 200 times per second globally
	Querying a Secret	Query a specified secret.	1,200 times per minute for a single user 4,800 times per minute globally
	Updating a Secret	Update the metadata of a specified secret.	300 times per minute for a single user 4,800 times per minute globally
	Deleting a Secret Immediately	Delete a specified secret. The deleted secret cannot be restored.	
	Restoring a Secret Object	Restore a secret by uploading the secret backup file.	
	Downloading Secret Backup	Download the backup file of a specified secret.	
	Creating a Scheduled Secret Deletion Task	Create a scheduled task to delete a secret after 7 to 30 days.	
	Canceling a Scheduled Secret Deletion Task	Cancel the scheduled deletion task of a secret. The secret will be changed to the available state.	
	Rotating a Secret	Execute rotation for a secret immediately. Create a new version of a secret to encrypt to encrypt and keep the generated random secret value. The created secret version is in **SYSCURRENT** state.	

Type	Name	Description	Quota
Secret version management	Creating a Secret Version	Create a new version of a secret to encrypt and keep the new value of the secret. By default, the created secret version in **SYSCURRENT** state. The previous version is in **SYSPREVIOUS** state. You can configure **VersionStage** to overwrite the default settings.	Basic traffic control: 80 times per second for a single user 200 times per second globally 80 times per second for applications 80 times per second for IP addresses
	Querying the Secret Version List	Query the version list of a specific secret.	300 times per minute for a single user 4,800 times per minute globally
	Updating the Secret Version	Currently, only the version validity period of a secret whose status is **ENABLED** can be updated. If the associated subscription events include version expired events, only one notification is triggered each time the version validity period is updated.	
	Querying the Secret Version and Value	Query the information about a specified secret version and the plaintext secret value in the version. Only secrets in **Enabled** state can be queried. The value of the latest secret version can be obtained via **/v1/{project_id}/secrets/{secret_name}/versions/latest**. (Set the **{version_id}** in the URL of the current API to **latest**).	Basic traffic control: 160 times per second for a single user 200 times per second globally 160 times per second for applications 160 times per second for IP addresses
Secret version status management	Updating the Version Status of a Secret	Update the version status of a secret.	300 times per minute for a single user 4,800 times per minute globally

Type	Name	Description	Quota
	Querying the Version Status of a Secret	Query the version of a specified secret version status tag.	
	Deleting the Version Status of a Secret	Delete the status of a specified secret version.	
Secret tag management	Querying a Secret Instance	Query a secret instance. Filter user secrets by tag and returns the secret list.	300 times per minute for a single user 4,800 times per minute globally
	Adding or Deleting Secret Tags in Batches	Add or delete secret tags in batches.	
	Querying Secret Tags	Query secret tags.	
	Querying Secret Tags	Add a secret tag.	
	Deleting a Secret Tag	Delete a secret tag.	
	Querying Project Tags	Query all secret tags of a user in a specified project.	
Incidents	Creating an Event	Create an event that can be configured on one or more secrets. When an event is enabled and the basic event type contained in the event is triggered on the secret, the cloud service sends the corresponding event notification to the notification topic specified by the event.	300 times per minute for a single user 4,800 times per minute globally
	Querying Events	Query information about a specified event.	
	Querying the Event List	Query all events created by the current user in the project.	
	Update an Event	Update the metadata of a specified event. The following metadata can be updated: event enabling status, basic type list, and notification topic.	

Type	Name	Description	Quota
	Deleting an Event Immediately	Delete a specified event. The deleted event cannot be restored. An event cannot be deleted if it is referenced by a secret. Disassociate the event from the secret.	
	Querying Triggered Event Notification Records	Query all event notification records triggered in the last three months.	

SSH Key Pair Management APIs

Type	Name	Description	Quota
Key pair management	Creating and Importing an SSH Key Pair	Create and import an SSH key pair.	300 times per minute for a single user 4,800 times per minute globally
	Accessing the page for clearing private keys	Delete the private key of an SSH key pair.	
	Querying the SSH Key Pair List	Query the list of SSH key pairs.	Basic traffic control: 160 times per second for a single user 200 times per second globally 160 times per second for applications 160 times per second for IP addresses
	Querying SSH Key Pair Details	Query details about an SSH key pair.	

Type	Name	Description	Quota
	Deleting an SSH Key Pair	Delete an SSH key pair.	300 times per minute for a single user 4,800 times per minute globally
	Updating SSH Key Pair Description	Update the description about an SSH key pair.	
	Accessing the page for importing private keys	Import a private key to a specified key pair.	
	Exporting a private key	Export the private key of a specified key pair.	
Key pair task management	Binding an SSH Key Pair	Bind an SSH key pair to a specified VM. The private key of the SSH key pair for the VM is required if you want to replace the key pair, but not required if you want to reset the key pair.	300 times per minute for a single user 4,800 times per minute globally
	Unbind an SSH Key Pair	Unbind an SSH key pair from a specified VM and restores SSH password login.	
	Binding SSH Key Pairs in Batches	Bind SSH key pairs in batches to a specified VM.	Basic traffic control: 10 times per minute for a single user 20 times per minute globally 10 times per minute for applications 10 times per minute for IP addresses
	Querying Task Information	Query the execution status of the current task based on the task ID returned by the SSH key pair API.	300 times per minute for a single user 4,800 times per minute globally
	Querying Running Tasks	Query running tasks.	

Type	Name	Description	Quota
	Querying Task Failure Information	Query information about failed tasks, such as binding and unbinding tasks.	
	Delete all failed tasks	Delete information about failed tasks.	
	Delete a failed task	Delete failed tasks.	

Global History

Global Type	Description
Key pair management APIs (V2.1)	Query the list of key pairs.
	Query details of a key pair.
	Create and import a key pair. You can manage the private keys on the cloud.
	Delete an SSH key pair based on the key pair name.
	Modify description of a key pair of a specified name.
Key pair management APIs (V2.0)	Query the list of key pairs.
	Query a key pair by its name.
	Create a key pair or import a public key to the cloud to generate a key pair. After an SSH key pair is created, you need to download the private key to a local directory. Then, you can use the private key to log in to an ECS. For ECS security purposes, the private key can be downloaded only once. Keep it secure.
	Delete an SSH key pair based on the key pair name.
	A tenant may contain multiple users. This API is used to copy the key pair from the target user to the current user under the same tenant account.

4 APIs

4.1 Key Management APIs

4.1.1 Key Lifecycle Management

4.1.1.1 Create a key

Function

This API is used to create a symmetric or asymmetric CMK. A symmetric key is a 256-bit AES key or a 128-bit SM4 key. It can be used to encrypt a small amount of data or DEKs. An asymmetric key is a RSA key or an ECC key pair (including SM2 key pair). It can be used for data encryption and decryption, digital signature, and signature verification.

Constraints

The alias of the default CMK automatically created by the service ends with **/default**. To avoid a conflict with the default CMK's alias, do not create a CMK whose alias also ends with **/default**. If you have created an enterprise project, the default CMKs can only be stored in the enterprise project. Enterprise resources cannot be moved around. Default CMKs can be used for cloud encryption in non-default enterprise projects to meet compliance requirements. If you need to manage enterprise resources, create a key and use it.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-2 Request header parameters

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

Table 4-3 Request body parameters

Parameter	Mandatory	Type	Description
key_alias	Yes	String	Alias of a non-default master key. The value is a string of 1 to 255 characters that match the regular expression ^[a-zA-Z0-9:/_-]{1,255}$ and must be different from the alias of the Default Master Key.
key_spec	No	String	Key generation algorithm. The default value is **AES_256**. Possible values are as follows: **AES_256 SM4 RSA_2048 RSA_3072 RSA_4096 EC_P256 EC_P384 SM2**
key_usage	No	String	Key usage. The default value of the symmetric key is **ENCRYPT_DECRYPT** and the default value of the asymmetric key is **SIGN_VERIFY**.
key_description	No	String	Key description. It can contain 0 to 255 characters.

Parameter	Mandatory	Type	Description
origin	No	String	Key source. The default value is **kms**. Possible values are as follows: **kms**: The key is generated by KMS. **external**: The key is imported.
enterprise_project_id	No	String	Enterprise project ID If you haven't created any enterprise project, do not configure this field. If you have created an enterprise project, configure this field during resource creation. If this field is not configured, newly created resources are listed in the default enterprise project whose ID is **0**. Note: If you do not have the permission to create resources in the default enterprise project whose ID is **0**, an API error is reported.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**
keystore_id	No	String	Keystore ID. The default KMS keystore is used by default.

Response Parameters

Status code: 200

Table 4-4 Response body parameters

Parameter	Type	Description
key_info	**KeKInfo** object	Key details.

Table 4-5 KeKInfo

Parameter	Type	Description
key_id	String	Key ID.
domain_id	String	User domain ID.

Status code: 400

Table 4-6 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-7 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-8 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-9 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-10 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-11 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-12 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-13 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-14 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-15 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-16 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-17 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-18 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-19 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Create a key whose alias is **test**.

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

Example Responses

Status code: 200

Request succeeded.

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

Status Codes

Status Code	Description
200	Request succeeded.

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

Error Codes

See **Error Codes**.

4.1.1.2 Enable a key

Function

This API is used to enable a key. Only enabled keys can be used. Note: Only keys that are disabled can be enabled.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/enable-key

Table 4-20 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-21 Request header parameters

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

Table 4-22 Request body parameters

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

Response Parameters

Status code: 200

Table 4-23 Response body parameters

Parameter	Type	Description
key_info	**KeyStatusInfo** object	Key status.

Table 4-24 KeyStatusInfo

Parameter	Type	Description
key_id	String	Key ID

Parameter	Type	Description
key_state	String	Key status. Possible values are as follows: **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported **7**: Frozen

Status code: 400

Table 4-25 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-26 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-27 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-28 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-29 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-30 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-31 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-32 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-33 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-34 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-35 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-36 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-37 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-38 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Enable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_info" : {
    "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "key_state" : "2"
```



```
    }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Enable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class EnableKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        EnableKeyRequest request = new EnableKeyRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            EnableKeyResponse response = client.enableKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Enable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.


```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = EnableKeyRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.enable_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Enable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```



```
        Build())

    request := &model.EnableKeyRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.EnableKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.1.3 Disable a key

Function

This API is used to disable a key. A disabled key cannot be used. Only an enabled key can be disabled.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-39 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-40 Request header parameters

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

Table 4-41 Request body parameters

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

Response Parameters

Status code: 200

Table 4-42 Response body parameters

Parameter	Type	Description
key_info	**KeyStatusInfo** object	Key status.

Table 4-43 KeyStatusInfo

Parameter	Type	Description
key_id	String	Key ID
key_state	String	Key status. Possible values are as follows: **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported **7**: Frozen

Status code: 400

Table 4-44 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-45 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-46 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-47 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-48 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-49 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-50 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-51 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-52 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-53 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-54 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-55 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-56 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-57 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Disable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_info" : {
    "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "key_state" : "3"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Disable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DisableKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DisableKeyRequest request = new DisableKeyRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            DisableKeyResponse response = client.disableKey(request);
```



```
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Disable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisableKeyRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.disable_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Disable the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
```



```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisableKeyRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.DisableKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.1.4 Schedule the deletion of a key

Function

This API is used to set the number of days after which a key will be deleted. The value ranges from **7** to **1,096** days.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-58 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-59 Request header parameters

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

Table 4-60 Request body parameters

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

Response Parameters

Status code: 200

Table 4-61 Response body parameters

Parameter	Type	Description
key_id	String	Key ID
key_state	String	Key status. Possible values are as follows: **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported **7**: Frozen

Status code: 400

Table 4-62 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-63 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-64 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-65 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-66 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-67 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-68 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-69 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-70 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-71 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-72 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-73 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-74 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-75 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Delete the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** after seven days.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "pending_days" : "7"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
  "key_state" : "4"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Delete the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** after seven days.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;
```



```
public class DeleteKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteKeyRequest request = new DeleteKeyRequest();
        ScheduleKeyDeletionRequestBody body = new ScheduleKeyDeletionRequestBody();
        body.withPendingDays("7");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            DeleteKeyResponse response = client.deleteKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Delete the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** after seven days.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
```



```
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteKeyRequest()
        request.body = ScheduleKeyDeletionRequestBody(
            pending_days="7",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.delete_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Delete the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** after seven days.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteKeyRequest{}
    request.Body = &model.ScheduleKeyDeletionRequestBody{
        PendingDays: "7",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.DeleteKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.1.5 Cancel the scheduled deletion of a key

Function

This API is used to cancel the scheduled deletion of a key. Only keys that are scheduled a deletion task can be deleted by this API.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/cancel-key-deletion

Table 4-76 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-77 Request header parameters

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

Table 4-78 Request body parameters

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

Response Parameters

Status code: 200

Table 4-79 Response body parameters

Parameter	Type	Description
key_id	String	Key ID
key_state	String	Key status. Possible values are as follows: **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported **7**: Frozen

Status code: 400

Table 4-80 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-81 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-82 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-83 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-84 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-85 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-86 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-87 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-88 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-89 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-90 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-91 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-92 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-93 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Cancel the scheduled deletion task of the key whose status ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
  "key_state" : "3"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Cancel the scheduled deletion task of the key whose status ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```



```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class CancelKeyDeletionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CancelKeyDeletionRequest request = new CancelKeyDeletionRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            CancelKeyDeletionResponse response = client.cancelKeyDeletion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Cancel the scheduled deletion task of the key whose status ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
```



```
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CancelKeyDeletionRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.cancel_key_deletion(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Cancel the scheduled deletion task of the key whose status ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CancelKeyDeletionRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.CancelKeyDeletion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.1.6 Modify a key alias

Function

This API is used to modify the alias of a CMK. The aliases of the following keys cannot be modified: Service default CMKs that end with **/default** Keys that are scheduled to be deleted

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-94 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-95 Request header parameters

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

Table 4-96 Request body parameters

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

Response Parameters

Status code: 200

Table 4-97 Response body parameters

Parameter	Type	Description
key_info	**KeyAliasInfo** object	Key alias.

Table 4-98 KeyAliasInfo

Parameter	Type	Description
key_id	String	Key ID.
key_alias	String	Key alias.

Status code: 400

Table 4-99 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-100 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-101 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-102 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-103 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-104 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-105 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-106 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-107 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-108 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-109 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-110 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-111 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-112 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Change the alias of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_info" : {
```



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

SDK Sample Code

The SDK sample code is as follows.

Java

Change the alias of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class UpdateKeyAliasSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateKeyAliasRequest request = new UpdateKeyAliasRequest();
        UpdateKeyAliasRequestBody body = new UpdateKeyAliasRequestBody();
        body.withKeyAlias("test");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            UpdateKeyAliasResponse response = client.updateKeyAlias(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

Change the alias of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateKeyAliasRequest()
        request.body = UpdateKeyAliasRequestBody(
            key_alias="test",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.update_key_alias(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Change the alias of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
```



```
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateKeyAliasRequest{}
    request.Body = &model.UpdateKeyAliasRequestBody{
        KeyAlias: "test",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.UpdateKeyAlias(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.1.7 Modify CMK description

Function

This API is used to modify the CMK details. The details of the following keys cannot be modified: Service default CMKs that end with **/default** Keys that are scheduled to be deleted

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-113 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-114 Request header parameters

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

Table 4-115 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

Parameter	Mandatory	Type	Description
key_description	Yes	String	Key description. It can contain 0 to 255 characters.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-116 Response body parameters

Parameter	Type	Description
key_info	**KeyDescriptionInfo** object	Key description.

Table 4-117 KeyDescriptionInfo

Parameter	Type	Description
key_id	String	Key ID.
key_description	String	Key description.

Status code: 400

Table 4-118 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-119 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-120 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-121 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-122 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-123 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-124 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-125 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-126 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-127 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-128 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-129 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-130 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-131 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Change the description of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

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

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Change the description of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class UpdateKeyDescriptionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```



```
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateKeyDescriptionRequest request = new UpdateKeyDescriptionRequest();
        UpdateKeyDescriptionRequestBody body = new UpdateKeyDescriptionRequestBody();
        body.withKeyDescription("test");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            UpdateKeyDescriptionResponse response = client.updateKeyDescription(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Change the description of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateKeyDescriptionRequest()
        request.body = UpdateKeyDescriptionRequestBody(
            key_description="test",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.update_key_description(request)
        print(response)
```



```
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Change the description of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to **test**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateKeyDescriptionRequest{}
    request.Body = &model.UpdateKeyDescriptionRequestBody{
        KeyDescription: "test",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.UpdateKeyDescription(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.2 DEK Management

4.1.2.1 Generate a random number

Function

This API is used to generate an 8-bit to 8,192-bit random number.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/gen-random

Table 4-132 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-133 Request header parameters

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

Table 4-134 Request body parameters

Parameter	Mandatory	Type	Description
random_data_length	Yes	String	Bit length of a random number. The value must be a multiple of 8 and ranges from **8** to **8192**, for example, **512**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-135 Response body parameters

Parameter	Type	Description
random_data	String	Random number in hexadecimal format. Two characters represent 1 byte. The length of the random number must be the same as that of the user input parameter **random_data_length**.

Status code: 400

Table 4-136 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-137 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-138 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-139 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-140 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-141 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-142 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-143 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-144 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-145 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-146 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-147 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-148 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-149 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Generate a 512-bit random number.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "random_data" : "5791C223E87120BE4B98D168F47A58BB2A88834EEADC"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Generate a 512-bit random number.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class CreateRandomSolution {

    public static void main(String[] args) {
```



```
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateRandomRequest request = new CreateRandomRequest();
        GenRandomRequestBody body = new GenRandomRequestBody();
        body.withRandomDataLength("512");
        request.withBody(body);
        try {
            CreateRandomResponse response = client.createRandom(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Generate a 512-bit random number.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateRandomRequest()
        request.body = GenRandomRequestBody(
            random_data_length="512"
```



```
        )
        response = client.create_random(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Generate a 512-bit random number.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateRandomRequest{}
    request.Body = &model.GenRandomRequestBody{
        RandomDataLength: "512",
    }
    response, err := client.CreateRandom(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.2.2 Create a DEK

Function

This API is used to create a DEK. The returned result contains both plaintext and ciphertext.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-150 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-151 Request header parameters

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

Table 4-152 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
key_spec	No	String	Length of generated keys. Unit: bit. The value can be **AES_256** or **AES_128**. **AES_256**: a 256-bit symmetric key **AES_128**: a 128-bit symmetric key Note: You must configure either **datakey_length** or **key_spec**. If both are empty, a 256-bit key is generated by default. If both are specified, only **datakey_length** takes effect.
datakey_length	No	String	Key length by bit. The value ranges from **8** to **8,192** and must be a multiple of 8. Note: You must configure either **datakey_length** or **key_spec**. If both are empty, a 256-bit key is generated by default. If both are specified, only **datakey_length** takes effect.

Parameter	Mandatory	Type	Description
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-153 Response body parameters

Parameter	Type	Description
key_id	String	Key ID.
plain_text	String	Plaintext DEK in hexadecimal format. Two characters represent 1 byte.
cipher_text	String	Ciphertext DEK in hexadecimal format. Two characters represent 1 byte.

Status code: 400

Table 4-154 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-155 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-156 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-157 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-158 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-159 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-160 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-161 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-162 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-163 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-164 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-165 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-166 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-167 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "plain_text" : "8151014275E426C72EE7D44267XXXXX...",
  "cipher_text" : "020098009EEAFCE122CAA5927D2XXX..."
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class CreateDatakeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDatakeyRequest request = new CreateDatakeyRequest();
        CreateDatakeyRequestBody body = new CreateDatakeyRequestBody();
        body.withDatakeyLength("512");
```



```
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            CreateDatakeyResponse response = client.createDatakey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDatakeyRequest()
        request.body = CreateDatakeyRequestBody(
            datakey_length="512",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.create_datakey(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.


```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDatakeyRequest{}
    datakeyLengthCreateDatakeyRequestBody:= "512"
    request.Body = &model.CreateDatakeyRequestBody{
        DatakeyLength: &datakeyLengthCreateDatakeyRequestBody,
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.CreateDatakey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.2.3 Create a plaintext-free DEK

Function

This API is used to create a data encryption key (DEK). The returned result contains only ciphertext.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-168 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-169 Request header parameters

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

Table 4-170 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
key_spec	No	String	Length of generated keys. Unit: bit. The value can be **AES_256** or **AES_128**. **AES_256**: a 256-bit symmetric key **AES_128**: a 128-bit symmetric key Note: You must configure either **datakey_length** or **key_spec**. If both are empty, a 256-bit key is generated by default. If both are specified, only **datakey_length** takes effect.
datakey_length	No	String	Key length by bit. The value ranges from **8** to **8,192** and must be a multiple of 8. Note: You must configure either **datakey_length** or **key_spec**. If both are empty, a 256-bit key is generated by default. If both are specified, only **datakey_length** takes effect.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-171 Response body parameters

Parameter	Type	Description
key_id	String	Key ID.
cipher_text	String	Encrypted ciphertext in Base64 format

Status code: 400

Table 4-172 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-173 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-174 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-175 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-176 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-177 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-178 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-179 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-180 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-181 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-182 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-183 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-184 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-185 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "cipher_text" : "020098009EEAFCE122CAA5927D2XXX..."
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class CreateDatakeyWithoutPlaintextSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDatakeyWithoutPlaintextRequest request = new CreateDatakeyWithoutPlaintextRequest();
        CreateDatakeyRequestBody body = new CreateDatakeyRequestBody();
        body.withDatakeyLength("512");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            CreateDatakeyWithoutPlaintextResponse response = client.createDatakeyWithoutPlaintext(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
```



```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDatakeyWithoutPlaintextRequest()
        request.body = CreateDatakeyRequestBody(
            datakey_length="512",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.create_datakey_without_plaintext(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Create a 512-bit plaintext DEK and encrypt it using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**. The plaintext key is not returned.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDatakeyWithoutPlaintextRequest{}
```



```
    datakeyLengthCreateDatakeyRequestBody:= "512"
    request.Body = &model.CreateDatakeyRequestBody{
        DatakeyLength: &datakeyLengthCreateDatakeyRequestBody,
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.CreateDatakeyWithoutPlaintext(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.2.4 Encrypt a DEK

Function

This API is used to encrypt a DEK using a specified CMK.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-186 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-187 Request header parameters

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

Table 4-188 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
plain_text	Yes	String	Plaintext and encrypted DEKs. If CMK uses the AES algorithm, the value is a combination of a plaintext DEK and a 32-byte SHA256-encrypted DEK. If CMK uses the SM4 algorithm, the value is combination of a plaintext DEK and a 32-byte SM3-encrypted DEK. The values are 32-byte hexadecimal strings.

Parameter	Mandatory	Type	Description
datakey_plain_length	Yes	String	Number of bytes of a DEK in plaintext. The value ranges from **1** to **1024**, for example, **64**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-189 Response body parameters

Parameter	Type	Description
key_id	String	Key ID
cipher_text	String	Ciphertext DEK in hexadecimal format. Two characters represent 1 byte.
datakey_length	String	Length of a DEK, in bytes.

Status code: 400

Table 4-190 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-191 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-192 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-193 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-194 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-195 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-196 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-197 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-198 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-199 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-200 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-201 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-202 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-203 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Example Requests

Encrypt a 512-bit plaintext DEK using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "plain_text" :
"7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4b18ff0512
525e527b10331100f357bf42125d8d5ced94ffbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0a9ee0305f
b9540f797",
  "datakey_plain_length" : "64"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "datakey_length" : "64",
  "cipher_text" : "020098009EEAFCE122CAA5927D2XXX..."
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Encrypt a 512-bit plaintext DEK using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class EncryptDatakeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```



```
        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        EncryptDatakeyRequest request = new EncryptDatakeyRequest();
        EncryptDatakeyRequestBody body = new EncryptDatakeyRequestBody();
        body.withDatakeyPlainLength("64");

body.withPlainText("7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e
5efe74e4b18ff0512525e527b10331100f357bf42125d8d5ced94ffbc8ac72b0785ca7fe33eb6776ce3990b11e32
b299d9c0a9ee0305fb9540f797");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            EncryptDatakeyResponse response = client.encryptDatakey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Encrypt a 512-bit plaintext DEK using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = EncryptDatakeyRequest()
        request.body = EncryptDatakeyRequestBody(
            datakey_plain_length="64",
```



```
plain_text="7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4
b18ff0512525e527b10331100f357bf42125d8d5ced94ffbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0
a9ee0305fb9540f797",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.encrypt_datakey(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Encrypt a 512-bit plaintext DEK using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.EncryptDatakeyRequest{}
    request.Body = &model.EncryptDatakeyRequestBody{
        DatakeyPlainLength: "64",
        PlainText:
"7549d9aea901767bf3c0b3e14b10722eaf6f59053bbd82045d04e075e809a0fe6ccab48f8e5efe74e4b18ff0512
525e527b10331100f357bf42125d8d5ced94ffbc8ac72b0785ca7fe33eb6776ce3990b11e32b299d9c0a9ee0305f
b9540f797",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.EncryptDatakey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.2.5 Decrypt a DEK

Function

This API is used to decrypt DEKs using specified CMKs.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-204 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-205 Request header parameters

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

Table 4-206 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
cipher_text	Yes	String	Hexadecimal string of the DEK ciphertext and the metadata. It is the value of cipher_text in the encryption result.
datakey_cipher_length	Yes	String	Number of bytes of a key. The value ranges from **1** to **1024**, for example, **64**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-207 Response body parameters

Parameter	Type	Description
data_key	String	Hexadecimal string of the plaintext of a DEK

Parameter	Type	Description
datakey_length	String	Length of a plaintext DEK, in bytes.
datakey_dgst	String	Hexadecimal string corresponding to the SHA-256 hash value of the plaintext of a DEK.

Status code: 400

Table 4-208 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-209 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-210 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-211 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-212 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-213 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-214 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-215 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-216 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-217 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-218 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-219 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-220 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-221 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Decrypt a ciphertext DEK into a 64-byte plaintext DEK using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "cipher_text" : "020098005273E14E6E8E95F5463BECDC27E80AFxxxxxxxxxx...",
  "datakey_cipher_length" : "64"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "data_key" : "000000e724d9cb35df84bb474a37fXXX...",
```



```
  "datakey_length" : "64",
  "datakey_dgst" : "F5A5FD42D16A20302798EF6ED3099XXX..."
}
```

Status Codes

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

Error Codes

See **Error Codes**.

4.1.3 Small-Size Data Encryption and Decryption

4.1.3.1 Encrypt data

Function

This API is used to encrypt data using a specified CMK.

Constraints

If you use an asymmetric key for encryption, record the key ID and encryption algorithm, which are required for decryption. If the specified key ID and encryption algorithm do not match those used for encrypting data, the decryption fails. If you use a symmetric key for decryption, you do not need to provide the key ID and encryption algorithm. KMS stores the information as ciphertext. KMS cannot store metadata in the ciphertext generated using an asymmetric key. A standard asymmetric key ciphertext does not contain configurable fields.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-222 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-223 Request header parameters

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

Table 4-224 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
plain_text	Yes	String	Plaintext data. It can be 1 to 4,096 bytes and should match the regular expression ^.{1,4096}$. After it is converted to a byte array, its length should still be 1 to 4096 bytes.

Parameter	Mandatory	Type	Description
encryption_algorithm	No	String	Data encryption algorithm. Specify this parameter if only asymmetric keys are used. The default value is **SYMMETRIC_DEFAULT**. Possible values are as follows: **SYMMETRIC_DEFAULT** **RSAES_OAEP_SHA_256** **SM2_ENCRYPT**
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-225 Response body parameters

Parameter	Type	Description
key_id	String	Key ID.
cipher_text	String	Encrypted ciphertext in Base64 format

Status code: 400

Table 4-226 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-227 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-228 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-229 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-230 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-231 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-232 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-233 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-234 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-235 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-236 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-237 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-238 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-239 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Example Requests

Encrypt plaintext data **hello world** using the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **SYMMETRIC_DEFAULT** encryption algorithm.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "plain_text" : "hello world",
  "encryption_algorithm" : "SYMMETRIC_DEFAULT"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
  "cipher_text" : "AgDoAG7EsEc2OHpQxz4gDFDH54CqwaelpTdEl+RFXXX..."
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Encrypt plaintext data **hello world** using the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **SYMMETRIC_DEFAULT** encryption algorithm.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class EncryptDataSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```



```
        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        EncryptDataRequest request = new EncryptDataRequest();
        EncryptDataRequestBody body = new EncryptDataRequestBody();

body.withEncryptionAlgorithm(EncryptDataRequestBody.EncryptionAlgorithmEnum.fromValue("SYMMETRIC
_DEFAULT"));
        body.withPlainText("hello world");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            EncryptDataResponse response = client.encryptData(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Encrypt plaintext data **hello world** using the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **SYMMETRIC_DEFAULT** encryption algorithm.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = EncryptDataRequest()
        request.body = EncryptDataRequestBody(
            encryption_algorithm="SYMMETRIC_DEFAULT",
            plain_text="hello world",
```



```
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.encrypt_data(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Encrypt plaintext data **hello world** using the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **SYMMETRIC_DEFAULT** encryption algorithm.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.EncryptDataRequest{}
    encryptionAlgorithmEncryptDataRequestBody:= model.GetEncryptDataRequestBodyEncryptionAlgorithmEnum().SYMMETRIC_DEFAULT
    request.Body = &model.EncryptDataRequestBody{
        EncryptionAlgorithm: &encryptionAlgorithmEncryptDataRequestBody,
        PlainText: "hello world",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.EncryptData(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.3.2 Decrypt data

Function

This API is used to decrypt data.

Constraints

When decrypting the data encrypted using asymmetric keys, you need to specify the key ID and encryption algorithm. If the specified key ID and encryption algorithm do not match those used for encrypting data, the decryption fails.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-240 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-241 Request header parameters

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

Table 4-242 Request body parameters

Parameter	Mandatory	Type	Description
cipher_text	Yes	String	Ciphertext of encrypted data, which is the value of **cipher_text** in the data encryption output. The value matches the regular expression **^[0-9a-zA-Z+/=]{128,5648}$**.
encryption_algorithm	No	String	Data encryption algorithm. Specify this parameter if only asymmetric keys are used. The default value is **SYMMETRIC_DEFAULT**. Possible values are as follows: **SYMMETRIC_DEFAULT** **RSAES_OAEP_SHA_256** **SM2_ENCRYPT**
key_id	No	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-243 Response body parameters

Parameter	Type	Description
key_id	String	Key ID.
plain_text	String	Plaintext.
plain_text_base64	String	Base64 value of the plaintext. In asymmetric encryption scenarios, if the encrypted plaintext contains invisible characters, the value is used as the decryption result.

Status code: 400

Table 4-244 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-245 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-246 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-247 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-248 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-249 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-250 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-251 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-252 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-253 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-254 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-255 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-256 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-257 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Decrypt ciphertext **AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx** that was encrypted using the **SYMMETRIC_DEFAULT** algorithm.

```
{
  "cipher_text" : "AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx",
```



```
  "encryption_algorithm" : "SYMMETRIC_DEFAULT"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
  "plain_text" : "hello world",
  "plain_text_base64" : "aGVsbG8gd29ybGQ="
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Decrypt ciphertext **AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx** that was encrypted using the **SYMMETRIC_DEFAULT** algorithm.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DecryptDataSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DecryptDataRequest request = new DecryptDataRequest();
        DecryptDataRequestBody body = new DecryptDataRequestBody();

body.withEncryptionAlgorithm(DecryptDataRequestBody.EncryptionAlgorithmEnum.fromValue("SYMMETRIC_DEFAULT"));
        body.withCipherText("AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx");
        request.withBody(body);
        try {
            DecryptDataResponse response = client.decryptData(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```



```
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Decrypt ciphertext **AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx** that was encrypted using the **SYMMETRIC_DEFAULT** algorithm.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DecryptDataRequest()
        request.body = DecryptDataRequestBody(
            encryption_algorithm="SYMMETRIC_DEFAULT",
            cipher_text="AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx"
        )
        response = client.decrypt_data(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Decrypt ciphertext **AgDoAG7EsEc2OHpQxz4gDFDH54Cqwaelxxxxxxx** that was encrypted using the **SYMMETRIC_DEFAULT** algorithm.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
```



```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DecryptDataRequest{}
    encryptionAlgorithmDecryptDataRequestBody:= model.GetDecryptDataRequestBodyEncryptionAlgorithmEnum().SYMMETRIC_DEFAULT
    request.Body = &model.DecryptDataRequestBody{
        EncryptionAlgorithm: &encryptionAlgorithmDecryptDataRequestBody,
        CipherText: "AgDoAG7EsEc2OHpQxz4gDFDH54CqwaelxxxxxxX",
    }
    response, err := client.DecryptData(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Status Code	Description
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.1.4 Signature and Verification

4.1.4.1 Signing Data

Function

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

Constraints

Only the asymmetric key whose **key_usage** is **SIGN_VERIFY** can be used for signature. SM2 keys can only be used to sign message digests.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-258 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-259 Request header parameters

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

Table 4-260 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
message	Yes	String	Message digest or message to be signed. The message must be Base64-coded and smaller than 4,096 bytes.
signing_algorithm	Yes	String	Signature algorithm. Possible values are as follows: **RSASSA_PSS_SHA_256 RSASSA_PSS_SHA_384 RSASSA_PSS_SHA_512 RSASSA_PKCS1_V1_5_SHA_256 RSASSA_PKCS1_V1_5_SHA_384 RSASSA_PKCS1_V1_5_SHA_512 ECDSA_SHA_256 ECDSA_SHA_384 ECDSA_SHA_512 SM2DSA_SM3**
message_type	No	String	Message type. The default value is **DIGEST**. Possible values are as follows: **DIGEST**: message digest **RAW**: message

Parameter	Mandatory	Type	Description
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-261 Response body parameters

Parameter	Type	Description
key_id	String	Key ID
signature	String	Base64-coded signature value

Status code: 400

Table 4-262 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-263 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-264 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-265 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-266 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-267 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-268 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-269 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-270 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-271 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-272 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-273 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-274 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-275 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Sign messages and digests using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **RSASSA_PKCS1_V1_5_SHA_256** algorithm.

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

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Sign messages and digests using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **RSASSA_PKCS1_V1_5_SHA_256** algorithm.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class SignSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        SignRequest request = new SignRequest();
        SignRequestBody body = new SignRequestBody();
```



```
body.withSigningAlgorithm(SignRequestBody.SigningAlgorithmEnum.fromValue("RSASSA_PKCS1_V1_5_SHA_
256"));
        body.withMessage("MmFiZWE0ZjI3ZGIxYTkzY2RmYmEzM2YwMTA1YmJjYw==");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            SignResponse response = client.sign(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Sign messages and digests using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **RSASSA_PKCS1_V1_5_SHA_256** algorithm.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SignRequest()
        request.body = SignRequestBody(
            signing_algorithm="RSASSA_PKCS1_V1_5_SHA_256",
            message="MmFiZWE0ZjI3ZGIxYTkzY2RmYmEzM2YwMTA1YmJjYw==",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.sign(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

Sign messages and digests using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **RSASSA_PKCS1_V1_5_SHA_256** algorithm.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SignRequest{}
    request.Body = &model.SignRequestBody{
        SigningAlgorithm: model.GetSignRequestBodySigningAlgorithmEnum().RSASSA_PKCS1_V1_5_SHA_256,
        Message: "MmFiZWE0ZjI3ZGIxYTkzY2RmYmEzM2YwMTA1YmJjYw==",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.Sign(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.5 Key Tag Management

4.1.5.1 Query key instances

Function

This API is used to query a key instance. You can use tags to filter and check CMK details.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/{resource_instances}/action

Table 4-276 Path Parameters

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

Request Parameters

Table 4-277 Request header parameters

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

Table 4-278 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of records to be queried. You do not need to specify this parameter if **action** is set to **count**. If **action** is set to **filter**, the default value of this parameter is **10**. The value of **limit** ranges from **1** to **1000**.
offset	No	String	Index location. The query starts from the next data specified by **offset**. If **action** is set to **count**, do not specify this parameter. If **action** is set to **filter**, the default value of **offset** is **0**. The value of **offset** must be a non-negative number.
action	No	String	Operation type. The value can be **filter** or **count**. **filter**: filtering **count**: total number of queried records

Parameter	Mandatory	Type	Description
tags	No	Array of **Tag** objects	Tag list, which is the value pairs of **key** and **value**. **key**: Tag key. A tag key can contain up to 10 values. This parameter cannot be left empty or repeated. The value of a **key** must be unique and contain up to 36 characters. **value**: Tag value. There can be multiple values and each value can contain up to 43 characters. Each pair contains one **key** and one **value**.
matches	No	Array of **TagItem** objects	Search field. Possible values are as follows: **key**: The field to be matched, for example, **resource_name**. **value**: The value to be matched. The value can contain up to 255 characters and cannot be blank.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Table 4-279 Tag

Parameter	Mandatory	Type	Description
key	No	String	Key. The value can contain up to 36 unicode characters. This parameter cannot be left empty. The following nonprintable characters are not supported: ASCII(0-31), *, <, >, , and =.
values	No	Array of strings	Tag value set.

Table 4-280 TagItem

Parameter	Mandatory	Type	Description
key	Yes	String	Key. The value can contain up to 36 unicodes. This parameter cannot be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.
value	No	String	Key. The value can contain up to 43 unicodes. This parameter can be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.

Response Parameters

Status code: 200

Table 4-281 Response body parameters

Parameter	Type	Description
resources	Array of **ActionResources** objects	Resource instance list. For details, see the data structure description of the **resource** field.
total_count	Integer	Total number of records.

Table 4-282 ActionResources

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

Table 4-283 KeyDetails

Parameter	Type	Description
key_id	String	Key ID.
domain_id	String	User domain ID.
key_alias	String	Key alias.
realm	String	Key realm.
key_spec	String	Key generation algorithm. Possible values are as follows: **AES_256 SM4 RSA_2048 RSA_3072 RSA_4096 EC_P256 EC_P384 SM2**
key_usage	String	Key usage. Possible values are as follows: **ENCRYPT_DECRYPT SIGN_VERIFY**
key_description	String	Key description.
creation_date	String	Time when the key was created. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_deletion_date	String	Time when the key was scheduled to be deleted. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
key_state	String	Key status which matches the regular expression **^[1-5]{1}$**. Possible values are as follows: **1**: To be activated **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported
default_key_flag	String	Master key identifier. The value is 1 for Default Master Keys and 0 for non-default master keys.
key_type	String	Key type.
expiration_time	String	Time when the key material expires. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
origin	String	Key source. The default value is **kms**. Possible values are as follows: **kms**: The key is generated by KMS. **external**: The key is imported.
key_rotation_enabled	String	Key rotation status. The default value is false, indicating that key rotation is disabled.

Parameter	Type	Description
sys_enterprise_project_id	String	Enterprise project ID. The default value is **0**. If you have created an enterprise project, the resources are listed in the default enterprise project. If you haven't created an enterprise project, the resources are not listed in the enterprise project.
keystore_id	String	Keystore ID

Table 4-284 TagItem

Parameter	Type	Description
key	String	Key. The value can contain up to 36 unicodes. This parameter cannot be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.
value	String	Key. The value can contain up to 43 unicodes. This parameter can be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.

Status code: 400

Table 4-285 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-286 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-287 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-288 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-289 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-290 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-291 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-292 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-293 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-294 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-295 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-296 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-297 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-298 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Example Requests

Query the key instance whose tag key is **key1** and tag value is **value1** or **value2**. The start position is **100** and 100 results are displayed.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "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,
      "key_rotation_enabled" : false,
      "expiration_time" : "",
      "origin" : "kms",
      "sys_enterprise_project_id" : "0",
      "realm" : "test"
    } ],
    "resource_name" : "tagTest_xmdmi",
    "tags" : [ {
      "key" : "key",
      "value" : "testValue!"
    }, {
      "key" : "haha",
      "value" : "testValue"
    } ]
  } ],
  "total_count" : 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Query the key instance whose tag key is **key1** and tag value is **value1** or **value2**. The start position is **100** and 100 results are displayed.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ListKmsByTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKmsByTagsRequest request = new ListKmsByTagsRequest();
        ListKmsByTagsRequestBody body = new ListKmsByTagsRequestBody();
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("value1");
        listTagsValues.add("value2");
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        body.withTags(listbodyTags);
        body.withAction("filter");
        body.withOffset("100");
        body.withLimit("100");
        request.withBody(body);
        try {
            ListKmsByTagsResponse response = client.listKmsByTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```



```
    }
}
```

Python

Query the key instance whose tag key is **key1** and tag value is **value1** or **value2**. The start position is **100** and 100 results are displayed.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKmsByTagsRequest()
        listValuesTags = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            Tag(
                key="key1",
                values=listValuesTags
            )
        ]
        request.body = ListKmsByTagsRequestBody(
            tags=listTagsbody,
            action="filter",
            offset="100",
            limit="100"
        )
        response = client.list_kms_by_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Query the key instance whose tag key is **key1** and tag value is **value1** or **value2**. The start position is **100** and 100 results are displayed.

```
package main

import (
    "fmt"
```



```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKmsByTagsRequest{}
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    keyTags:= "key1"
    var listTagsbody = []model.Tag{
        {
            Key: &keyTags,
            Values: &listValuesTags,
        },
    }
    actionListKmsByTagsRequestBody:= "filter"
    offsetListKmsByTagsRequestBody:= "100"
    limitListKmsByTagsRequestBody:= "100"
    request.Body = &model.ListKmsByTagsRequestBody{
        Tags: &listTagsbody,
        Action: &actionListKmsByTagsRequestBody,
        Offset: &offsetListKmsByTagsRequestBody,
        Limit: &limitListKmsByTagsRequestBody,
    }
    response, err := client.ListKmsByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.5.2 Query key tags

Function

This API is used to query key tags.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1.0/{project_id}/kms/{key_id}/tags

Table 4-299 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	Key ID

Request Parameters

Table 4-300 Request header parameters

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

Response Parameters

Status code: 200

Table 4-301 Response body parameters

Parameter	Type	Description
tags	Array of **TagItem** objects	Tag list, which is the value pairs of **key** and **value**. **key**: Tag key. A tag key can contain up to 10 values. This parameter cannot be left empty or repeated. The value of a **key** must be unique and contain up to 36 characters. **value**: Tag value. There can be multiple values and each value can contain up to 43 characters. Each pair contains one **key** and one **value**.
existTagsNum	Integer	Number of key tags

Table 4-302 TagItem

Parameter	Type	Description
key	String	Key. The value can contain up to 36 unicodes. This parameter cannot be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.
value	String	Key. The value can contain up to 43 unicodes. This parameter can be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.

Status code: 400

Table 4-303 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-304 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-305 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-306 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-307 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-308 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-309 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-310 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-311 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-312 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-313 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-314 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-315 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-316 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ],
  "existTagsNum" : 2
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```



```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowKmsTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowKmsTagsRequest request = new ShowKmsTagsRequest();
        try {
            ShowKmsTagsResponse response = client.showKmsTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
    try:
        request = ShowKmsTagsRequest()
        response = client.show_kms_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKmsTagsRequest{}
    response, err := client.ShowKmsTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameters.

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

Error Codes

See **Error Codes**.

4.1.5.3 Add tags to a key

Function

This API is used to add a key tag.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/{key_id}/tags

Table 4-317 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	Key ID

Request Parameters

Table 4-318 Request header parameters

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

Table 4-319 Request body parameters

Parameter	Mandatory	Type	Description
tag	No	**TagItem** object	Tag.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Table 4-320 TagItem

Parameter	Mandatory	Type	Description
key	Yes	String	Key. The value can contain up to 36 unicodes. This parameter cannot be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.
value	No	String	Key. The value can contain up to 43 unicodes. This parameter can be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.

Response Parameters

Status code: 400

Table 4-321 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-322 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-323 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-324 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-325 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-326 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-327 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-328 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-329 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-330 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-331 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-332 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-333 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-334 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

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

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
```



```
import com.huaweicloud.sdk.kms.v2.model.*;


public class CreateKmsTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateKmsTagRequest request = new CreateKmsTagRequest();
        CreateKmsTagRequestBody body = new CreateKmsTagRequestBody();
        TagItem tagbody = new TagItem();
        tagbody.withKey("DEV")
            .withValue("DEV1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateKmsTagResponse response = client.createKmsTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

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

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)
```



```
    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateKmsTagRequest()
        tagbody = TagItem(
            key="DEV",
            value="DEV1"
        )
        request.body = CreateKmsTagRequestBody(
            tag=tagbody
        )
        response = client.create_kms_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

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

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateKmsTagRequest{}
    valueTag:= "DEV1"
    tagbody := &model.TagItem{
        Key: "DEV",
        Value: &valueTag,
    }
    request.Body = &model.CreateKmsTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateKmsTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```



```
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.5.4 Query project tags

Function

This API is used to query all tags in a specified project.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1.0/{project_id}/kms/tags

Table 4-335 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-336 Request header parameters

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

Response Parameters

Status code: 200

Table 4-337 Response body parameters

Parameter	Type	Description
tags	Array of **Tag** objects	Tag list, which is the value pairs of **key** and **value**. **key**: Tag key. A tag key can contain up to 10 values. This parameter cannot be left empty or repeated. The value of a **key** must be unique and contain up to 36 characters. **value**: Tag value. There can be multiple values and each value can contain up to 43 characters. Each pair contains one **key** and one **value**.

Table 4-338 Tag

Parameter	Type	Description
key	String	Key. The value can contain up to 36 unicode characters. This parameter cannot be left empty. The following nonprintable characters are not supported: ASCII(0-31), *, <, >, , and =.
values	Array of strings	Tag value set.

Status code: 400

Table 4-339 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-340 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-341 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-342 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-343 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-344 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-345 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-346 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-347 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-348 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-349 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-350 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-351 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-352 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```



```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ListKmsTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKmsTagsRequest request = new ListKmsTagsRequest();
        try {
            ListKmsTagsResponse response = client.listKmsTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
```



```
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKmsTagsRequest()
        response = client.list_kms_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKmsTagsRequest{}
    response, err := client.ListKmsTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.

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

Error Codes

See **Error Codes**.

4.1.5.5 Batch add or delete key tags

Function

This API is used to add or delete key tags in batches.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/{key_id}/tags/action

Table 4-353 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	Key ID

Request Parameters

Table 4-354 Request header parameters

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

Table 4-355 Request body parameters

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

Table 4-356 TagItem

Parameter	Mandatory	Type	Description
key	Yes	String	Key. The value can contain up to 36 unicodes. This parameter cannot be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.
value	No	String	Key. The value can contain up to 43 unicodes. This parameter can be left empty. The following nonprintable characters cannot be included: ASCII(0-31), *, <, >, , and =.

Response Parameters

Status code: 400

Table 4-357 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-358 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-359 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-360 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-361 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-362 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-363 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-364 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-365 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-366 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-367 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-368 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-369 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-370 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Add multiple key tags. For tag 1, the key is **key1** and the value is **value1**. For tag 2, the key is **key** and the value is **value3**.

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Add multiple key tags. For tag 1, the key is **key1** and the value is **value1**. For tag 2, the key is **key** and the value is **value3**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```



```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateKmsTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateKmsTagsRequest request = new BatchCreateKmsTagsRequest();
        BatchCreateKmsTagsRequestBody body = new BatchCreateKmsTagsRequestBody();
        List<TagItem> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagItem()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new TagItem()
                .withKey("key")
                .withValue("value3")
        );
        body.withAction("create");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateKmsTagsResponse response = client.batchCreateKmsTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Add multiple key tags. For tag 1, the key is **key1** and the value is **value1**. For tag 2, the key is **key** and the value is **value3**.


```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateKmsTagsRequest()
        listTagsbody = [
            TagItem(
                key="key1",
                value="value1"
            ),
            TagItem(
                key="key",
                value="value3"
            )
        ]
        request.body = BatchCreateKmsTagsRequestBody(
            action="create",
            tags=listTagsbody
        )
        response = client.batch_create_kms_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Add multiple key tags. For tag 1, the key is **key1** and the value is **value1**. For tag 2, the key is **key** and the value is **value3**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```



```
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateKmsTagsRequest{}
    valueTags:= "value1"
    valueTags1:= "value3"
    var listTagsbody = []model.TagItem{
        {
            Key: "key1",
            Value: &valueTags,
        },
        {
            Key: "key",
            Value: &valueTags1,
        },
    }
    request.Body = &model.BatchCreateKmsTagsRequestBody{
        Action: "create",
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateKmsTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Status Code	Description
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.1.5.6 Delete key tags

Function

This API is used to delete a key tag.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-371 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
key_id	Yes	String	Key ID
key	Yes	String	Value of a tag key.

Request Parameters

Table 4-372 Request header parameters

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

Response Parameters

Status code: 400

Table 4-373 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-374 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-375 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-376 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-377 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-378 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-379 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-380 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-381 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-382 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-383 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-384 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-385 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-386 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```



```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DeleteTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteTagRequest request = new DeleteTagRequest();
        try {
            DeleteTagResponse response = client.deleteTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
    try:
        request = DeleteTagRequest()
        response = client.delete_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteTagRequest{}
    response, err := client.DeleteTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	No Content
400	Invalid request parameters.

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

Error Codes

See **Error Codes**.

4.1.6 Key API Version Query

4.1.6.1 Query a version

Function

This API is used to query a specified API version.

Calling Method

For details, see **Calling APIs**.

URI

GET /{version_id}

Table 4-387 Path Parameters

Parameter	Mandatory	Type	Description
version_id	Yes	String	API version.

Request Parameters

None

Response Parameters

Status code: 200

Table 4-388 Response body parameters

Parameter	Type	Description
version	Object	Version object list. For details, see the data structure description of the **ApiVersionDetail** field.

Table 4-389 ApiVersionDetail

Parameter	Type	Description
id	String	Version number, for example, v1.0.
links	Array of **ApiLink** objects	JSON object. For details, see the data structure of the **links** field.
version	String	If the APIs of this version support microversions, the supported maximum microversion is returned. If microversions are not supported, an empty string is returned.
status	String	Version status. Possible values are as follows: **CURRENT**: The version is widely used. **SUPPORTED**: It is an earlier version which can still be used. **DEPRECATED**: The version is deprecated and may be deleted later.
updated	String	Version release time. The time must be in UTC format. For example, v1 is released on **2014-06-28T12:20:21Z**.
min_version	String	If the APIs of this version support microversions, the supported minimum microversion is returned. If microversions are not supported, an empty string is returned.

Table 4-390 ApiLink

Parameter	Type	Description
href	String	API URL.
rel	String	The default value is self.

Status code: 500

Table 4-391 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-392 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-393 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-394 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-395 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-396 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the V1.0 API information of the current site.

```
/v1.0
```

Example Responses

Status code: 200

A specified API version is queried.

```
{
  "version" : {
    "min_version" : "",
    "links" : [ {
      "rel" : "self",
      "href" : "https://kms.region_id.domain.com/v1.0/"
    } ],
    "id" : "v1.0",
    "version" : "",
    "updated" : "2016-10-29T02:00:00Z",
    "status" : "CURRENT"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowVersionRequest request = new ShowVersionRequest();
        try {
            ShowVersionResponse response = client.showVersion(request);
```



```
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowVersionRequest()
        response = client.show_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```



```
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowVersionRequest{}
    response, err := client.ShowVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	A specified API version is queried.
500	Internal service error.
502	Failed to complete the request. The server receives an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.1.7 Querying Key API Versions

4.1.7.1 Query version list

Function

This API is used to query the API version list.

Calling Method

For details, see **Calling APIs**.

URI

GET /

Request Parameters

None

Response Parameters

Status code: 200

Table 4-397 Response body parameters

Parameter	Type	Description
versions	Array of **ApiVersionDetail** objects	Related version object list. For details, see the data structure description of the **versions** field.

Table 4-398 ApiVersionDetail

Parameter	Type	Description
id	String	Version number, for example, v1.0.
links	Array of **ApiLink** objects	JSON object. For details, see the data structure of the **links** field.
version	String	If the APIs of this version support microversions, the supported maximum microversion is returned. If microversions are not supported, an empty string is returned.
status	String	Version status. Possible values are as follows: **CURRENT**: The version is widely used. **SUPPORTED**: It is an earlier version which can still be used. **DEPRECATED**: The version is deprecated and may be deleted later.
updated	String	Version release time. The time must be in UTC format. For example, v1 is released on **2014-06-28T12:20:21Z**.
min_version	String	If the APIs of this version support microversions, the supported minimum microversion is returned. If microversions are not supported, an empty string is returned.

Table 4-399 ApiLink

Parameter	Type	Description
href	String	API URL.
rel	String	The default value is self.

Status code: 500

Table 4-400 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-401 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-402 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-403 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-404 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-405 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "versions" : [ {
    "min_version" : "",
    "links" : [ {
      "rel" : "self",
      "href" : "https://kms.region_id.domain.com/v1.0/"
    } ],
    "id" : "v1.0",
    "version" : "",
    "updated" : "2016-10-29T02:00:00Z",
    "status" : "CURRENT"
  } ]
}
```

Status Codes

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

Error Codes

See **Error Codes**.

4.1.8 Key Import Management

4.1.8.1 Obtain parameters for importing a key

Function

This API is used to obtain the parameters required for importing keys, including import passwords and encryption public keys. An RSA_2048 public key is returned by default.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-406 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-407 Request header parameters

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

Table 4-408 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
wrapping_algorithm	Yes	String	Key material encryption algorithm. Possible values are as follows: **RSAES_OAEP_SHA_256** **SM2_ENCRYPT**. Some sites cannot be imported using this algorithm.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-409 Response body parameters

Parameter	Type	Description
key_id	String	Key ID.
import_token	String	Key import token.
expiration_time	Long	Import parameter expiration time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
public_key	String	Public key of the DEK material, in Base64 format.

Status code: 400

Table 4-410 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-411 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-412 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-413 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-414 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-415 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-416 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-417 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-418 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-419 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-420 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-421 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-422 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-423 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Obtain the wrapping materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** using the **RSAES_OAEP_SHA_256** algorithm.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "wrapping_algorithm" : "RSAES_OAEP_SHA_256"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
  "import_token" : "AACIBjY2ZTQxYjBmLTY3ZWItNDU4Ny04OTIxLWVhZXXX...",
  "expiration_time" : 1501578672,
  "public_key" : "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCXXX..."
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Obtain the wrapping materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** using the **RSAES_OAEP_SHA_256** algorithm.


```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class CreateParametersForImportSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateParametersForImportRequest request = new CreateParametersForImportRequest();
        GetParametersForImportRequestBody body = new GetParametersForImportRequestBody();

body.withWrappingAlgorithm(GetParametersForImportRequestBody.WrappingAlgorithmEnum.fromValue("RSAES_OAEP_SHA_256"));
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            CreateParametersForImportResponse response = client.createParametersForImport(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Obtain the wrapping materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** using the **RSAES_OAEP_SHA_256** algorithm.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
```



```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateParametersForImportRequest()
        request.body = GetParametersForImportRequestBody(
            wrapping_algorithm="RSAES_OAEP_SHA_256",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.create_parameters_for_import(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Obtain the wrapping materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** using the **RSAES_OAEP_SHA_256** algorithm.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateParametersForImportRequest{}
    request.Body = &model.GetParametersForImportRequestBody{
        WrappingAlgorithm:
model.GetGetParametersForImportRequestBodyWrappingAlgorithmEnum().RSAES_OAEP_SHA_256,
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
```



```
    }
    response, err := client.CreateParametersForImport(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.8.2 Import key materials

Function

This API is used to import key materials.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/import-key-material

Table 4-424 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-425 Request header parameters

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

Table 4-426 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
import_token	Yes	String	Key import token in Base64 format, which matches the regular expression ^[0-9a-zA-Z+/=]{200,6144}$.
encrypted_key_material	Yes	String	Encrypted symmetric key material. The value is in Base64 format and matches the regular expression **^[0-9a-zA-Z+/=]{344,360}$**. If an asymmetric key is imported, this parameter is used as a temporary intermediate key during private key encryption.

Parameter	Mandatory	Type	Description
encrypted_privatekey	No	String	Private key encrypted using a temporary intermediate key. Specify this parameter if an asymmetric key is imported. The value is in Base64 format and matches the regular expression **^[0-9a-zA-Z+/=]{200,6144}$**.
expiration_time	No	Long	Expiration time of the key material. The value is a timestamp which indicates how many seconds it has been since January 1, 1970, for example, **1550291833**. The key material will be deleted by KMS in 24 hours after the expiration.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 400

Table 4-427 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-428 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-429 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-430 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-431 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-432 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-433 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-434 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-435 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-436 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-437 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-438 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-439 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-440 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Example Requests

Import ciphertext key materials to the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "import_token" : "AACIBjY2ZTQxYItNDU4Ny04OTIxLWVhZTVhZjg5NDZm....",
  "encrypted_key_material" : "e0wTU/YJT/HDxsEv2NE+3CKT1..."
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Import ciphertext key materials to the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ImportKeyMaterialSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ImportKeyMaterialRequest request = new ImportKeyMaterialRequest();
        ImportKeyMaterialRequestBody body = new ImportKeyMaterialRequestBody();
```



```
        body.withEncryptedKeyMaterial("e0wTU/YJT/HDxsEv2NE+3CKT1...");
        body.withImportToken("AACIBjY2ZTQxYItNDU4Ny04OTIxLWVhZTVhZjg5NDZm....");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            ImportKeyMaterialResponse response = client.importKeyMaterial(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Import ciphertext key materials to the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ImportKeyMaterialRequest()
        request.body = ImportKeyMaterialRequestBody(
            encrypted_key_material="e0wTU/YJT/HDxsEv2NE+3CKT1...",
            import_token="AACIBjY2ZTQxYItNDU4Ny04OTIxLWVhZTVhZjg5NDZm....",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.import_key_material(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

Import ciphertext key materials to the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ImportKeyMaterialRequest{}
    request.Body = &model.ImportKeyMaterialRequestBody{
        EncryptedKeyMaterial: "e0wTU/YJT/HDxsEv2NE+3CKT1...",
        ImportToken: "AACIBjY2ZTQxYItNDU4Ny04OTIxLWVhZTVhZjg5NDZm....",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.ImportKeyMaterial(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.8.3 Delete key materials

Function

This API is used to delete key materials.

Constraints

Materials of imported asymmetric keys cannot be deleted. To delete an asymmetric key, schedule a deletion task.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/delete-imported-key-material

Table 4-441 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-442 Request header parameters

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

Table 4-443 Request body parameters

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

Response Parameters

Status code: 400

Table 4-444 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-445 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-446 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-447 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-448 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-449 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-450 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-451 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-452 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-453 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-454 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-455 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-456 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-457 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Delete the materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Delete the materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DeleteImportedKeyMaterialSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
```



```
        DeleteImportedKeyMaterialRequest request = new DeleteImportedKeyMaterialRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            DeleteImportedKeyMaterialResponse response = client.deleteImportedKeyMaterial(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Delete the materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteImportedKeyMaterialRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.delete_imported_key_material(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Delete the materials of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.


```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteImportedKeyMaterialRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.DeleteImportedKeyMaterial(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.9 Querying Keys

4.1.9.1 Query the key list

Function

This API is used to query the key list.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-458 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-459 Request header parameters

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

Table 4-460 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of returned records. The default value is **2000**. The value should be no larger than the maximum number of keys, for example, **20**. If the number of retrieved results is greater than this value, **true** is returned for the response parameter **truncated**, indicating that multiple pages of results are returned.
marker	No	String	Start position of pagination query. If truncated is true in the response, you can send consecutive requests to obtain more records. Set marker to the value of next_marker in the response. Example: 10
key_state	No	String	Key status which matches the regular expression **^[1-5]{1}$**. Possible values are as follows: **1**: To be activated **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported
key_spec	No	String	Key generation algorithm. The default value is **AES_256**. Set this parameter to **ALL** to query all the keys (including asymmetric keys). Possible values are as follows: **AES_256 SM4 RSA_2048 RSA_3072 RSA_4096 EC_P256 EC_P384 SM2 ALL**

Parameter	Mandatory	Type	Description
enterprise_project_id	No	String	Enterprise project ID. If you haven't created any enterprise project, do not configure this field. If you have created an enterprise project, configure this field for resource querying. If you do not configure this field, all resources of enterprise projects that are granted are queried by default. If the value of "**enterprise_project_id**" is **all**. The value must meet one of the following requirements: The value is **all**. The value is **0**. The value matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-461 Response body parameters

Parameter	Type	Description
keys	Array of strings	Key ID list.
key_details	Array of **KeyDetails** objects	List of key details. For details, see **KeyDetails**.
next_marker	String	Value of marker used for obtaining the next page of results. If truncated is false, next_marker is left blank.
truncated	String	Whether there is another page. **true**: There is more data on the next page. **false**: This is the last page.
total	Integer	Total number of keys.

Table 4-462 KeyDetails

Parameter	Type	Description
key_id	String	Key ID.
domain_id	String	User domain ID.
key_alias	String	Key alias.
realm	String	Key realm.
key_spec	String	Key generation algorithm. Possible values are as follows: **AES_256 SM4 RSA_2048 RSA_3072 RSA_4096 EC_P256 EC_P384 SM2**
key_usage	String	Key usage. Possible values are as follows: **ENCRYPT_DECRYPT SIGN_VERIFY**
key_description	String	Key description.
creation_date	String	Time when the key was created. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_deletion_date	String	Time when the key was scheduled to be deleted. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
key_state	String	Key status which matches the regular expression **^[1-5]{1}$**. Possible values are as follows: **1**: To be activated **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported
default_key_flag	String	Master key identifier. The value is 1 for Default Master Keys and 0 for non-default master keys.
key_type	String	Key type.
expiration_time	String	Time when the key material expires. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
origin	String	Key source. The default value is **kms**. Possible values are as follows: **kms**: The key is generated by KMS. **external**: The key is imported.
key_rotation_enabled	String	Key rotation status. The default value is false, indicating that key rotation is disabled.

Parameter	Type	Description
sys_enterprise_project_id	String	Enterprise project ID. The default value is **0**. If you have created an enterprise project, the resources are listed in the default enterprise project. If you haven't created an enterprise project, the resources are not listed in the enterprise project.
keystore_id	String	Keystore ID

Status code: 400

Table 4-463 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-464 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-465 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-466 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-467 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-468 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-469 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-470 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-471 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-472 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-473 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-474 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-475 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-476 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the keys whose aliases start with **1**. At most two keys are returned.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "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" : "test",
    "realm" : "cn-north-7",
    "key_description" : "key_description",
    "creation_date" : "1502799822000",
    "scheduled_deletion_date" : "",
    "key_spec" : "AES_256",
    "key_usage" : "ENCRYPT_DECRYPT",
    "key_state" : "2",
    "default_key_flag" : "0",
    "key_type" : "1",
    "expiration_time" : "1501578672000",
    "origin" : "kms",
    "key_rotation_enabled" : "true",
    "sys_enterprise_project_id" : "0"
  }, {
    "key_id" : "2e258389-bb1e-4568-a1d5-e1f50adf70ea",
    "domain_id" : "00074811d5c27c4f8d48bb91e4a1dcfd",
    "key_alias" : "test",
    "realm" : "realm",
    "key_description" : "key_description",
    "creation_date" : "1502799822000",
    "scheduled_deletion_date" : "",
    "key_spec" : "AES_256",
    "key_usage" : "ENCRYPT_DECRYPT",
    "key_state" : "2",
    "default_key_flag" : "0",
    "key_type" : "1",
    "expiration_time" : "1501578672000",
    "origin" : "kms",
    "key_rotation_enabled" : "true",
    "sys_enterprise_project_id" : "0"
  } ],
  "next_marker" : "",
  "truncated" : "false",
  "total" : 2
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

Query the keys whose aliases start with **1**. At most two keys are returned.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ListKeysSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKeysRequest request = new ListKeysRequest();
        ListKeysRequestBody body = new ListKeysRequestBody();
        body.withMarker("1");
        body.withLimit("2");
        request.withBody(body);
        try {
            ListKeysResponse response = client.listKeys(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Query the keys whose aliases start with **1**. At most two keys are returned.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKeysRequest()
        request.body = ListKeysRequestBody(
            marker="1",
            limit="2"
        )
        response = client.list_keys(request)
        print(response)
```



```
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Query the keys whose aliases start with **1**. At most two keys are returned.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKeysRequest{}
    markerListKeysRequestBody:= "1"
    limitListKeysRequestBody:= "2"
    request.Body = &model.ListKeysRequestBody{
        Marker: &markerListKeysRequestBody,
        Limit: &limitListKeysRequestBody,
    }
    response, err := client.ListKeys(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.9.2 Query key details

Function

This API is used to query key details.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/describe-key

Table 4-477 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-478 Request header parameters

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

Table 4-479 Request body parameters

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

Response Parameters

Status code: 200

Table 4-480 Response body parameters

Parameter	Type	Description
key_info	**KeyDetails** object	Key details.

Table 4-481 KeyDetails

Parameter	Type	Description
key_id	String	Key ID.

Parameter	Type	Description
domain_id	String	User domain ID.
key_alias	String	Key alias.
realm	String	Key realm.
key_spec	String	Key generation algorithm. Possible values are as follows: **AES_256 SM4 RSA_2048 RSA_3072 RSA_4096 EC_P256 EC_P384 SM2**
key_usage	String	Key usage. Possible values are as follows: **ENCRYPT_DECRYPT SIGN_VERIFY**
key_description	String	Key description.
creation_date	String	Time when the key was created. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
scheduled_deletion_date	String	Time when the key was scheduled to be deleted. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
key_state	String	Key status which matches the regular expression **^[1-5]{1}$**. Possible values are as follows: **1**: To be activated **2**: Enabled **3**: Disabled **4**: To be deleted **5**: To be imported
default_key_flag	String	Master key identifier. The value is 1 for Default Master Keys and 0 for non-default master keys.
key_type	String	Key type.
expiration_time	String	Time when the key material expires. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
origin	String	Key source. The default value is **kms**. Possible values are as follows: **kms**: The key is generated by KMS. **external**: The key is imported.
key_rotation_enabled	String	Key rotation status. The default value is false, indicating that key rotation is disabled.
sys_enterprise_project_id	String	Enterprise project ID. The default value is **0**. If you have created an enterprise project, the resources are listed in the default enterprise project. If you haven't created an enterprise project, the resources are not listed in the enterprise project.

Parameter	Type	Description
keystore_id	String	Keystore ID

Status code: 400

Table 4-482 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-483 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-484 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-485 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-486 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-487 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-488 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-489 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-490 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-491 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-492 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-493 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-494 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-495 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the details of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_info" : {
    "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "domain_id" : "00074811d5c27c4f8d48bb91e4a1dcfd",
    "key_alias" : "test",
    "realm" : "test",
```



```
    "key_description" : "key_description",
    "creation_date" : "1502799822000",
    "scheduled_deletion_date" : "",
    "key_spec" : "AES_256",
    "key_usage" : "ENCRYPT_DECRYPT",
    "key_state" : "2",
    "default_key_flag" : "0",
    "key_type" : "1",
    "expiration_time" : "1501578672000",
    "origin" : "kms",
    "key_rotation_enabled" : "false",
    "sys_enterprise_project_id" : "0"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Query the details of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ListKeyDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKeyDetailRequest request = new ListKeyDetailRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            ListKeyDetailResponse response = client.listKeyDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
```



```
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Query the details of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKeyDetailRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.list_key_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Query the details of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
```



```
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKeyDetailRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.ListKeyDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.9.3 Querying a Public Key

Function

This API is used to query the public key of a specified asymmetric key.

Constraints

This API cannot be used to obtain the public key of a symmetric key.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/get-publickey

Table 4-496 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-497 Request header parameters

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

Table 4-498 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

Parameter	Mandatory	Type	Description
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-499 Response body parameters

Parameter	Type	Description
key_id	String	Key ID
public_key	String	Public key information

Status code: 400

Table 4-500 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-501 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-502 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-503 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-504 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-505 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-506 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-507 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-508 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-509 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-510 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-511 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-512 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-513 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the public key of an asymmetric key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
  "public_key" : "-----BEGIN PUBLIC KEY-----\r\nMIICCgKCAgEA3RQAXXwya9k4zV1/
Q3AFr37GO8JgFobDKZioAklLQdgElHZ/uxmP\r\n4bveNHpY6OI0Okk/1Ov8oJf+9W10VqVxbzihWa5n/
RMN0720DzLV7KuH4YylCGDb\r\n3JH/+bMbhF2qRArrrKod0kR9rYHrdPkI7O5fYQQprZ3kWnPgrhDoDFC8ja
+OelOg\r\nn4MMOGGYA/DAOb0XyxPnGl26PnUtvvF7aZbMW5x/Yq2yAVFE1cjqLaH7/j1C8KYE\r
\naOSYtl2nOif28WoweFavXpgVsb/iICTfqgC91BtCSFC5pT8vqZCimfoHmJCAkZa5\r
\nZ8QIqkOOf9F6iMqqIz7pGKgQSUmoKKY9j6DK3OwXDOB5gKu0vyuz+gW3b4SZn+Xa\r
\nKkEN8ZpXsdQdEGpe4SwIzSVyUGYNBOCLrsydBcPR7jWgQ6gs56IJrV2pdAtmBwKd\r\n6l33z1tQ7+/
h3IrZxXuuej/fRRUMlbVcmhTS2l6vle7HXgZj/dWzPsLLg9MGHu0+\r\no9PRr+brxTbrf5e2Zdr1ad35X/
b86gx7Grg1sYPkly2aEI4fsnDGPgFrudG+Hzx/\r\nABHejYfJEI6P0SXCzB/oDMkjw6XKhTSojMzuncAP/AM
+0LVYQxQe750qkb3hjBT0\r\nq/HBl/
4zMXA03tMb9QySnLK63uo64JMJiBsEe7wPLhHB3VzBZk9SvvECAwEAAQ==\r\n-----END PUBLIC KEY-----\r\n"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Query the public key of an asymmetric key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowPublicKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);
```



```
        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPublicKeyRequest request = new ShowPublicKeyRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            ShowPublicKeyResponse response = client.showPublicKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Query the public key of an asymmetric key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowPublicKeyRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.show_public_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

Query the public key of an asymmetric key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPublicKeyRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.ShowPublicKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.9.4 Query instance quantity

Function

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

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-514 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-515 Request header parameters

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

Response Parameters

Status code: 200

Table 4-516 Response body parameters

Parameter	Type	Description
instance_num	Integer	Number of non-default CMKs.

Status code: 400

Table 4-517 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-518 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-519 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-520 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-521 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-522 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-523 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-524 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-525 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-526 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-527 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-528 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-529 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-530 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the number of instances created in the project whose ID is **0dea2644dc80d5d22ff1c01e3ebea6fc**.

```
/v1.0/0dea2644dc80d5d22ff1c01e3ebea6fc/kms/user-instances
```

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowUserInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowUserInstancesRequest request = new ShowUserInstancesRequest();
        try {
            ShowUserInstancesResponse response = client.showUserInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowUserInstancesRequest()
        response = client.show_user_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowUserInstancesRequest{}
    response, err := client.ShowUserInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.9.5 Query quotas

Function

This API is used to query quotas, including the total number of CMKs that can be created by a user and the current quota usage.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-531 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-532 Request header parameters

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

Response Parameters

Status code: 200

Table 4-533 Response body parameters

Parameter	Type	Description
quotas	**Quotas** object	Quota details.

Table 4-534 Quotas

Parameter	Type	Description
resources	Array of **Resources** objects	Resource quota list. For details, see **Resources**.

Table 4-535 Resources

Parameter	Type	Description
type	String	Quota type. Possible values are as follows: **CMK**: custom master key (CMK) **grant_per_CMK**: number of grants can be created by a single CMK
used	Integer	Used quotas.
quota	Integer	Total quotas.

Status code: 400

Table 4-536 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-537 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-538 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-539 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-540 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-541 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-542 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-543 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-544 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-545 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-546 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-547 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-548 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-549 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the quota usage of the project whose ID is **0dea2644dc80d5d22ff1c01e3ebea6fc**.

```
/v1.0/0dea2644dc80d5d22ff1c01e3ebea6fc/kms/user-quotas
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "quotas" : {
    "resources" : [ {
      "quota" : 20,
      "used" : 20,
      "type" : "CMK"
    }, {
      "quota" : 100,
      "used" : 0,
      "type" : "grant_per_CMK"
    } ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowUserQuotasSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowUserQuotasRequest request = new ShowUserQuotasRequest();
        try {
            ShowUserQuotasResponse response = client.showUserQuotas(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
```



```
    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowUserQuotasRequest()
        response = client.show_user_quotas(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowUserQuotasRequest{}
    response, err := client.ShowUserQuotas(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.10 Key Grant Management

4.1.10.1 Create a grant

Function

This API is used to create a grant. Granted users can perform operations on the granted keys. The service default CMK whose suffix is **/default** cannot be granted.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-550 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-551 Request header parameters

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

Table 4-552 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
grantee_principal	Yes	String	ID of the granted user. The value contains 1 to 64 bytes and matches the regular expression **^[a-zA-Z0-9]{1,64}$**, for example, **0d0466b00d0466b00d0466b00d0466b0**.
operations	Yes	Array of strings	List of granted operations. Possible values are as follows: **create-datakey**: Create a DEK. **create-datakey-without-plaintext**: Create a DEK that does not contain plaintext. **encrypt-datakey**: Encrypt DEK. **decrypt-datakey**: Decrypt DEK. **describe-key**: Query the key information. **create-grant**: Create the grant. **retire-grant**: Retire the grant. **encrypt-data**: Encrypt data. **decrypt-data**: Decrypt data. The value cannot be only **create-grant**.

Parameter	Mandatory	Type	Description
name	No	String	Grant name. The value is a string of 1 to 255 characters and matches the regular expression ^[a-zA-Z0-9:/_-]{1,255}$.
retiring_principal	No	String	ID of the user whose grant can be retired. The value contains 1 to 64 bytes and matches the regular expression **^[a-zA-Z0-9]{1,64}$**, for example, **0d0466b00d0466b00d0466b00d0466b0**.
grantee_principal_type	No	String	Grant type. Values: user, domain. The default value is user.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-553 Response body parameters

Parameter	Type	Description
grant_id	String	Grant ID, which contains 64 bytes.

Status code: 400

Table 4-554 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-555 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-556 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-557 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-558 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-559 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-560 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-561 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-562 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-563 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-564 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-565 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-566 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-567 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Grant user **13gg44z4g2sglzk0egw0u726zoyzvrs8"ID "0d0466b0-e727-4d9c-b35d-f84bb474a37f** with permission to query, create, and encrypt a DEK.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "operations" : [ "describe-key", "create-datakey", "encrypt-datakey" ],
  "grantee_principal" : "13gg44z4g2sglzk0egw0u726zoyzvrs8",
  "grantee_principal_type" : "user",
  "retiring_principal" : "13gg44z4g2sglzk0egw0u726zoyzvrs8"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "grant_id" : "7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Grant user **13gg44z4g2sglzk0egw0u726zoyzvrs8"ID "0d0466b0-e727-4d9c-b35d-f84bb474a37f** with permission to query, create, and encrypt a DEK.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
```



```
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateGrantSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateGrantRequest request = new CreateGrantRequest();
        CreateGrantRequestBody body = new CreateGrantRequestBody();
        List<String> listbodyOperations = new ArrayList<>();
        listbodyOperations.add("describe-key");
        listbodyOperations.add("create-datakey");
        listbodyOperations.add("encrypt-datakey");

body.withGranteePrincipalType(CreateGrantRequestBody.GranteePrincipalTypeEnum.fromValue("user"));
        body.withRetiringPrincipal("13gg44z4g2sglzk0egw0u726zoyzvrs8");
        body.withOperations(listbodyOperations);
        body.withGranteePrincipal("13gg44z4g2sglzk0egw0u726zoyzvrs8");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            CreateGrantResponse response = client.createGrant(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Grant user **13gg44z4g2sglzk0egw0u726zoyzvrs8"ID "0d0466b0-e727-4d9c-b35d-f84bb474a37f** with permission to query, create, and encrypt a DEK.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
```



```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateGrantRequest()
        listOperationsbody = [
            "describe-key",
            "create-datakey",
            "encrypt-datakey"
        ]
        request.body = CreateGrantRequestBody(
            grantee_principal_type="user",
            retiring_principal="13gg44z4g2sglzk0egw0u726zoyzvrs8",
            operations=listOperationsbody,
            grantee_principal="13gg44z4g2sglzk0egw0u726zoyzvrs8",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.create_grant(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Grant user **13gg44z4g2sglzk0egw0u726zoyzvrs8"ID "0d0466b0-e727-4d9c-b35d-f84bb474a37f** with permission to query, create, and encrypt a DEK.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
            WithCredential(auth).
            Build())

    request := &model.CreateGrantRequest{}
    var listOperationsbody = []string{
        "describe-key",
        "create-datakey",
        "encrypt-datakey",
    }
    granteePrincipalTypeCreateGrantRequestBody:=
model.GetCreateGrantRequestBodyGranteePrincipalTypeEnum().USER
    retiringPrincipalCreateGrantRequestBody:= "13gg44z4g2sglzk0egw0u726zoyzvrs8"
    request.Body = &model.CreateGrantRequestBody{
        GranteePrincipalType: &granteePrincipalTypeCreateGrantRequestBody,
        RetiringPrincipal: &retiringPrincipalCreateGrantRequestBody,
        Operations: listOperationsbody,
        GranteePrincipal: "13gg44z4g2sglzk0egw0u726zoyzvrs8",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.CreateGrant(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.10.2 Revoke a grant

Function

This API is used to revoke a grant. Note: Only the key creator can revoke the grant of a key.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/revoke-grant

Table 4-568 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-569 Request header parameters

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

Table 4-570 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte Key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

Parameter	Mandatory	Type	Description
grant_id	Yes	String	A 64-byte grant ID which matches the regular expression **^[A-Fa-f0-9]{64}$**, for example, **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**.
sequence	No	String	36-byte sequence number of a request message. Example: 919c82d4-8046-4722-9094-35c3c6524cff

Response Parameters

Status code: 400

Table 4-571 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-572 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-573 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-574 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-575 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-576 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-577 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-578 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-579 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-580 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-581 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-582 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-583 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-584 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Revoke the grant (ID: **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**) for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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



```
  "grant_id" : "7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d"
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Revoke the grant (ID: **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**) for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class CancelGrantSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CancelGrantRequest request = new CancelGrantRequest();
        RevokeGrantRequestBody body = new RevokeGrantRequestBody();
        body.withGrantId("7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d");
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            CancelGrantResponse response = client.cancelGrant(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
```



```
        }
    }
}
```

Python

Revoke the grant (ID: **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**) for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CancelGrantRequest()
        request.body = RevokeGrantRequestBody(
            grant_id="7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d",
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.cancel_grant(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Revoke the grant (ID: **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**) for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```



```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CancelGrantRequest{}
    request.Body = &model.RevokeGrantRequestBody{
        GrantId: "7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d",
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.CancelGrant(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.10.3 Retire a grant

Function

This API is used to retire grants. After grants are retired, the granted users do not have the permission to operate keys. For example, user A grants user B the permission to operate key **A** and grants user C the permission to revoke the grant. In this case, all users can retire the grant. After the grant is retired, user B cannot use key **A**. Only the following users can retire a grant: User who created the grant User who has been granted the **retiring_principal** permission User who has been granted the **grantee_principal** permission if **retire-grant** is contained in the grant list

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/retire-grant

Table 4-585 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-586 Request header parameters

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

Table 4-587 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte Key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
grant_id	Yes	String	A 64-byte grant ID which matches the regular expression **^[A-Fa-f0-9]{64}$**, for example, **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**.
sequence	No	String	36-byte sequence number of a request message. Example: 919c82d4-8046-4722-9094-35c3c6524cff

Response Parameters

Status code: 400

Table 4-588 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-589 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-590 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-591 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-592 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-593 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-594 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-595 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-596 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-597 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-598 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-599 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-600 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-601 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Example Requests

Delete the grant (ID: **7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d**) for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "grant_id" : "7c9a3286af4fcca5f0a385ad13e1d21a50e27b6dbcab50f37f30f93b8939827d"
}
```

Example Responses

None

Status Codes

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

Error Codes

See **Error Codes**.

4.1.10.4 Query the grant list

Function

This API is used to query the users who have been granted the permissions of a key.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/list-grants

Table 4-602 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-603 Request header parameters

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

Table 4-604 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

Parameter	Mandatory	Type	Description
limit	No	String	Number of returned records. The default value is **1000**. The value cannot be larger than the maximum number of grants, for example, **100**. If the number of retrieved results is greater than this value, **true** is returned for the response parameter **truncated**, indicating that multiple pages of results are returned.
marker	No	String	Start position of the paginated query. If **truncated** is **true** in the response, you can send consecutive requests to obtain more records. Set **marker** to the value of **next_marker** in the response, for example, **10**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-605 Response body parameters

Parameter	Type	Description
grants	Array of **Grants** objects	Grant list. For details, see the data structure description of the **grants** field.
next_marker	String	Value of **marker** used for obtaining the next page of results. If the value of **truncated** is **false**, **next_marker** is left empty.
truncated	String	Whether there is another page. **true**: There is more data on the next page. **false**: This is the last page.
total	Integer	Total number of grants.

Table 4-606 Grants

Parameter	Type	Description
key_id	String	Key ID.
grant_id	String	Grant ID, which contains 64 bytes.
grantee_principal	String	ID of the granted user. The value contains 1 to 64 bytes and matches the regular expression **^[a-zA-Z0-9]{1,64}$**, for example, **0d0466b00d0466b00d0466b00d0466b0**.
grantee_principal_type	String	Grant type. The value can be **user** or **domain**.
operations	Array of strings	List of granted operations. Possible values are as follows: **create-datakey**: Create a DEK. **create-datakey-without-plaintext**: Create a DEK that does not contain plaintext. **encrypt-datakey**: Encrypt DEK. **decrypt-datakey**: Decrypt DEK. **describe-key**: Query the key information. **create-grant**: Create the grant. **retire-grant**: Retire the grant. **encrypt-data**: Encrypt data. **decrypt-data**: Decrypt data. The value cannot be only **create-grant**.
issuing_principal	String	ID of the user who created the grant. The value contains 1 to 64 bytes and matches the regular expression **"^[a-zA-Z0-9]{1,64}$,"** for example, **0d0466b00d0466b00d0466b00d0466b0**.
creation_date	String	Creation time. The value is a timestamp which indicates how many seconds it has been since January 1, 1970, for example, **1497341531000**.
name	String	Grant name. The value is a string of 1 to 255 characters and matches the regular expression ^[a-zA-Z0-9:/_-]{1,255}$.
retiring_principal	String	ID of the user whose grant can be retired. The value contains 1 to 64 bytes and matches the regular expression **^[a-zA-Z0-9]{1,64}$**, for example, **0d0466b00d0466b00d0466b00d0466b0**.

Status code: 400

Table 4-607 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-608 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-609 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-610 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-611 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-612 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-613 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-614 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-615 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-616 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-617 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-618 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-619 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-620 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the grant list of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "grants" : [ {
    "operations" : [ "create-datakey", "describe-key" ],
    "issuing_principal" : "8b961fb414344d59825ba0c8c008c815",
    "key_id" : "737fd52b-36c4-4c91-972e-f6e202de9f6e",
    "grant_id" : "dd3f03e9229a5e47a41be6c27a630e60d5cbdbad2be89465d63109ad034db7d8",
    "grantee_principal" : "13gg44z4g2sglzk0egw0u726zoyzvrs8",
    "name" : "13gg44z4g2sglzk0egw0u726zoyzvrs8",
    "creation_date" : "1597062260000",
    "grantee_principal_type" : "user"
  } ],
  "next_marker" : "",
  "total" : 1,
  "truncated" : "false"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Query the grant list of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.


```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ListGrantsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListGrantsRequest request = new ListGrantsRequest();
        ListGrantsRequestBody body = new ListGrantsRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            ListGrantsResponse response = client.listGrants(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Query the grant list of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```



```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGrantsRequest()
        request.body = ListGrantsRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.list_grants(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Query the grant list of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListGrantsRequest{}
    request.Body = &model.ListGrantsRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.ListGrants(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```



```
  }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.10.5 Query grants that can be retired

Function

This API is used to query the list of grants that can be retired.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/list-retirable-grants

Table 4-621 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-622 Request header parameters

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

Table 4-623 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	String	Number of returned records of grants that can be retired. If the number of retrieved results is greater than this value, **true** is returned for the response parameter **truncated**, indicating that multiple pages of results are returned.
marker	No	String	Start position of the paginated query. If **truncated** is **true** in the response, you can send consecutive requests to obtain more records. Set **marker** to the value of **next_marker** in the response, for example, **10**.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-624 Response body parameters

Parameter	Type	Description
grants	Array of **Grants** objects	Grant list. For details, see the data structure description of the **grants** field.
next_marker	String	Value of **marker** used for obtaining the next page of results. If the value of **truncated** is **false**, **next_marker** is left empty.
total	Integer	Number of grants that can be retired
truncated	String	Whether there is another page. **true**: There is more data on the next page. **false**: This is the last page.

Table 4-625 Grants

Parameter	Type	Description
key_id	String	Key ID.
grant_id	String	Grant ID, which contains 64 bytes.
grantee_principal	String	ID of the granted user. The value contains 1 to 64 bytes and matches the regular expression **^[a-zA-Z0-9]{1,64}$**, for example, **0d0466b00d0466b00d0466b00d0466b0**.
grantee_principal_type	String	Grant type. The value can be **user** or **domain**.
operations	Array of strings	List of granted operations. Possible values are as follows: **create-datakey**: Create a DEK. **create-datakey-without-plaintext**: Create a DEK that does not contain plaintext. **encrypt-datakey**: Encrypt DEK. **decrypt-datakey**: Decrypt DEK. **describe-key**: Query the key information. **create-grant**: Create the grant. **retire-grant**: Retire the grant. **encrypt-data**: Encrypt data. **decrypt-data**: Decrypt data. The value cannot be only **create-grant**.
issuing_principal	String	ID of the user who created the grant. The value contains 1 to 64 bytes and matches the regular expression **"^[a-zA-Z0-9]{1,64}$,"**, for example, **0d0466b00d0466b00d0466b00d0466b0**.
creation_date	String	Creation time. The value is a timestamp which indicates how many seconds it has been since January 1, 1970, for example, **1497341531000**.

Parameter	Type	Description
name	String	Grant name. The value is a string of 1 to 255 characters and matches the regular expression ^[a-zA-Z0-9:/_-]{1,255}$.
retiring_principal	String	ID of the user whose grant can be retired. The value contains 1 to 64 bytes and matches the regular expression **^[a-zA-Z0-9]{1,64}$**, for example, **0d0466b00d0466b00d0466b00d0466b0**.

Status code: 400

Table 4-626 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-627 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-628 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-629 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-630 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-631 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-632 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-633 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-634 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-635 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-636 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-637 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-638 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-639 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query grants that can be retired. At most 1,000 grants can be returned.

```
{
  "limit" : "1000"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "grants" : [ {
    "operations" : [ "create-datakey", "describe-key" ],
    "issuing_principal" : "8b961fb414344d59825ba0c8c008c815",
```



```
    "key_id" : "737fd52b-36c4-4c91-972e-f6e202de9f6e",
    "grant_id" : "dd3f03e9229a5e47a41be6c27a630e60d5cbdbad2be89465d63109ad034db7d8",
    "grantee_principal" : "13gg44z4g2sglzk0egw0u726zoyzvrs8",
    "name" : "13gg44z4g2sglzk0egw0u726zoyzvrs8",
    "creation_date" : "1597062260000",
    "grantee_principal_type" : "user"
  } ],
  "next_marker" : "",
  "total" : 1,
  "truncated" : "false"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Query grants that can be retired. At most 1,000 grants can be returned.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ListRetirableGrantsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListRetirableGrantsRequest request = new ListRetirableGrantsRequest();
        ListRetirableGrantsRequestBody body = new ListRetirableGrantsRequestBody();
        body.withLimit("1000");
        request.withBody(body);
        try {
            ListRetirableGrantsResponse response = client.listRetirableGrants(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
```



```
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Query grants that can be retired. At most 1,000 grants can be returned.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListRetirableGrantsRequest()
        request.body = ListRetirableGrantsRequestBody(
            limit="1000"
        )
        response = client.list_retirable_grants(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Query grants that can be retired. At most 1,000 grants can be returned.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```



```
    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRetirableGrantsRequest{}
    limitListRetirableGrantsRequestBody:= "1000"
    request.Body = &model.ListRetirableGrantsRequestBody{
        Limit: &limitListRetirableGrantsRequestBody,
    }
    response, err := client.ListRetirableGrants(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.11 Key Rotation Management

4.1.11.1 Enable key rotation

Function

This API is used to enable CMK rotation. Note: The default rotation period is 365 days. Default CMKs and imported keys cannot be rotated.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/enable-key-rotation

Table 4-640 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-641 Request header parameters

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

Table 4-642 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

Parameter	Mandatory	Type	Description
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 400

Table 4-643 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-644 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-645 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-646 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-647 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-648 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-649 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-650 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-651 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-652 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-653 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-654 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-655 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-656 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Enable rotation for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

None

Status Codes

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

Error Codes

See **Error Codes**.

4.1.11.2 Disable key rotation

Function

This API is used to disable CMK rotation.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-657 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-658 Request header parameters

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

Table 4-659 Request body parameters

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

Response Parameters

Status code: 400

Table 4-660 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-661 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-662 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-663 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-664 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-665 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-666 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-667 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-668 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-669 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-670 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-671 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-672 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-673 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Disable rotation for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Disable rotation for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DisableKeyRotationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
```



```
        DisableKeyRotationRequest request = new DisableKeyRotationRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            DisableKeyRotationResponse response = client.disableKeyRotation(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Disable rotation for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisableKeyRotationRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.disable_key_rotation(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Disable rotation for the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.


```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisableKeyRotationRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.DisableKeyRotation(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.11.3 Modify key rotation interval

Function

This API is used to change the CMK rotation period.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-674 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-675 Request header parameters

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

Table 4-676 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
rotation_interval	Yes	Integer	Rotation period. The value is an integer ranging from **30** to **365**. Set this parameter based on the key usage frequency. If the key is frequently used, set this parameter to a rather low value, vice versa.
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 400

Table 4-677 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-678 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-679 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-680 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-681 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-682 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-683 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-684 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-685 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-686 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-687 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-688 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-689 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-690 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Example Requests

Change the rotation period of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to 30 days.

```
{
  "key_id" : "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
  "rotation_interval" : 30
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Change the rotation period of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to 30 days.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class UpdateKeyRotationIntervalSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateKeyRotationIntervalRequest request = new UpdateKeyRotationIntervalRequest();
        UpdateKeyRotationIntervalRequestBody body = new UpdateKeyRotationIntervalRequestBody();
        body.withRotationInterval(30);
```



```
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            UpdateKeyRotationIntervalResponse response = client.updateKeyRotationInterval(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Change the rotation period of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to 30 days.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateKeyRotationIntervalRequest()
        request.body = UpdateKeyRotationIntervalRequestBody(
            rotation_interval=30,
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.update_key_rotation_interval(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Change the rotation period of the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** to 30 days.


```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateKeyRotationIntervalRequest{}
    request.Body = &model.UpdateKeyRotationIntervalRequestBody{
        RotationInterval: int32(30),
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.UpdateKeyRotationInterval(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

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

Error Codes

See **Error Codes**.

4.1.11.4 Query key rotation status

Function

This API is used to query the CMK rotation status.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/kms/get-key-rotation-status

Table 4-691 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-692 Request header parameters

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

Table 4-693 Request body parameters

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

Response Parameters

Status code: 200

Table 4-694 Response body parameters

Parameter	Type	Description
key_rotation_enabled	Boolean	Key rotation status. The default value is false, indicating that key rotation is disabled.
rotation_interval	Integer	Rotation period. The value is an integer ranging from **30** to **365**. Set this parameter based on the key usage frequency. If the key is frequently used, set this parameter to a rather low value, vice versa.
last_rotation_time	String	Last key rotation time. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
number_of_rotations	Integer	Number of key rotations.

Status code: 400

Table 4-695 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-696 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-697 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-698 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-699 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-700 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-701 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-702 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-703 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-704 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-705 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-706 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-707 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-708 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Query the rotation status of the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

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

Example Responses

Status code: 200

Request succeeded.

```
{
  "key_rotation_enabled" : true,
  "rotation_interval" : 30,
  "last_rotation_time" : "1501578672000",
  "number_of_rotations" : 3
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Query the rotation status of the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
```



```
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowKeyRotationStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowKeyRotationStatusRequest request = new ShowKeyRotationStatusRequest();
        OperateKeyRequestBody body = new OperateKeyRequestBody();
        body.withKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        request.withBody(body);
        try {
            ShowKeyRotationStatusResponse response = client.showKeyRotationStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Query the rotation status of the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
```



```
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowKeyRotationStatusRequest()
        request.body = OperateKeyRequestBody(
            key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f"
        )
        response = client.show_key_rotation_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Query the rotation status of the CMK whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKeyRotationStatusRequest{}
    request.Body = &model.OperateKeyRequestBody{
        KeyId: "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    }
    response, err := client.ShowKeyRotationStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.12 Dedicated Keystore Management

4.1.12.1 Creating a Dedicated Keystore

Function

Create a dedicated keystore. The keystore uses Dedicated HSM instances to store keys.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/keystores

Table 4-709 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-710 Request header parameters

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

Table 4-711 Request body parameters

Parameter	Mandatory	Type	Description
keystore_alias	Yes	String	Alias of the dedicated keystore. The value contains 1 to 255 characters, matches the regular expression **^[a-zA-Z0-9:/_-]{1,255}$**, and must be unique.
hsm_cluster_id	Yes	String	ID of the Dedicated HSM cluster. Ensure that no dedicated keystores are created in the current cluster.
hsm_ca_cert	Yes	String	CA certificate of the Dedicated HSM cluster

Response Parameters

Status code: 200

Table 4-712 Response body parameters

Parameter	Type	Description
keystore	**KeystoreInfo** object	Keystore information

Table 4-713 KeystoreInfo

Parameter	Type	Description
keystore_id	String	Keystore ID

Parameter	Type	Description
domain_id	String	User domain ID

Status code: 400

Table 4-714 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-715 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-716 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-717 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-718 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-719 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-720 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-721 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-722 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-723 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-724 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-725 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-726 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-727 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Create a dedicated keystore whose alias is **keystore_alia1** and cluster ID is **hsm_cluster_id**.

```
{
  "keystore_alias" : "keystore_alia1",
  "hsm_cluster_id" : "hsm_cluster_id",
  "hsm_ca_cert" : "-----BEGIN CERTIFICATE---******----END CERTIFICATE-----"
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "keystore" : {
    "keystore_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "domain_id" : "b168fe00ff56492495a7d22974df2d0b"
```



```
  }
}
```

Status Codes

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

Error Codes

See **Error Codes**.

4.1.12.2 Querying the List of Dedicated Keystores

Function

This API is used to query the dedicated keystore list.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1.0/{project_id}/keystores

Table 4-728 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 4-729 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of returned records. The default value is **10**.
offset	No	Integer	Index location. The query starts from the next data specified by **offset**.

Request Parameters

Table 4-730 Request header parameters

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

Response Parameters

Status code: 200

Table 4-731 Response body parameters

Parameter	Type	Description
total	Integer	Total number of keystores
keystores	Array of **KeystoreDetails** objects	List of keystore details. For details, see **KeystoreDetails**.

Table 4-732 KeystoreDetails

Parameter	Type	Description
keystore_id	String	Keystore ID
domain_id	String	User domain ID
keystore_alias	String	Keystore alias
keystore_type	String	Keystore type

Parameter	Type	Description
hsm_cluster_id	String	Dedicated HSM cluster ID. Ensure that no dedicated keystores are created in the current cluster.
create_time	String	Keystore creation time, which is a UTC timestamp.

Status code: 400

Table 4-733 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-734 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-735 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-736 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-737 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-738 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-739 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-740 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-741 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-742 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-743 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-744 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-745 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-746 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "total" : 1,
  "keystores" : [ {
    "keystore_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "domain_id" : "b168fe00ff56492495a7d22974df2d0b",
    "keystore_alias" : "test",
    "keystore_type" : "typetest",
    "hsm_cluster_id" : "cluster_id",
```



```
    "create_time" : 1581507580000
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ListKeyStoresSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKeyStoresRequest request = new ListKeyStoresRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListKeyStoresResponse response = client.listKeyStores(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```



```
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKeyStoresRequest()
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_key_stores(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKeyStoresRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
```



```
    response, err := client.ListKeyStores(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.12.3 Obtaining a Dedicated Keystore

Function

This API is used to obtain a dedicated keystore.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1.0/{project_id}/keystores/{keystore_id}

Table 4-747 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keystore_id	Yes	String	Keystore ID

Request Parameters

Table 4-748 Request header parameters

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

Response Parameters

Status code: 200

Table 4-749 Response body parameters

Parameter	Type	Description
keystore	**KeystoreDetails** object	Keystore details

Table 4-750 KeystoreDetails

Parameter	Type	Description
keystore_id	String	Keystore ID
domain_id	String	User domain ID
keystore_alias	String	Keystore alias
keystore_type	String	Keystore type
hsm_cluster_id	String	Dedicated HSM cluster ID. Ensure that no dedicated keystores are created in the current cluster.

Parameter	Type	Description
create_time	String	Keystore creation time, which is a UTC timestamp.

Status code: 400

Table 4-751 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-752 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-753 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-754 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-755 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-756 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-757 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-758 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-759 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-760 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-761 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-762 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-763 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-764 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "keystore" : {
    "keystore_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "domain_id" : "b168fe00ff56492495a7d22974df2d0b",
    "keystore_alias" : "test",
    "keystore_type" : "typetest",
    "hsm_cluster_id" : "cluster_id",
    "create_time" : 1581507580000
  }
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class ShowKeyStoreSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowKeyStoreRequest request = new ShowKeyStoreRequest();
        try {
            ShowKeyStoreResponse response = client.showKeyStore(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowKeyStoreRequest()
        response = client.show_key_store(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKeyStoreRequest{}
    response, err := client.ShowKeyStore(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.12.4 Deleting a Dedicated Keystore

Function

This API is used to delete a dedicated keystore.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v1.0/{project_id}/keystores/{keystore_id}

Table 4-765 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keystore_id	Yes	String	Keystore ID

Request Parameters

Table 4-766 Request header parameters

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

Response Parameters

Status code: 400

Table 4-767 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-768 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-769 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-770 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-771 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-772 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-773 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-774 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-775 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-776 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-777 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-778 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-779 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-780 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DeleteKeyStoreSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteKeyStoreRequest request = new DeleteKeyStoreRequest();
        try {
            DeleteKeyStoreResponse response = client.deleteKeyStore(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteKeyStoreRequest()
        response = client.delete_key_store(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteKeyStoreRequest{}
    response, err := client.DeleteKeyStore(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.12.5 Enabling a Dedicated Keystore

Function

This API is used to enable a dedicated keystore.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/keystores/{keystore_id}/enable

Table 4-781 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keystore_id	Yes	String	Keystore ID

Request Parameters

Table 4-782 Request header parameters

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

Response Parameters

Status code: 200

Table 4-783 Response body parameters

Parameter	Type	Description
keystore	**KeyStoreStateInfo** object	Keystore status information

Table 4-784 KeyStoreStateInfo

Parameter	Type	Description
keystore_id	String	Keystore ID
keystore_state	String	Keystore status

Status code: 400

Table 4-785 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-786 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request

Parameter	Type	Description
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-787 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-788 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-789 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-790 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-791 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-792 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-793 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-794 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-795 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-796 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-797 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-798 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "keystore" : {
    "keystore_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "keystore_state" : "2"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class EnableKeyStoreSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```



```
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        EnableKeyStoreRequest request = new EnableKeyStoreRequest();
        try {
            EnableKeyStoreResponse response = client.enableKeyStore(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = EnableKeyStoreRequest()
        response = client.enable_key_store(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
```



```
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.EnableKeyStoreRequest{}
    response, err := client.EnableKeyStore(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.12.6 Disabling a Dedicated Keystore

Function

This API is used to disable a dedicated keystore.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1.0/{project_id}/keystores/{keystore_id}/disable

Table 4-799 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keystore_id	Yes	String	Keystore ID

Request Parameters

Table 4-800 Request header parameters

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

Response Parameters

Status code: 200

Table 4-801 Response body parameters

Parameter	Type	Description
keystore	**KeyStoreStateInfo** object	Keystore status information

Table 4-802 KeyStoreStateInfo

Parameter	Type	Description
keystore_id	String	Keystore ID
keystore_state	String	Keystore status

Status code: 400

Table 4-803 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-804 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-805 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-806 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-807 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-808 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-809 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-810 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-811 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-812 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-813 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-814 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-815 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-816 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "keystore" : {
    "keystore_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "keystore_state" : "3"
  }
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kms.v2.region.KmsRegion;
import com.huaweicloud.sdk.kms.v2.*;
import com.huaweicloud.sdk.kms.v2.model.*;


public class DisableKeyStoreSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KmsClient client = KmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DisableKeyStoreRequest request = new DisableKeyStoreRequest();
        try {
            DisableKeyStoreResponse response = client.disableKeyStore(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkms.v2.region.kms_region import KmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkms.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisableKeyStoreRequest()
        response = client.disable_key_store(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kms/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kms.NewKmsClient(
        kms.KmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisableKeyStoreRequest{}
    response, err := client.DisableKeyStore(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.1.13 Signature Verification

4.1.13.1 Verifying a Signature

Function

This API is used to verify the signature of a message or digest using the public key of an asymmetric key.

Constraints

Only the asymmetric key whose **key_usage** is **SIGN_VERIFY** can be used for signature verification. SM2 keys can only be used to sign message digests.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-817 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-818 Request header parameters

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

Table 4-819 Request body parameters

Parameter	Mandatory	Type	Description
key_id	Yes	String	A 36-byte key ID which matches the regular expression **^[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}$**, for example, **0d0466b0-e727-4d9c-b35d-f84bb474a37f**.
message	Yes	String	Message digest or message to be signed. The message must be smaller than 4,096 bytes and in Base64 format.
signature	Yes	String	Value of signature to be valued, which is in Base64 format.

Parameter	Mandatory	Type	Description
signing_algorithm	Yes	String	Signature algorithm. Possible values are as follows: **RSASSA_PSS_SHA_256 RSASSA_PSS_SHA_384 RSASSA_PSS_SHA_512 RSASSA_PKCS1_V1_5_SHA_256 RSASSA_PKCS1_V1_5_SHA_384 RSASSA_PKCS1_V1_5_SHA_512 ECDSA_SHA_256 ECDSA_SHA_384 ECDSA_SHA_512 SM2DSA_SM3**
message_type	No	String	Message type. The default value is **DIGEST**. Possible values are as follows: **DIGEST**: message digest **RAW**: message
sequence	No	String	A 36-byte serial number of a request message, for example, **919c82d4-8046-4722-9094-35c3c6524cff**

Response Parameters

Status code: 200

Table 4-820 Response body parameters

Parameter	Type	Description
key_id	String	Key ID
signature_valid	String	Whether the verification signature is valid. **true**: The signature is valid. **false**: The signature is invalid.

Status code: 400

Table 4-821 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-822 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 401

Table 4-823 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-824 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 403

Table 4-825 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-826 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 404

Table 4-827 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-828 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 500

Table 4-829 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-830 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 502

Table 4-831 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-832 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Status code: 504

Table 4-833 Response body parameters

Parameter	Type	Description
error	Object	Error message.

Table 4-834 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned by the error request
error_msg	String	Error information returned by the error request

Example Requests

Verify the signatures of messages and digests using the key whose ID is **0d0466b0-e727-4d9c-b35d-f84bb474a37f** and the **RSASSA_PKCS1_V1_5_SHA_256** algorithm.

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

Example Responses

Status code: 200

Request succeeded.

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

Status Codes

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

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

Error Codes

See **Error Codes**.

4.2 Key Pair Management APIs

4.2.1 Key Pair Management

4.2.1.1 Creating and Importing an SSH Key Pair

Function

Creating and Importing an SSH Key Pair

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs

Table 4-835 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

Table 4-836 Request header parameters

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

Table 4-837 Request body parameters

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

Table 4-838 CreateKeypairAction

Parameter	Mandatory	Type	Description
name	Yes	String	SSH key pair name. • The key pair name must be unique. • The SSH key pair name can contain at most 255 characters, including letters, digits, underscores (_), and hyphens (-).
type	No	String	SSH key pair type. The value can be **ssh** or **x509**.
public_key	No	String	String of an imported public key
scope	No	String	Tenant-level or user-level. The value can be **domain** or **user**.
user_id	No	String	User that an SSH key pair belongs to
key_protection	No	**KeyProtection** object	Private key hosting and protection for the SSH key pair.

Table 4-839 KeyProtection

Parameter	Mandatory	Type	Description
private_key	No	String	Private key of the imported SSH key pair.
encryption	Yes	**Encryption** object	How a private key is encrypted and stored.

Table 4-840 Encryption

Parameter	Mandatory	Type	Description
type	Yes	String	Value options: - default: The default encryption mode. Applicable to sites where KMS is not deployed. - kms: KMS encryption mode. If the KMS service is not available at the site, set this parameter to default.
kms_key_name	No	String	KMS key name. • If **type** is set to **kms**, you must enter the KMS key name or ID.
kms_key_id	No	String	KMS key ID. • If **type** is set to **kms**, you must enter the KMS key name or ID.

Response Parameters

Status code: 200

Table 4-841 Response body parameters

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

Table 4-842 CreateKeypairResp

Parameter	Type	Description
name	String	SSH key pair name
type	String	SSH key pair type. The value can be **ssh** or **x509**.
public_key	String	Public key information about an SSH key pair
private_key	String	Private key information about an SSH key pair. - When an SSH key pair is created, the response contains private_key information. - When an SSH key pair is imported, the response does not contain private_key information.
fingerprint	String	Fingerprint information about an SSH key pair
user_id	String	User that an SSH key pair belongs to

Status code: 400

Table 4-843 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

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

Example Responses

Status code: 200

Request succeeded.

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

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class CreateKeypairSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateKeypairRequest request = new CreateKeypairRequest();
        CreateKeypairRequestBody body = new CreateKeypairRequestBody();
        CreateKeypairAction keypairbody = new CreateKeypairAction();
        keypairbody.withName("demo2");
        body.withKeypair(keypairbody);
        request.withBody(body);
        try {
            CreateKeypairResponse response = client.createKeypair(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateKeypairRequest()
        keypairbody = CreateKeypairAction(
            name="demo2"
        )
        request.body = CreateKeypairRequestBody(
            keypair=keypairbody
        )
        response = client.create_keypair(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
```



```
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateKeypairRequest{}
    keypairbody := &model.CreateKeypairAction{
        Name: "demo2",
    }
    request.Body = &model.CreateKeypairRequestBody{
        Keypair: keypairbody,
    }
    response, err := client.CreateKeypair(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

See **Error Codes**.

4.2.1.2 Accessing the page for clearing private keys

Function

This API is used to delete the private key of an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-844 Path Parameters

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

Request Parameters

Table 4-845 Request header parameters

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

Response Parameters

Status code: 404

Table 4-846 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```



```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ClearPrivateKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ClearPrivateKeyRequest request = new ClearPrivateKeyRequest();
        try {
            ClearPrivateKeyResponse response = client.clearPrivateKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
```



```
        .build()

    try:
        request = ClearPrivateKeyRequest()
        response = client.clear_private_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ClearPrivateKeyRequest{}
    response, err := client.ClearPrivateKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	-

Status Code	Description
404	The requested resource does not exist or is not found.

Error Codes

See **Error Codes**.

4.2.1.3 Querying the SSH Key Pair List

Function

Querying the SSH Key Pair List

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/keypairs

Table 4-847 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 4-848 Query Parameters

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

Request Parameters

Table 4-849 Request header parameters

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

Response Parameters

Status code: 200

Table 4-850 Response body parameters

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

Table 4-851 Keypairs

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

Table 4-852 Keypair

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

Parameter	Type	Description
is_key_protection	Boolean	Whether to host keys
frozen_state	String	Frozen Status - 0: normal, not frozen - 1: frozen due to common causes - 2: frozen by the public security bureau - 3: frozen due to common causes and by the public security bureau - 4: frozen due to violations - 5: frozen due to common causes and violations - 6: frozen by the public security bureau and due to violations - 7: frozen by the public security bureau and due to common causes and violations - 8: frozen due to lack of real-name authentication - 9: frozen due to common causes and lack of real-name authentication - 10: frozen by the public security bureau and due to lack of real-name authentication

Table 4-853 PageInfo

Parameter	Type	Description
next_marker	String	Query Address for Returning to the Next Page
previous_marker	String	Return to the query address of the previous page.
current_count	Integer	Number of returned items.

Status code: 400

Table 4-854 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.


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

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ListKeypairsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKeypairsRequest request = new ListKeypairsRequest();
        request.withLimit("<limit>");
        request.withMarker("<marker>");
```



```
        try {
            ListKeypairsResponse response = client.listKeypairs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKeypairsRequest()
        request.limit = "<limit>"
        request.marker = "<marker>"
        response = client.list_keypairs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```



```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKeypairsRequest{}
    limitRequest:= "<limit>"
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListKeypairs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

See **Error Codes**.

4.2.1.4 Querying SSH Key Pair Details

Function

Querying SSH Key Pair Details

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-855 Path Parameters

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

Request Parameters

Table 4-856 Request header parameters

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

Response Parameters

Status code: 200

Table 4-857 Response body parameters

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

Table 4-858 KeypairDetail

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

Parameter	Type	Description
public_key	String	Public key information about an SSH key pair
fingerprint	String	Fingerprint information about an SSH key pair
is_key_protection	Boolean	Whether to host keys
deleted	Boolean	Tag that indicates an SSH key pair is deleted
description	String	Description of an SSH key pair
user_id	String	User that an SSH key pair belongs to
create_time	Long	Time when the SSH key pair was created. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
delete_time	Long	Time when the SSH key pair was deleted. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
update_time	Long	Time when the SSH key pair was updated. The timestamp indicates the total seconds past the start of the epoch date (January 1, 1970).
frozen_state	Integer	Frozen Status - 0: normal, not frozen - 1: frozen due to common causes - 2: frozen by the public security bureau - 3: frozen due to common causes and by the public security bureau - 4: frozen due to violations - 5: frozen due to common causes and violations - 6: frozen by the public security bureau and due to violations - 7: frozen by the public security bureau and due to common causes and violations - 8: frozen due to lack of real-name authentication - 9: frozen due to common causes and lack of real-name authentication - 10: frozen by the public security bureau and due to lack of real-name authentication
key_id	String	Specifies the key pair ID.
algorithm	String	Generation algorithm.

Status code: 400

Table 4-859 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes

Parameter	Type	Description
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

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

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ListKeypairDetailSolution {
```



```
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKeypairDetailRequest request = new ListKeypairDetailRequest();
        try {
            ListKeypairDetailResponse response = client.listKeypairDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKeypairDetailRequest()
        response = client.list_keypair_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```



```
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKeypairDetailRequest{}
    response, err := client.ListKeypairDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

See **Error Codes**.

4.2.1.5 Deleting an SSH Key Pair

Function

This API is used to delete an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-860 Path Parameters

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

Request Parameters

Table 4-861 Request header parameters

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

Response Parameters

Status code: 400

Table 4-862 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class DeleteKeypairSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteKeypairRequest request = new DeleteKeypairRequest();
        try {
            DeleteKeypairResponse response = client.deleteKeypair(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
```



```
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteKeypairRequest()
        response = client.delete_keypair(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```



```
        WithCredential(auth).
        Build())

    request := &model.DeleteKeypairRequest{}
    response, err := client.DeleteKeypair(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.1.6 Updating SSH Key Pair Description

Function

This API is used to update the description of an SSH key pair.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-863 Path Parameters

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

Request Parameters

Table 4-864 Request header parameters

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

Table 4-865 Request body parameters

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

Table 4-866 UpdateKeypairDescriptionReq

Parameter	Mandatory	Type	Description
description	Yes	String	Description

Response Parameters

Status code: 400

Table 4-867 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

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

Example Responses

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class UpdateKeypairDescriptionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateKeypairDescriptionRequest request = new UpdateKeypairDescriptionRequest();
        UpdateKeypairDescriptionRequestBody body = new UpdateKeypairDescriptionRequestBody();
        UpdateKeypairDescriptionReq keypairbody = new UpdateKeypairDescriptionReq();
        keypairbody.withDescription("description");
        body.withKeypair(keypairbody);
        request.withBody(body);
        try {
            UpdateKeypairDescriptionResponse response = client.updateKeypairDescription(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateKeypairDescriptionRequest()
        keypairbody = UpdateKeypairDescriptionReq(
            description="description"
        )
        request.body = UpdateKeypairDescriptionRequestBody(
            keypair=keypairbody
        )
        response = client.update_keypair_description(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
```



```
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateKeypairDescriptionRequest{}
    keypairbody := &model.UpdateKeypairDescriptionReq{
        Description: "description",
    }
    request.Body = &model.UpdateKeypairDescriptionRequestBody{
        Keypair: keypairbody,
    }
    response, err := client.UpdateKeypairDescription(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.1.7 Accessing the page for importing private keys

Function

Import the private key to the specified key pair.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-868 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-869 Request header parameters

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

Table 4-870 Request body parameters

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

Table 4-871 ImportPrivateKeyKeypairBean

Parameter	Mandatory	Type	Description
name	Yes	String	SSH key pair name. - The name of a key pair must be unique. - The key pair name can contain letters, digits, underscores (_), and hyphens (-) and cannot exceed 64 bytes.
user_id	No	String	User that an SSH key pair belongs to
key_protection	Yes	**ImportPrivateKeyProtection** object	Private key hosting and protection for the SSH key pair.

Table 4-872 ImportPrivateKeyProtection

Parameter	Mandatory	Type	Description
private_key	Yes	String	Private key of the imported SSH key pair.
encryption	Yes	**Encryption** object	How a private key is encrypted and stored.

Table 4-873 Encryption

Parameter	Mandatory	Type	Description
type	Yes	String	Value options: - default: The default encryption mode. Applicable to sites where KMS is not deployed. - kms: KMS encryption mode. If the KMS service is not available at the site, set this parameter to default.
kms_key_name	No	String	KMS key name. • If **type** is set to **kms**, you must enter the KMS key name or ID.
kms_key_id	No	String	KMS key ID. • If **type** is set to **kms**, you must enter the KMS key name or ID.

Response Parameters

Status code: 200

Table 4-874 Response body parameters

Parameter	Type	Description
keypair	**ImportPrivateKeyKeypairBean** object	One

Table 4-875 ImportPrivateKeyKeypairBean

Parameter	Type	Description
name	String	SSH key pair name. - The name of a key pair must be unique. - The key pair name can contain letters, digits, underscores (_), and hyphens (-) and cannot exceed 64 bytes.
user_id	String	User that an SSH key pair belongs to
key_protection	**ImportPrivateKeyProtection** object	Private key hosting and protection for the SSH key pair.

Table 4-876 ImportPrivateKeyProtection

Parameter	Type	Description
private_key	String	Private key of the imported SSH key pair.
encryption	**Encryption** object	How a private key is encrypted and stored.

Table 4-877 Encryption

Parameter	Type	Description
type	String	Value options: - default: The default encryption mode. Applicable to sites where KMS is not deployed. - kms: KMS encryption mode. If the KMS service is not available at the site, set this parameter to default.
kms_key_name	String	KMS key name. • If **type** is set to **kms**, you must enter the KMS key name or ID.
kms_key_id	String	KMS key ID. • If **type** is set to **kms**, you must enter the KMS key name or ID.

Status code: 400

Table 4-878 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes

Parameter	Type	Description
error_msg	String	Description

Example Requests

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

Example Responses

Status code: 200

Request succeeded.

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

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ImportPrivateKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```



```
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ImportPrivateKeyRequest request = new ImportPrivateKeyRequest();
        ImportPrivateKeyRequestBody body = new ImportPrivateKeyRequestBody();
        Encryption encryptionKeyProtection = new Encryption();
        encryptionKeyProtection.withType(Encryption.TypeEnum.fromValue("kms"))
            .withKmsKeyName("kps/default");
        ImportPrivateKeyProtection keyProtectionKeypair = new ImportPrivateKeyProtection();
        keyProtectionKeypair.withPrivateKey("-----BEGIN RSA PRIVATE KEY-----...")
            .withEncryption(encryptionKeyProtection);
        ImportPrivateKeyKeypairBean keypairbody = new ImportPrivateKeyKeypairBean();
        keypairbody.withName("demo2")
            .withKeyProtection(keyProtectionKeypair);
        body.withKeypair(keypairbody);
        request.withBody(body);
        try {
            ImportPrivateKeyResponse response = client.importPrivateKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
```



```
        .build()

    try:
        request = ImportPrivateKeyRequest()
        encryptionKeyProtection = Encryption(
            type="kms",
            kms_key_name="kps/default"
        )
        keyProtectionKeypair = ImportPrivateKeyProtection(
            private_key="-----BEGIN RSA PRIVATE KEY-----...",
            encryption=encryptionKeyProtection
        )
        keypairbody = ImportPrivateKeyKeypairBean(
            name="demo2",
            key_protection=keyProtectionKeypair
        )
        request.body = ImportPrivateKeyRequestBody(
            keypair=keypairbody
        )
        response = client.import_private_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ImportPrivateKeyRequest{}
    kmsKeyNameEncryption:= "kps/default"
    encryptionKeyProtection := &model.Encryption{
        Type: model.GetEncryptionTypeEnum().KMS,
        KmsKeyName: &kmsKeyNameEncryption,
    }
    keyProtectionKeypair := &model.ImportPrivateKeyProtection{
        PrivateKey: "-----BEGIN RSA PRIVATE KEY-----...",
        Encryption: encryptionKeyProtection,
    }
```



```
    keypairbody := &model.ImportPrivateKeyKeypairBean{
        Name: "demo2",
        KeyProtection: keyProtectionKeypair,
    }
    request.Body = &model.ImportPrivateKeyRequestBody{
        Keypair: keypairbody,
    }
    response, err := client.ImportPrivateKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

See **Error Codes**.

4.2.1.8 Exporting a private key

Function

Export the private key of a specified key pair.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-879 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-880 Request header parameters

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

Table 4-881 Request body parameters

Parameter	Mandatory	Type	Description
keypair	Yes	**KeypairBean** object	-

Table 4-882 KeypairBean

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the key pair.

Response Parameters

Status code: 200

Table 4-883 Response body parameters

Parameter	Type	Description
keypair	**ExportPrivateKeyKeypairBean** object	-

Table 4-884 ExportPrivateKeyKeypairBean

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

Status code: 400

Table 4-885 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

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

Example Responses

Status code: 200

Request succeeded.

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

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ExportPrivateKeySolution {

    public static void main(String[] args) {
```



```
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ExportPrivateKeyRequest request = new ExportPrivateKeyRequest();
        ExportPrivateKeyRequestBody body = new ExportPrivateKeyRequestBody();
        KeypairBean keypairbody = new KeypairBean();
        keypairbody.withName("demo2");
        body.withKeypair(keypairbody);
        request.withBody(body);
        try {
            ExportPrivateKeyResponse response = client.exportPrivateKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExportPrivateKeyRequest()
        keypairbody = KeypairBean(
            name="demo2"
```



```
        )
        request.body = ExportPrivateKeyRequestBody(
            keypair=keypairbody
        )
        response = client.export_private_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportPrivateKeyRequest{}
    keypairbody := &model.KeypairBean{
        Name: "demo2",
    }
    request.Body = &model.ExportPrivateKeyRequestBody{
        Keypair: keypairbody,
    }
    response, err := client.ExportPrivateKey(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Error response

Error Codes

See **Error Codes**.

4.2.2 Key pair task management

4.2.2.1 Binding an SSH Key Pair

Function

Binds an SSH key pair to a specified VM. (The private key of the SSH key pair for the VM is required if you want to replace the key pair, but not required if you want to reset the key pair.)

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs/associate

Table 4-886 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

Table 4-887 Request header parameters

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

Table 4-888 Request body parameters

Parameter	Mandatory	Type	Description
keypair_name	Yes	String	SSH key pair name
server	Yes	**EcsServerInfo** object	Information about the VM that the key pair is to be bound to

Table 4-889 EcsServerInfo

Parameter	Mandatory	Type	Description
id	Yes	String	ID of the VM that the SSH key pair is to be bound to (in order to replace or reset the original key pair).
auth	No	**Auth** object	(Optional) Authentication type. This parameter is required for key pair replacement but not required for key pair reset.
disable_password	No	Boolean	• true: SSH login is disabled on the VM. - false: SSH login is enabled on the VM.
port	No	Long	SSH listening port.

Table 4-890 Auth

Parameter	Mandatory	Type	Description
type	No	String	The value can be **password** or **keypair**.
key	No	String	• If type is set to password, key indicates the password. • If type is set to keypair, key indicates the private key.

Response Parameters

Status code: 202

Table 4-891 Response body parameters

Parameter	Type	Description
task_id	String	ID returned when a task is delivered
server_id	String	Specifies the ID of the bound VM.
status	String	Indicates the delivery status of a task. SUCCESS or FAILED.
error_code	String	Error code returned when a task fails to be delivered.
error_msg	String	Error information returned when a task fails to be delivered.

Status code: 400

Table 4-892 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

- ```
  {
    "keypair_name" : "newkeypair1",
    "server" : {
      "id" : "d76baba7-ef09-40a2-87ff-3eafec0696e7",
      "auth" : {
        "type" : "keypair",
        "key" : "-----BEGINRSAPRIVATEKEY-----M..."
      }
    }
  }
  ```
- ```
  {
    "keypair_name" : "newkeypair2",
    "server" : {
      "id" : "d76baba7-ef09-40a2-87ff-3eafec0696e7"
    }
  }
  ```

Example Responses

Status code: 202

OK

```
{
  "task_id" : "aee8d2fe-5484-4753-9177-5a38dc15546c"
}
```

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

- ```
  package com.huaweicloud.sdk.test;

  import com.huaweicloud.sdk.core.auth.ICredential;
  import com.huaweicloud.sdk.core.auth.BasicCredentials;
  import com.huaweicloud.sdk.core.exception.ConnectionException;
  import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
  import com.huaweicloud.sdk.core.exception.ServiceResponseException;
  import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
  import com.huaweicloud.sdk.kps.v3.*;
  import com.huaweicloud.sdk.kps.v3.model.*;


  public class AssociateKeypairSolution {

      public static void main(String[] args) {
          // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
          // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
          String ak = System.getenv("CLOUD_SDK_AK");
          String sk = System.getenv("CLOUD_SDK_SK");

          ICredential auth = new BasicCredentials()
                  .withAk(ak)
                  .withSk(sk);

          KpsClient client = KpsClient.newBuilder()
                  .withCredential(auth)
                  .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                  .build();
          AssociateKeypairRequest request = new AssociateKeypairRequest();
          AssociateKeypairRequestBody body = new AssociateKeypairRequestBody();
          Auth authServer = new Auth();
          authServer.withType(Auth.TypeEnum.fromValue("keypair"))
              .withKey("-----BEGINRSAPRIVATEKEY-----M...");
          EcsServerInfo serverbody = new EcsServerInfo();
          serverbody.withId("d76baba7-ef09-40a2-87ff-3eafec0696e7")
              .withAuth(authServer);
          body.withServer(serverbody);
          body.withKeypairName("newkeypair1");
          request.withBody(body);
          try {
              AssociateKeypairResponse response = client.associateKeypair(request);
              System.out.println(response.toString());
          } catch (ConnectionException e) {
              e.printStackTrace();
          } catch (RequestTimeoutException e) {
              e.printStackTrace();
          } catch (ServiceResponseException e) {
              e.printStackTrace();
              System.out.println(e.getHttpStatusCode());
              System.out.println(e.getRequestId());
              System.out.println(e.getErrorCode());
  ```


  ```

```
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- ```
  package com.huaweicloud.sdk.test;

  import com.huaweicloud.sdk.core.auth.ICredential;
  import com.huaweicloud.sdk.core.auth.BasicCredentials;
  import com.huaweicloud.sdk.core.exception.ConnectionException;
  import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
  import com.huaweicloud.sdk.core.exception.ServiceResponseException;
  import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
  import com.huaweicloud.sdk.kps.v3.*;
  import com.huaweicloud.sdk.kps.v3.model.*;


  public class AssociateKeypairSolution {

      public static void main(String[] args) {
          // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
          // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
          String ak = System.getenv("CLOUD_SDK_AK");
          String sk = System.getenv("CLOUD_SDK_SK");

          ICredential auth = new BasicCredentials()
                  .withAk(ak)
                  .withSk(sk);

          KpsClient client = KpsClient.newBuilder()
                  .withCredential(auth)
                  .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                  .build();
          AssociateKeypairRequest request = new AssociateKeypairRequest();
          AssociateKeypairRequestBody body = new AssociateKeypairRequestBody();
          EcsServerInfo serverbody = new EcsServerInfo();
          serverbody.withId("d76baba7-ef09-40a2-87ff-3eafec0696e7");
          body.withServer(serverbody);
          body.withKeypairName("newkeypair2");
          request.withBody(body);
          try {
              AssociateKeypairResponse response = client.associateKeypair(request);
              System.out.println(response.toString());
          } catch (ConnectionException e) {
              e.printStackTrace();
          } catch (RequestTimeoutException e) {
              e.printStackTrace();
          } catch (ServiceResponseException e) {
              e.printStackTrace();
              System.out.println(e.getHttpStatusCode());
              System.out.println(e.getRequestId());
              System.out.println(e.getErrorCode());
              System.out.println(e.getErrorMsg());
          }
      }
  }
  ```

Python

- ```
  # coding: utf-8

  import os
  from huaweicloudsdkcore.auth.credentials import BasicCredentials
  from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
  from huaweicloudsdkcore.exceptions import exceptions
  from huaweicloudsdkkps.v3 import *
  ```



```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AssociateKeypairRequest()
        authServer = Auth(
            type="keypair",
            key="-----BEGINRSAPRIVATEKEY-----M..."
        )
        serverbody = EcsServerInfo(
            id="d76baba7-ef09-40a2-87ff-3eafec0696e7",
            auth=authServer
        )
        request.body = AssociateKeypairRequestBody(
            server=serverbody,
            keypair_name="newkeypair1"
        )
        response = client.associate_keypair(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- ```
  # coding: utf-8

  import os
  from huaweicloudsdkcore.auth.credentials import BasicCredentials
  from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
  from huaweicloudsdkcore.exceptions import exceptions
  from huaweicloudsdkkps.v3 import *

  if __name__ == "__main__":
      # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
      # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
      ak = os.environ["CLOUD_SDK_AK"]
      sk = os.environ["CLOUD_SDK_SK"]

      credentials = BasicCredentials(ak, sk)

      client = KpsClient.new_builder() \
          .with_credentials(credentials) \
          .with_region(KpsRegion.value_of("<YOUR REGION>")) \
          .build()

      try:
          request = AssociateKeypairRequest()
          serverbody = EcsServerInfo(
              id="d76baba7-ef09-40a2-87ff-3eafec0696e7"
          )
  ```


  ```

```
        request.body = AssociateKeypairRequestBody(
            server=serverbody,
            keypair_name="newkeypair2"
)
        response = client.associate_keypair(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- ```
  package main

  import (
      "fmt"
      "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
      kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
      "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
      region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
  )

  func main() {
      // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
      // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
      ak := os.Getenv("CLOUD_SDK_AK")
      sk := os.Getenv("CLOUD_SDK_SK")

      auth := basic.NewCredentialsBuilder().
          WithAk(ak).
          WithSk(sk).
          Build()

      client := kps.NewKpsClient(
          kps.KpsClientBuilder().
              WithRegion(region.ValueOf("<YOUR REGION>")).
              WithCredential(auth).
              Build())

      request := &model.AssociateKeypairRequest{}
      typeAuth:= model.GetAuthTypeEnum().KEYPAIR
      keyAuth:= "-----BEGINRSAPRIVATEKEY-----M..."
      authServer := &model.Auth{
          Type: &typeAuth,
          Key: &keyAuth,
      }
      serverbody := &model.EcsServerInfo{
          Id: "d76baba7-ef09-40a2-87ff-3eafec0696e7",
          Auth: authServer,
      }
      request.Body = &model.AssociateKeypairRequestBody{
          Server: serverbody,
          KeypairName: "newkeypair1",
      }
      response, err := client.AssociateKeypair(request)
      if err == nil {
          fmt.Printf("%+v\n", response)
      } else {
          fmt.Println(err)
      }
  }
  ```
- ```
  package main
  ```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AssociateKeypairRequest{}
    serverbody := &model.EcsServerInfo{
        Id: "d76baba7-ef09-40a2-87ff-3eafec0696e7",
    }
    request.Body = &model.AssociateKeypairRequestBody{
        Server: serverbody,
        KeypairName: "newkeypair2",
    }
    response, err := client.AssociateKeypair(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
202	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.2.2 Unbind an SSH Key Pair

Function

Unbinds an SSH key pair from a specified VM and restores SSH password login.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs/disassociate

Table 4-893 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-894 Request header parameters

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

Table 4-895 Request body parameters

Parameter	Mandatory	Type	Description
server	Yes	**DisassociateEcsServerInfo** object	Information about the VM that the key pair is to be bound to

Table 4-896 DisassociateEcsServerInfo

Parameter	Mandatory	Type	Description
id	Yes	String	ID of the VM that the SSH key pair is to be bound to (in order to replace or reset the original key pair).
auth	No	**Auth** object	(Optional) Authentication type. This parameter is required for key pair replacement but not required for key pair reset.

Table 4-897 Auth

Parameter	Mandatory	Type	Description
type	No	String	The value can be **password** or **keypair**.
key	No	String	• If type is set to password, key indicates the password. • If type is set to keypair, key indicates the private key.

Response Parameters

Status code: 202

Table 4-898 Response body parameters

Parameter	Type	Description
task_id	String	ID returned when a task is delivered
server_id	String	Specifies the ID of the bound VM.
status	String	Indicates the delivery status of a task. SUCCESS or FAILED.
error_code	String	Error code returned when a task fails to be delivered.
error_msg	String	Error information returned when a task fails to be delivered.

Status code: 400

Table 4-899 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

- ```
  {
    "server" : {
      "id" : "d76baba7-ef09-40a2-87ff-3eafec0696e7",
      "auth" : {
        "type" : "keypair",
        "key" : "-----BEGINRSAPRIVATEKEY-----\nM..."
      }
    }
  }
  ```
- ```
  {
    "server" : {
      "id" : "x76baba7-ef09-40a2-87ff-3eafec0696e7"
    }
  }
  ```

Example Responses

Status code: 202

OK

```
{
  "task_id" : "aee8d2fe-5484-4753-9177-5a38dc15546c"
}
```

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

- ```
  package com.huaweicloud.sdk.test;

  import com.huaweicloud.sdk.core.auth.ICredential;
  import com.huaweicloud.sdk.core.auth.BasicCredentials;
  import com.huaweicloud.sdk.core.exception.ConnectionException;
  import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
  import com.huaweicloud.sdk.core.exception.ServiceResponseException;
  import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
  import com.huaweicloud.sdk.kps.v3.*;
  import com.huaweicloud.sdk.kps.v3.model.*;
  ```


  ```

```
public class DisassociateKeypairSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        DisassociateKeypairRequest request = new DisassociateKeypairRequest();
        DisassociateKeypairRequestBody body = new DisassociateKeypairRequestBody();
        Auth authServer = new Auth();
        authServer.withType(Auth.TypeEnum.fromValue("keypair"))
            .withKey("-----BEGINRSAPRIVATEKEY-----
M...");
        DisassociateEcsServerInfo serverbody = new DisassociateEcsServerInfo();
        serverbody.withId("d76baba7-ef09-40a2-87ff-3eafec0696e7")
            .withAuth(authServer);
        body.withServer(serverbody);
        request.withBody(body);
        try {
            DisassociateKeypairResponse response = client.disassociateKeypair(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- ```
  package com.huaweicloud.sdk.test;

  import com.huaweicloud.sdk.core.auth.ICredential;
  import com.huaweicloud.sdk.core.auth.BasicCredentials;
  import com.huaweicloud.sdk.core.exception.ConnectionException;
  import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
  import com.huaweicloud.sdk.core.exception.ServiceResponseException;
  import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
  import com.huaweicloud.sdk.kps.v3.*;
  import com.huaweicloud.sdk.kps.v3.model.*;


  public class DisassociateKeypairSolution {

      public static void main(String[] args) {
          // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
          // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
          String ak = System.getenv("CLOUD_SDK_AK");
  ```


  ```

```
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        DisassociateKeypairRequest request = new DisassociateKeypairRequest();
        DisassociateKeypairRequestBody body = new DisassociateKeypairRequestBody();
        DisassociateEcsServerInfo serverbody = new DisassociateEcsServerInfo();
        serverbody.withId("x76baba7-ef09-40a2-87ff-3eafec0696e7");
        body.withServer(serverbody);
        request.withBody(body);
        try {
            DisassociateKeypairResponse response = client.disassociateKeypair(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- ```
  # coding: utf-8

  import os
  from huaweicloudsdkcore.auth.credentials import BasicCredentials
  from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
  from huaweicloudsdkcore.exceptions import exceptions
  from huaweicloudsdkkps.v3 import *

  if __name__ == "__main__":
      # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
      # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
      ak = os.environ["CLOUD_SDK_AK"]
      sk = os.environ["CLOUD_SDK_SK"]

      credentials = BasicCredentials(ak, sk)

      client = KpsClient.new_builder() \
          .with_credentials(credentials) \
          .with_region(KpsRegion.value_of("<YOUR REGION>")) \
          .build()

      try:
          request = DisassociateKeypairRequest()
          authServer = Auth(
              type="keypair",
              key="-----BEGINRSAPRIVATEKEY-----
          M..."
          )
          serverbody = DisassociateEcsServerInfo(
              id="d76baba7-ef09-40a2-87ff-3eafec0696e7",
  ```


  ```

```
            auth=authServer
)
        request.body = DisassociateKeypairRequestBody(
            server=serverbody
)
        response = client.disassociate_keypair(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- ```
  # coding: utf-8

  import os
  from huaweicloudsdkcore.auth.credentials import BasicCredentials
  from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
  from huaweicloudsdkcore.exceptions import exceptions
  from huaweicloudsdkkps.v3 import *

  if __name__ == "__main__":
      # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
      # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
      ak = os.environ["CLOUD_SDK_AK"]
      sk = os.environ["CLOUD_SDK_SK"]

      credentials = BasicCredentials(ak, sk)

      client = KpsClient.new_builder() \
          .with_credentials(credentials) \
          .with_region(KpsRegion.value_of("<YOUR REGION>")) \
          .build()

      try:
          request = DisassociateKeypairRequest()
          serverbody = DisassociateEcsServerInfo(
              id="x76baba7-ef09-40a2-87ff-3eafec0696e7"
          )
          request.body = DisassociateKeypairRequestBody(
              server=serverbody
          )
          response = client.disassociate_keypair(request)
          print(response)
      except exceptions.ClientRequestException as e:
          print(e.status_code)
          print(e.request_id)
          print(e.error_code)
          print(e.error_msg)
  ```

Go

- ```
  package main

  import (
      "fmt"
      "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
      kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
      "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
      region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
  )

  func main() {
      // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
  ```



```
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisassociateKeypairRequest{}
    typeAuth:= model.GetAuthTypeEnum().KEYPAIR
    keyAuth:= "-----BEGINRSAPRIVATEKEY-----
M..."
    authServer := &model.Auth{
        Type: &typeAuth,
        Key: &keyAuth,
    }
    serverbody := &model.DisassociateEcsServerInfo{
        Id: "d76baba7-ef09-40a2-87ff-3eafec0696e7",
        Auth: authServer,
    }
    request.Body = &model.DisassociateKeypairRequestBody{
        Server: serverbody,
    }
    response, err := client.DisassociateKeypair(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- ```
  package main

  import (
      "fmt"
      "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
      kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
      "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
      region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
  )

  func main() {
      // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
  security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
  environment variables and decrypted during use to ensure security.
      // In this example, AK and SK are stored in environment variables for authentication. Before
  running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
  environment
      ak := os.Getenv("CLOUD_SDK_AK")
      sk := os.Getenv("CLOUD_SDK_SK")

      auth := basic.NewCredentialsBuilder().
          WithAk(ak).
          WithSk(sk).
          Build()

      client := kps.NewKpsClient(
          kps.KpsClientBuilder().
              WithRegion(region.ValueOf("<YOUR REGION>")).
              WithCredential(auth).
              Build())
  ```


  ```

```
    request := &model.DisassociateKeypairRequest{}
    serverbody := &model.DisassociateEcsServerInfo{
        Id: "x76baba7-ef09-40a2-87ff-3eafec0696e7",
    }
    request.Body = &model.DisassociateKeypairRequestBody{
        Server: serverbody,
    }
    response, err := client.DisassociateKeypair(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
202	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.2.3 Binding SSH Key Pairs in Batches

Function

This interface is used to bind new SSH key pairs to specified VMs in batches.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/keypairs/batch-associate

Table 4-900 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-901 Request header parameters

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

Table 4-902 Request body parameters

Parameter	Mandatory	Type	Description
batch_keypairs	Yes	Array of **AssociateKeypairRequestBody** objects	You can select a maximum of 10 ECSs to bind key pairs at a time. Constraints: Only the same key pair can be selected. The ECS is in the Running state and has not been bound to any key pair.

Table 4-903 AssociateKeypairRequestBody

Parameter	Mandatory	Type	Description
keypair_name	Yes	String	SSH key pair name
server	Yes	**EcsServerInfo** object	Information about the VM that the key pair is to be bound to

Table 4-904 EcsServerInfo

Parameter	Mandatory	Type	Description
id	Yes	String	ID of the VM that the SSH key pair is to be bound to (in order to replace or reset the original key pair).
auth	No	**Auth** object	(Optional) Authentication type. This parameter is required for key pair replacement but not required for key pair reset.

Parameter	Mandatory	Type	Description
disable_password	No	Boolean	• true: SSH login is disabled on the VM. - false: SSH login is enabled on the VM.
port	No	Long	SSH listening port.

Table 4-905 Auth

Parameter	Mandatory	Type	Description
type	No	String	The value can be **password** or **keypair**.
key	No	String	• If type is set to password, key indicates the password. • If type is set to keypair, key indicates the private key.

Response Parameters

Status code: 202

Table 4-906 Response body parameters

Parameter	Type	Description
tasks	Array of **TaskResponseBody** objects	Bind key pairs in batches.

Table 4-907 TaskResponseBody

Parameter	Type	Description
task_id	String	ID returned when a task is delivered
server_id	String	Specifies the ID of the bound VM.
status	String	Indicates the delivery status of a task. SUCCESS or FAILED.
error_code	String	Error code returned when a task fails to be delivered.
error_msg	String	Error information returned when a task fails to be delivered.

Status code: 400

Table 4-908 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

```
{
  "batch_keypairs" : [ {
    "keypair_name" : "1",
    "server" : {
      "id" : "fxxx16e3-74b8-4025-9852-1f451932c20c",
      "disable_password" : false,
      "auth" : {
        "type" : "password",
        "key" : "password"
      }
    }
  }, {
    "keypair_name" : "1",
    "server" : {
      "id" : "4xxxxfc4-b4bf-49c2-b983-a1811c9760c1",
      "disable_password" : false,
      "auth" : {
        "type" : "password",
        "key" : "password"
      }
    }
  } ]
}
```

Example Responses

Status code: 202

Request succeeded.

```
{
  "tasks" : [ {
    "server_id" : "xxx",
    "task_id" : "xxx",
    "status" : "SUCCESS"
  }, {
    "server_id" : "xxx",
    "status" : "Failed",
    "error_code" : "xxxx",
    "error_msg" : "xxxx"
  } ]
}
```

Status code: 400

Error response

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


SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchAssociateKeypairSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchAssociateKeypairRequest request = new BatchAssociateKeypairRequest();
        BatchAssociateKeypairRequestBody body = new BatchAssociateKeypairRequestBody();
        Auth authServer = new Auth();
        authServer.withType(Auth.TypeEnum.fromValue("password"))
            .withKey("password");
        EcsServerInfo serverBatchKeypairs = new EcsServerInfo();
        serverBatchKeypairs.withId("4xxxxfc4-b4bf-49c2-b983-a1811c9760c1")
            .withAuth(authServer)
            .withDisablePassword(false);
        Auth authServer1 = new Auth();
        authServer1.withType(Auth.TypeEnum.fromValue("password"))
            .withKey("password");
        EcsServerInfo serverBatchKeypairs1 = new EcsServerInfo();
        serverBatchKeypairs1.withId("fxxx16e3-74b8-4025-9852-1f451932c20c")
            .withAuth(authServer1)
            .withDisablePassword(false);
        List<AssociateKeypairRequestBody> listbodyBatchKeypairs = new ArrayList<>();
        listbodyBatchKeypairs.add(
            new AssociateKeypairRequestBody()
                .withKeypairName("1")
                .withServer(serverBatchKeypairs1)
        );
        listbodyBatchKeypairs.add(
            new AssociateKeypairRequestBody()
                .withKeypairName("1")
                .withServer(serverBatchKeypairs)
        );
        body.withBatchKeypairs(listbodyBatchKeypairs);
        request.withBody(body);
        try {
```



```
            BatchAssociateKeypairResponse response = client.batchAssociateKeypair(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchAssociateKeypairRequest()
        authServer = Auth(
            type="password",
            key="password"
        )
        serverBatchKeypairs = EcsServerInfo(
            id="4xxxxfc4-b4bf-49c2-b983-a1811c9760c1",
            auth=authServer,
            disable_password=False
        )
        authServer1 = Auth(
            type="password",
            key="password"
        )
        serverBatchKeypairs1 = EcsServerInfo(
            id="fxxx16e3-74b8-4025-9852-1f451932c20c",
            auth=authServer1,
            disable_password=False
        )
        listBatchKeypairsbody = [
            AssociateKeypairRequestBody(
                keypair_name="1",
                server=serverBatchKeypairs1
            ),
            AssociateKeypairRequestBody(
                keypair_name="1",
```



```
            server=serverBatchKeypairs
        )
    ]
    request.body = BatchAssociateKeypairRequestBody(
        batch_keypairs=listBatchKeypairsbody
    )
    response = client.batch_associate_keypair(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchAssociateKeypairRequest{}
    typeAuth:= model.GetAuthTypeEnum().PASSWORD
    keyAuth:= "password"
    authServer := &model.Auth{
        Type: &typeAuth,
        Key: &keyAuth,
    }
    disablePasswordServer:= false
    serverBatchKeypairs := &model.EcsServerInfo{
        Id: "4xxxxfc4-b4bf-49c2-b983-a1811c9760c1",
        Auth: authServer,
        DisablePassword: &disablePasswordServer,
    }
    typeAuth1:= model.GetAuthTypeEnum().PASSWORD
    keyAuth1:= "password"
    authServer1 := &model.Auth{
        Type: &typeAuth1,
        Key: &keyAuth1,
    }
    disablePasswordServer1:= false
    serverBatchKeypairs1 := &model.EcsServerInfo{
        Id: "fxxx16e3-74b8-4025-9852-1f451932c20c",
        Auth: authServer1,
```



```
        DisablePassword: &disablePasswordServer1,
    }
    var listBatchKeypairsbody = []model.AssociateKeypairRequestBody{
        {
            KeypairName: "1",
            Server: serverBatchKeypairs1,
        },
        {
            KeypairName: "1",
            Server: serverBatchKeypairs,
        },
    }
    request.Body = &model.BatchAssociateKeypairRequestBody{
        BatchKeypairs: listBatchKeypairsbody,
    }
    response, err := client.BatchAssociateKeypair(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
202	Request succeeded.
400	Error response

Error Codes

See **Error Codes**.

4.2.2.4 Querying Task Information

Function

Queries the execution status of the current task based on the task ID returned by the SSH key pair API.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/tasks/{task_id}

Table 4-909 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
task_id	Yes	String	The task ID.

Request Parameters

Table 4-910 Request header parameters

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

Response Parameters

Status code: 200

Table 4-911 Response body parameters

Parameter	Type	Description
server_id	String	Tenant VM ID
task_id	String	ID returned when a task is successfully delivered
task_status	String	Key pair processing state. It can be: - READY_RESET: Preparing for resetting the key pair. - RUNNING_RESET: Resetting the key pair. - FAILED_RESET: Reset failed. - SUCCESS_RESET: Reset succeeded. - READY_REPLACE: Preparing for replacing the key pair. - RUNNING_REPLACE: Replacing the key pair. - FAILED_RESET: Replacement failed. - SUCCESS_RESET: Replacement succeeded. - READY_UNBIND: Preparing for unbinding the key pair. - RUNNING_UNBIND: Unbinding the key pair. - FAILED_UNBIND: Unbinding failed. - SUCCESS_UNBIND: Unbinding succeeded.

Status code: 400

Table 4-912 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "task_id" : "aee8d2fe-5484-4753-9177-5a38dc15546c",
  "task_status" : "RUNNING_RESET",
  "server_id" : "c9aa197b-a6b6-4c33-b3a6-fa0b4ec50006"
}
```

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ListKeypairTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```



```
        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListKeypairTaskRequest request = new ListKeypairTaskRequest();
        try {
            ListKeypairTaskResponse response = client.listKeypairTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListKeypairTaskRequest()
        response = client.list_keypair_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```



```
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListKeypairTaskRequest{}
    response, err := client.ListKeypairTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.2.5 Querying Running Tasks

Function

Queries running tasks.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/running-tasks

Table 4-913 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 4-914 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of items displayed per page. Default value: 1000
offset	No	Integer	Offset of the first displayed task that is being processed.

Request Parameters

Table 4-915 Request header parameters

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

Response Parameters

Status code: 200

Table 4-916 Response body parameters

Parameter	Type	Description
total	Integer	Total number of running tasks.

Parameter	Type	Description
tasks	Array of RunningTasks objects	List of running tasks.

Table 4-917 RunningTasks

Parameter	Type	Description
task_id	String	VM ID.
operate_type	String	Operation type - FAILED_RESET: reset - FAILED_REPLACE: replace - FAILED_UNBIND: unbind
task_time	String	Task time
server_name	String	VM Name
server_id	String	VM ID.
keypair_name	String	Key pair name.

Status code: 400

Table 4-918 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "total" : 1,
  "tasks" : [ {
    "task_id" : "aee8d2fe-5484-4753-9177-5a38dc15546c",
    "operate_type" : "RUNNING_RESET",
    "task_time" : "1523342130000",
    "server_name" : "Test",
    "server_id" : "c9aa197b-a6b6-4c33-b3a6-fa0b4ec50006",
```



```
    "keypair_name" : "KeyPair-xt"
  } ]
}
```

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ListRunningTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListRunningTaskRequest request = new ListRunningTaskRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListRunningTaskResponse response = client.listRunningTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```



```
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListRunningTaskRequest()
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_running_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
```



```
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

    request := &model.ListRunningTaskRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    response, err := client.ListRunningTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.2.6 Querying Task Failure Information

Function

Queries information about failed tasks, such as binding and unbinding tasks.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/failed-tasks

Table 4-919 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Table 4-920 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of items displayed per page. Default value: 1000
offset	No	Integer	Offset of the failed task information list.

Request Parameters

Table 4-921 Request header parameters

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

Response Parameters

Status code: 200

Table 4-922 Response body parameters

Parameter	Type	Description
total	Integer	Total number of failed tasks.
tasks	Array of **FailedTasks** objects	List of failed tasks.

Table 4-923 FailedTasks

Parameter	Type	Description
task_id	String	VM ID.
operate_type	String	Operation type of a task. It can be: - FAILED_RESET: reset - FAILED_REPLACE: replace - FAILED_UNBIND: unbind
task_time	String	Task time

Parameter	Type	Description
task_error_code	String	Error code of a task failure
task_error_msg	String	Error code of a task failure
server_name	String	VM Name
server_id	String	VM ID.
keypair_name	String	Key pair name

Status code: 400

Table 4-924 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 200

OK

```
{
  "total" : 1,
  "tasks" : [ {
    "task_id" : "aee8d2fe-5484-4753-9177-5a38dc15546c",
    "operate_type" : "RUNNING_RESET",
    "task_time" : "1523342130000",
    "task_error_code" : null,
    "task_error_msg" : "Update public key error",
    "server_name" : "Test",
    "server_id" : "c9aa197b-a6b6-4c33-b3a6-fa0b4ec50006",
    "keypair_name" : "KeyPair-xt"
  } ]
}
```

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class ListFailedTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListFailedTaskRequest request = new ListFailedTaskRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListFailedTaskResponse response = client.listFailedTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
```



```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListFailedTaskRequest()
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_failed_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFailedTaskRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    response, err := client.ListFailedTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```



```
  }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.2.7 Delete all failed tasks

Function

Deletes information about failed tasks.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/failed-tasks

Table 4-925 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.

Request Parameters

Table 4-926 Request header parameters

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

Response Parameters

Status code: 400

Table 4-927 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
```



```
import com.huaweicloud.sdk.kps.v3.model.*;


public class DeleteAllFailedTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteAllFailedTaskRequest request = new DeleteAllFailedTaskRequest();
        try {
            DeleteAllFailedTaskResponse response = client.deleteAllFailedTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteAllFailedTaskRequest()
        response = client.delete_all_failed_task(request)
```



```
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAllFailedTaskRequest{}
    response, err := client.DeleteAllFailedTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

See **Error Codes**.

4.2.2.8 Delete a failed task

Function

Deletes failed tasks.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/failed-tasks/{task_id}

Table 4-928 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
task_id	Yes	String	Task ID.

Request Parameters

Table 4-929 Request header parameters

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

Response Parameters

Status code: 400

Table 4-930 Response body parameters

Parameter	Type	Description
error_code	String	Error Codes
error_msg	String	Description

Example Requests

None

Example Responses

Status code: 400

Error response

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kps.v3.region.KpsRegion;
import com.huaweicloud.sdk.kps.v3.*;
import com.huaweicloud.sdk.kps.v3.model.*;


public class DeleteFailedTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        KpsClient client = KpsClient.newBuilder()
                .withCredential(auth)
                .withRegion(KpsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteFailedTaskRequest request = new DeleteFailedTaskRequest();
        try {
            DeleteFailedTaskResponse response = client.deleteFailedTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
```



```
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkps.v3.region.kps_region import KpsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkps.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = KpsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KpsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteFailedTaskRequest()
        response = client.delete_failed_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kps "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kps/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kps.NewKpsClient(
        kps.KpsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```



```
        WithCredential(auth).
        Build())

    request := &model.DeleteFailedTaskRequest{}
    response, err := client.DeleteFailedTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	OK
400	Error response

Error Codes

See **Error Codes**.

4.3 Secret Management APIs

4.3.1 Lifecycle Management

4.3.1.1 Creating a Secret

Function

You can create a secret and store its value in its original version. Secret values are encrypted and stored in secret versions. A version can have multiple statuses. Versions without any statuses are regarded as deprecated versions and can be automatically deleted by CSMS. The initial version is marked by the **SYSCURRENT** status tag.

Constraints

You can specify a symmetric CMK to encrypt secrets. If the **kms_key_id** parameter is not specified, the default master key **csms/default** will be used to encrypt the secrets created under your account in a project. If the CMK you specified does not exist under your account, it will be automatically created. You need the **kms:dek:create** permission of the specified CMK to encrypt secret values.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/secrets

Table 4-931 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-932 Request header parameters

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

Table 4-933 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the credential to be created. Constraints: The value contains 1 to 64 characters and matches the regular expression ^[a-zA-Z0-9_-]{1,64}$.

Parameter	Mandatory	Type	Description
kms_key_id	No	String	ID of the KMS master key used to encrypt the protection secret value. If you do not specify this parameter, the secret management service uses the default master key named csms/default by default to encrypt the secret value created by your account. If the KMS you specified does not exist under your account, it will be automatically created.
description	No	String	Description of a secret. Constraint: It can contain up to 2,048 bytes.
secret_binary	No	String	Plaintext of a binary secret encoded using Base64. CSMS encrypts the plaintext and stores it in the initial version of the secret. Type: Base64-encoded binary data object Constraint: Either **secret_binary** or **secret_string** must be configured. The maximum size is 32 KB. When secret_type is set to RDS: The secret value format is as follows: "{'users': [{'name':'','password':''}]}", where name indicates the RDS DB instance account name and password indicates the RDS DB instance account password.
secret_string	No	String	Plaintext of a text secret. CSMS encrypts the plaintext and stores it in the initial version of the secret. Constraint: Either **secret_binary** or **secret_string** must be configured. The maximum size is 32 KB.

Parameter	Mandatory	Type	Description
secret_type	No	String	Secret type. The options are as follows: COMMON: Common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	No	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	No	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.

Parameter	Mandatory	Type	Description
rotation_config	No	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"rds_instance_id":"","Secret_sub_type":""}. Note: This parameter is mandatory when secret_type is set to RDS. rds_instance_id indicates the instance ID of RDS. Secret_sub_type indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
event_subscriptions	No	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.

Parameter	Mandatory	Type	Description
enterprise_project_id	No	String	This parameter is provided for enterprise users. If you are an enterprise user and have created an enterprise project, select the enterprise project to be bound to the key from the drop-down list. The default project is default. For users who have not enabled Enterprise Management, the Enterprise Project parameter is not displayed on the page and does not need to be configured.

Response Parameters

Status code: 200

Table 4-934 Response body parameters

Parameter	Type	Description
secret	**Secret** object	Secret object

Table 4-935 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.

Parameter	Type	Description
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp

Parameter	Type	Description
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Status code: 400

Table 4-936 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-937 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-938 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-939 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-940 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-941 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-942 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

Create a secret named demo and use KMS key ID 0d0466b0-e727-4d9c-b35d-f84bb474a37f to encrypt the value of this is a demo secret string.

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


Example Responses

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000,
    "secret_type" : "RDS",
    "auto_rotation" : true,
    "rotation_config" :
"{'RDSInstanceId':'63616bceef2c45409575d762a498318bin01','SecretSubType':'MultiUser'}",
    "rotation_period" : "1d",
    "rotation_time" : 1668567940000,
    "next_rotation_time" : 1668629140000,
    "event_subscriptions" : [ "pocEvent" ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Create a secret named demo and use KMS key ID 0d0466b0-e727-4d9c-b35d-f84bb474a37f to encrypt the value of this is a demo secret string.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class CreateSecretSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
```



```
                .build();
        CreateSecretRequest request = new CreateSecretRequest();
        CreateSecretRequestBody body = new CreateSecretRequestBody();
        body.withSecretString("this is a demo secret string");
        body.withKmsKeyId("0d0466b0-e727-4d9c-b35d-f84bb474a37f");
        body.withName("demo");
        request.withBody(body);
        try {
            CreateSecretResponse response = client.createSecret(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Create a secret named demo and use KMS key ID 0d0466b0-e727-4d9c-b35d-f84bb474a37f to encrypt the value of this is a demo secret string.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecretRequest()
        request.body = CreateSecretRequestBody(
            secret_string="this is a demo secret string",
            kms_key_id="0d0466b0-e727-4d9c-b35d-f84bb474a37f",
            name="demo"
        )
        response = client.create_secret(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

Create a secret named demo and use KMS key ID 0d0466b0-e727-4d9c-b35d-f84bb474a37f to encrypt the value of this is a demo secret string.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecretRequest{}
    secretStringCreateSecretRequestBody:= "this is a demo secret string"
    kmsKeyIdCreateSecretRequestBody:= "0d0466b0-e727-4d9c-b35d-f84bb474a37f"
    request.Body = &model.CreateSecretRequestBody{
        SecretString: &secretStringCreateSecretRequestBody,
        KmsKeyId: &kmsKeyIdCreateSecretRequestBody,
        Name: "demo",
    }
    response, err := client.CreateSecret(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.

Status Code	Description
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.2 Querying the Secret List

Function

This API is used to query all secrets created by the current user in the current project.

Constraints

The information returned by this API is secret metadata, which does not contain secret values.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/secrets

Table 4-943 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 4-944 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	String	Specifies the number of records on each page. Default value: 50
marker	No	String	Start secret name of pagination query. If this parameter is left blank, only the first page is queried.
event_name	No	String	When an event name is specified, only the secrets associated with the event are returned.

Request Parameters

Table 4-945 Request header parameters

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

Response Parameters

Status code: 200

Table 4-946 Response body parameters

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

Table 4-947 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret

Parameter	Type	Description
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.

Parameter	Type	Description
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Table 4-948 PageInfo

Parameter	Type	Description
next_marker	String	Query address of the next page (name of the secret at the end of the current page, name of the secret at the start of the next page).
previous_marker	String	Name of the start secret on the current page and name of the end secret on the previous page.
current_count	Integer	Number of records returned on this page.

Status code: 400

Table 4-949 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-950 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-951 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-952 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-953 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-954 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-955 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "secrets" : [ {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "secret-name-test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000,
    "secret_type" : "RDS",
    "auto_rotation" : true,
    "rotation_config" : "{'RDSInstanceId':'indstance id','SecretSubType':'MultiUser'}",
    "rotation_period" : "1d",
    "rotation_time" : 1668567940000,
    "next_rotation_time" : 1668629140000,
    "event_subscriptions" : [ "pocEvent" ]
  } ],
  "page_info" : {
    "next_marker" : "secret-name-test",
    "previous_marker" : "secret-name-test",
    "current_count" : 1
  }
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ListSecretsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSecretsRequest request = new ListSecretsRequest();
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        request.withEventName("<event_name>");
        try {
            ListSecretsResponse response = client.listSecrets(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
```



```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecretsRequest()
        request.limit = "<limit>"
        request.marker = "<marker>"
        request.event_name = "<event_name>"
        response = client.list_secrets(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecretsRequest{}
    limitRequest:= "<limit>"
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    eventNameRequest:= "<event_name>"
    request.EventName = &eventNameRequest
    response, err := client.ListSecrets(request)
    if err == nil {
```



```
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.3 Querying a Secret

Function

This API is used to query information about a specified secret.

Constraints

The information returned by this API is secret metadata, which does not contain secret values.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-956 Path Parameters

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

Request Parameters

Table 4-957 Request header parameters

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

Response Parameters

Status code: 200

Table 4-958 Response body parameters

Parameter	Type	Description
secret	**Secret** object	Secret object

Table 4-959 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret

Parameter	Type	Description
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.

Parameter	Type	Description
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Status code: 400

Table 4-960 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-961 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-962 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-963 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-964 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-965 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-966 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "secret" : {
    "id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "name" : "test",
    "state" : "ENABLED",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
    "description" : "description",
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "scheduled_delete_time" : 1581507580000,
    "secret_type" : "RDS",
    "auto_rotation" : true,
    "rotation_config" : "{'RDSInstanceId':'indstance id','SecretSubType':'MultiUser'}",
    "rotation_period" : "1d",
    "rotation_time" : 1668567940000,
    "next_rotation_time" : 1668629140000,
    "enterprise_project_id" : "0",
    "event_subscriptions" : [ "pocEvent" ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ShowSecretSolution {
```



```
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSecretRequest request = new ShowSecretRequest();
        try {
            ShowSecretResponse response = client.showSecret(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecretRequest()
        response = client.show_secret(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecretRequest{}
    response, err := client.ShowSecret(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.

Status Code	Description
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.4 Updating a Secret

Function

Updates the metadata of a specified secret.

Constraints

This API can only be used to modify secret metadata. It cannot modify the secret value.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-967 Path Parameters

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

Request Parameters

Table 4-968 Request header parameters

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

Table 4-969 Request body parameters

Parameter	Mandatory	Type	Description
kms_key_id	No	String	ID of the KMS CMK used to encrypt a secret value. If the CMK of a secret is updated, only the secret versions created after the update will be encrypted using the new CMK. The secret versions earlier than the update are still decrypted using the old CMK ID.
description	No	String	Description of a secret. Constraint: It can contain up to 2,048 bytes.
auto_rotation	No	Boolean	Automatic rotation The value can be true (enabled) or false (disabled).
rotation_period	No	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.

Parameter	Mandatory	Type	Description
event_subscriptions	No	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.

Response Parameters

Status code: 200

Table 4-970 Response body parameters

Parameter	Type	Description
secret	**Secret** object	Secret object

Table 4-971 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.

Parameter	Type	Description
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Status code: 400

Table 4-972 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-973 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-974 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-975 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-976 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-977 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-978 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

Updating the secret KMS key ID to **test** and description to **update description**

```
{
  "kms_key_id" : "test",
  "description" : "update description",
  "event_subscriptions" : [ "pocEvent2" ]
}
```

Example Responses

Status code: 200

Request succeeded.

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



```
  "secret_type" : "RDS",
  "auto_rotation" : true,
  "rotation_config" : "{'RDSInstanceId':'indstance id','SecretSubType':'MultiUser'}",
  "rotation_period" : "1d",
  "rotation_time" : 1668567940000,
  "next_rotation_time" : 1668629140000,
  "event_subscriptions" : [ "pocEvent" ]
 }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the secret KMS key ID to **test** and description to **update description**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateSecretSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateSecretRequest request = new UpdateSecretRequest();
        UpdateSecretRequestBody body = new UpdateSecretRequestBody();
        List<String> listbodyEventSubscriptions = new ArrayList<>();
        listbodyEventSubscriptions.add("pocEvent2");
        body.withEventSubscriptions(listbodyEventSubscriptions);
        body.withDescription("update description");
        body.withKmsKeyId("test");
        request.withBody(body);
        try {
            UpdateSecretResponse response = client.updateSecret(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
```



```
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updating the secret KMS key ID to **test** and description to **update description**

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecretRequest()
        listEventSubscriptionsbody = [
            "pocEvent2"
        ]
        request.body = UpdateSecretRequestBody(
            event_subscriptions=listEventSubscriptionsbody,
            description="update description",
            kms_key_id="test"
        )
        response = client.update_secret(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updating the secret KMS key ID to **test** and description to **update description**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)
```



```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecretRequest{}
    var listEventSubscriptionsbody = []string{
        "pocEvent2",
    }
    descriptionUpdateSecretRequestBody:= "update description"
    kmsKeyIdUpdateSecretRequestBody:= "test"
    request.Body = &model.UpdateSecretRequestBody{
        EventSubscriptions: &listEventSubscriptionsbody,
        Description: &descriptionUpdateSecretRequestBody,
        KmsKeyId: &kmsKeyIdUpdateSecretRequestBody,
    }
    response, err := client.UpdateSecret(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.

Status Code	Description
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.5 Deleting a Secret Immediately

Function

The specified secret is deleted immediately and cannot be restored.

Constraints

Secrets deleted via this API cannot be restored.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-979 Path Parameters

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

Request Parameters

Table 4-980 Request header parameters

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

Response Parameters

Status code: 400

Table 4-981 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-982 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-983 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-984 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-985 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-986 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-987 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;
```



```
public class DeleteSecretSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteSecretRequest request = new DeleteSecretRequest();
        try {
            DeleteSecretResponse response = client.deleteSecret(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecretRequest()
        response = client.delete_secret(request)
        print(response)
    except exceptions.ClientRequestException as e:
```



```
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecretRequest{}
    response, err := client.DeleteSecret(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.

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

Error Codes

See **Error Codes**.

4.3.1.6 Restoring a Secret Object

Function

Restore the secret object by uploading the secret backup file.

Constraints

The information returned by this API is secret metadata, which does not contain secret values.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/secrets/restore

Table 4-988 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-989 Request header parameters

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

Table 4-990 Request body parameters

Parameter	Mandatory	Type	Description
secret_blob	Yes	String	Secret backup file obtained after a specified secret object is backed up. The backup file contains all secret version information. The backup file is encrypted and encoded and cannot be directly read.

Response Parameters

Status code: 200

Table 4-991 Response body parameters

Parameter	Type	Description
secret	**Secret** object	Secret object

Table 4-992 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret

Parameter	Type	Description
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.

Parameter	Type	Description
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Status code: 400

Table 4-993 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-994 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-995 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-996 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-997 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-998 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-999 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

Upload the secret backup file.

```
{
  "secret_blob" :
")CloudSecretManagementBackupV1.huaweicloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTUwNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKzNaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-Cbmf6qpJF61kXKHX4TBe6-oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxUXpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7GugcNi7jTKuvxu4l2Q2v7wnk"
}
```

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Upload the secret backup file.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```



```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class UploadSecretBlobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UploadSecretBlobRequest request = new UploadSecretBlobRequest();
        UploadSecretBlobRequestBody body = new UploadSecretBlobRequestBody();

body.withSecretBlob("")CloudSecretManagementBackupV1.huaweicloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTUwNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKzNaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-Cbmf6qpJF61kXKHX4TBe6-oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxUXpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7GugcNi7jTKuvxu4l2Q2v7wnk");
        request.withBody(body);
        try {
            UploadSecretBlobResponse response = client.uploadSecretBlob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Upload the secret backup file.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
```



```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UploadSecretBlobRequest()
        request.body = UploadSecretBlobRequestBody(

secret_blob="")CloudSecretManagementBackupV1.huaweicloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTUwNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKzNaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-Cbmf6qpJF61kXKHX4TBe6-oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxUXpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7GugcNi7jTKuvxu4l2Q2v7wnk"
        )
        response = client.upload_secret_blob(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Upload the secret backup file.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```



```
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UploadSecretBlobRequest{}
    request.Body = &model.UploadSecretBlobRequestBody{
        SecretBlob:
")CloudSecretManagementBackupV1.huaweicloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTUwNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKzNaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-Cbmf6qpJF61kXKHX4TBe6-oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxUXpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7GugcNi7jTKuvxu4l2Q2v7wnk",
    }
    response, err := client.UploadSecretBlob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.7 Downloading Secret Backup

Function

Download the backup file of a specified secret.

Constraints

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

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1000 Path Parameters

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

Request Parameters

Table 4-1001 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1002 Response body parameters

Parameter	Type	Description
secret_blob	String	Secret backup file contains all secret version information. The backup file is encrypted and encoded and cannot be directly read.

Status code: 400

Table 4-1003 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1004 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1005 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1006 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1007 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1008 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1009 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

{
  "secret_blob" :
")CloudSecretManagementBackupV1.huaweicloud.comeyJraWQiOiI5ZjNlZmRjNS0zZjVlLTRiZWQtYThkMS05NjE2ZTUwNDQzYWIiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBLU9BRVAtMjU2In0.CtrOcFMSeW_qMdQjgKzNaWtC6hkSTdjOSMSr2IOKNa8OpbJH8rOaCt9l4LYLHKw8CF70YLWOODgaYrLiWuHgdR-O9hlALkT6CbXxJ-Cbmf6qpJF61kXKHX4TBe6-oV8t4PaPaSDDR_oeyt4Xl2EOOlHxs9PnU1st9Fkd7wOHNa4ueM16Ze5ICEdQK3cN1hnelid0zlb1qq58KhsSroNeI8B5RnoYDB-0eiFWD0XWJLppgkLnewXpuPLmLN_c558yUQ0u0VoUyBGB6EFePPbbT-Z1_LUCSRyiP9Y2S0Vz5jzzeabWZ4vZkW8JX57Wc-onHplUpsUUpIqcdHLjp40NEQ.VtA6Sg--


```
jeA1QavYxY9z7Q.Mr6dLyontoJCaDaRFMAYg_qUdEPzd-aIIrCHWH7wvYayNpSFUjR5QJd3XPpGGy93y22jN-
DoHZHclgMeureQwKq39QQF0xIdRqhOR2Lxy69PkgRaNtpz7ikLOlsbjh1wd7mbSmyolsK_0t1X9OlvOSmUMjxU
XpXLzqLXxPY0R_MUxEanHb3V_vsLArF9sN1X7Km-
fdUKXTV1EzVUq1eC5aSYqg3rGkLHPHG6lPXOetPWNsVCE1bX0Voh0XnlyFLSSoYzX45l04hR8JXgcP42FXfD7Gug
cNi7jTKuvxu4l2Q2v7wnk"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class DownloadSecretBlobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DownloadSecretBlobRequest request = new DownloadSecretBlobRequest();
        try {
            DownloadSecretBlobResponse response = client.downloadSecretBlob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```



```
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DownloadSecretBlobRequest()
        response = client.download_secret_blob(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DownloadSecretBlobRequest{}
    response, err := client.DownloadSecretBlob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```



```
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.8 Creating a Scheduled Secret Deletion Task

Function

Specify the deletion delay time and create a scheduled task for deleting secrets. You can set the deletion delay time to 7 to 30 days.

Constraints

For a secret in the **Scheduled deletion** state, its metadata cannot be updated and its secret cannot be checked.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1010 Path Parameters

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

Request Parameters

Table 4-1011 Request header parameters

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

Table 4-1012 Request body parameters

Parameter	Mandatory	Type	Description
recovery_window_in_days	Yes	Integer	Delay in deleting a secret in a scheduled task. Constraint: The value range is 7 to 30 days. Default value: 30

Response Parameters

Status code: 200

Table 4-1013 Response body parameters

Parameter	Type	Description
secret	**Secret** object	Secret object

Table 4-1014 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name

Parameter	Type	Description
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.

Parameter	Type	Description
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Status code: 400

Table 4-1015 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1016 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1017 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1018 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1019 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1020 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1021 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

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

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

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

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

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class DeleteSecretForScheduleSolution {

    public static void main(String[] args) {
```



```
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteSecretForScheduleRequest request = new DeleteSecretForScheduleRequest();
        DeleteSecretForScheduleRequestBody body = new DeleteSecretForScheduleRequestBody();
        body.withRecoveryWindowInDays(15);
        request.withBody(body);
        try {
            DeleteSecretForScheduleResponse response = client.deleteSecretForSchedule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

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

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecretForScheduleRequest()
        request.body = DeleteSecretForScheduleRequestBody(
            recovery_window_in_days=15
        )
```



```
        response = client.delete_secret_for_schedule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

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

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecretForScheduleRequest{}
    request.Body = &model.DeleteSecretForScheduleRequestBody{
        RecoveryWindowInDays: int32(15),
    }
    response, err := client.DeleteSecretForSchedule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.9 Canceling a Scheduled Secret Deletion Task

Function

The scheduled secret deletion task is canceled, and the secret object is restored to the available state.

Constraints

This API can only be called to delete secrets in the **Scheduled deletion** state.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1022 Path Parameters

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

Request Parameters

Table 4-1023 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1024 Response body parameters

Parameter	Type	Description
secret	**Secret** object	Secret object

Table 4-1025 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.

Parameter	Type	Description
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp

Parameter	Type	Description
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Status code: 400

Table 4-1026 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1027 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1028 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1029 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1030 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1031 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1032 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "secret" : {
```



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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class RestoreSecretSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        RestoreSecretRequest request = new RestoreSecretRequest();
        try {
            RestoreSecretResponse response = client.restoreSecret(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RestoreSecretRequest()
        response = client.restore_secret(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestoreSecretRequest{}
    response, err := client.RestoreSecret(request)
```



```
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.1.10 Rotating a Secret

Function

Executes rotation secrets immediately. Creates a new secret version in the specified secret to encrypt and store secret values randomly generated in the background. At the same time, the newly created secret version is marked as SYSCURRENT.

Constraints

The RotateSecret API does not support rotation of common secrets. A user account has the following rights: Change the RDS database password Query key details Query the key list Create a DEK Decrypt the DEK

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1033 Path Parameters

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

Request Parameters

Table 4-1034 Request header parameters

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

Response Parameters

Status code: 400

Table 4-1035 Response body parameters

Parameter	Type	Description
error	ErrorDetail object	Error message.

Table 4-1036 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1037 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1038 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1039 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1040 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1041 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1042 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1043 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1044 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1045 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1046 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1047 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1048 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "rotate" : {
    "version_id" : "V2",
    "secret_name" : "test"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class RotateSecretSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```



```
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        RotateSecretRequest request = new RotateSecretRequest();
        try {
            RotateSecretResponse response = client.rotateSecret(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RotateSecretRequest()
        response = client.rotate_secret(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
```



```
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RotateSecretRequest{}
    response, err := client.RotateSecret(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.2 Secret Version Management

4.3.2.1 Updating the Secret Version

Function

Currently, the validity period of a specified secret version can be updated. Only secrets in the ENABLED state can be updated. If the associated subscription events contain the basic event type Version Expiration, only one event notification is triggered each time the version validity period is updated.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1049 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name
version_id	Yes	String	Version identifier of a secret.

Request Parameters

Table 4-1050 Request header parameters

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

Table 4-1051 Request body parameters

Parameter	Mandatory	Type	Description
expire_time	No	Long	Expiration time of the secret version. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time. When the secret subscribes to the version expiration event type, the validity period is determined based on this parameter. This parameter is left blank by default.

Response Parameters

Status code: 200

Table 4-1052 Response body parameters

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

Table 4-1053 VersionMetadata

Parameter	Type	Description
id	String	Secret version ID, which is unique under a secret object.
create_time	Long	Time when a secret version was created. The timestamp indicates the total seconds since January 1, 1970.
expire_time	Long	Expiration time of the secret version. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time. When the secret subscribes to the version expiration event type, the validity period is determined based on this parameter. This parameter is left blank by default.
kms_key_id	String	ID of the KMS CMK used to encrypt a secret version value.
secret_name	String	Secret name

Parameter	Type	Description
version_stages	Array of strings	Secret version status list. Every version status is unique under a secret. If you add a status tag in use to a new version, the tag will be automatically removed from the old version. If **version_stage** is not specified, the temporary tag **SYSCURRENT** will be added to this version.

Status code: 400

Table 4-1054 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1055 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1056 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1057 Response body parameters

Parameter	Type	Description
error_code	String	Error codes

Parameter	Type	Description
error_msg	String	Error description

Status code: 500

Table 4-1058 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1059 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1060 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "version_metadata" : {
    "id" : "v3",
    "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
```



```
    "create_time" : 1581507580000,
    "secret_name" : "secret-name-demo",
    "version_stages" : [ "SYSCURRENT" ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class UpdateVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateVersionRequest request = new UpdateVersionRequest();
        UpdateVersionRequestBody body = new UpdateVersionRequestBody();
        request.withBody(body);
        try {
            UpdateVersionResponse response = client.updateVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8
```



```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateVersionRequest()
        request.body = UpdateVersionRequestBody(
        )
        response = client.update_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateVersionRequest{}
    request.Body = &model.UpdateVersionRequestBody{
    }
    response, err := client.UpdateVersion(request)
```



```
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.2.2 Querying the Secret Version and Value

Function

This API is used to query information about a specified secret version and plaintext secret values in the version. Only secrets in the ENABLED state can be queried. You can use /v1/{project_id}/secrets/{secret_name}/versions/latest (that is, set {version_id} in the URL of the current interface to latest) to access the secret value of the latest version.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1061 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
secret_name	Yes	String	Secret name
version_id	Yes	String	Version identifier of a secret.

Request Parameters

Table 4-1062 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1063 Response body parameters

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

Table 4-1064 Version

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

Table 4-1065 VersionMetadata

Parameter	Type	Description
id	String	Secret version ID, which is unique under a secret object.
create_time	Long	Time when a secret version was created. The timestamp indicates the total seconds since January 1, 1970.
expire_time	Long	Expiration time of the secret version. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time. When the secret subscribes to the version expiration event type, the validity period is determined based on this parameter. This parameter is left blank by default.
kms_key_id	String	ID of the KMS CMK used to encrypt a secret version value.
secret_name	String	Secret name
version_stages	Array of strings	Secret version status list. Every version status is unique under a secret. If you add a status tag in use to a new version, the tag will be automatically removed from the old version. If **version_stage** is not specified, the temporary tag **SYSCURRENT** will be added to this version.

Status code: 400

Table 4-1066 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1067 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1068 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1069 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1070 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1071 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1072 Response body parameters

Parameter	Type	Description
error_code	String	Error codes

Parameter	Type	Description
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "version" : {
    "version_metadata" : {
      "id" : "v6",
      "kms_key_id" : "b168fe00ff56492495a7d22974df2d0b",
      "create_time" : 1581507580000,
      "secret_name" : "secret-name-demo",
      "version_stages" : [ "pending", "used" ]
    },
    "secret_string" : {
      "\\\"demo_key\\\":\\\"demo_value\\\"" : null
    }
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ShowSecretVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);
```



```
        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSecretVersionRequest request = new ShowSecretVersionRequest();
        try {
            ShowSecretVersionResponse response = client.showSecretVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecretVersionRequest()
        response = client.show_secret_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)
```



```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecretVersionRequest{}
    response, err := client.ShowSecretVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.3 Secret Version Status Management

4.3.3.1 Updating the Version Status of a Secret

Function

Update the version status of a secret.

Constraints

- The version status of a secret can uniquely identify only one version under the same secret. If you add a status in use to a new version, the status will be automatically removed from the old version. A version without any version status identifier is considered deprecated and can be automatically deleted by CSMS. * A secret can have up to 12 version statuses. A status can be used for only one version. **SYSCURRENT** and **SYSPREVIOUS** are built-in secret statuses.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1073 Path Parameters

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

Request Parameters

Table 4-1074 Request header parameters

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

Table 4-1075 Request body parameters

Parameter	Mandatory	Type	Description
version_id	Yes	String	Version identifier of a secret.

Response Parameters

Status code: 200

Table 4-1076 Response body parameters

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

Table 4-1077 Stage

Parameter	Type	Description
name	String	Status of a secret version. Constraint: The value can contain up to 64 characters.
update_time	Long	Time when a secret version status was updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
secret_name	String	Secret name.
version_id	String	Version identifier of a secret

Status code: 400

Table 4-1078 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1079 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1080 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1081 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1082 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1083 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1084 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

Updates the version status of a secret. The version is version_id.

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

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Updates the version status of a secret. The version is version_id.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class UpdateSecretStageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
```



```
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateSecretStageRequest request = new UpdateSecretStageRequest();
        UpdateSecretStageRequestBody body = new UpdateSecretStageRequestBody();
        body.withVersionId("version_id");
        request.withBody(body);
        try {
            UpdateSecretStageResponse response = client.updateSecretStage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updates the version status of a secret. The version is version_id.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecretStageRequest()
        request.body = UpdateSecretStageRequestBody(
            version_id="version_id"
        )
        response = client.update_secret_stage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```



```
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updates the version status of a secret. The version is version_id.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecretStageRequest{}
    request.Body = &model.UpdateSecretStageRequestBody{
        VersionId: "version_id",
    }
    response, err := client.UpdateSecretStage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.

Status Code	Description
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.3.2 Querying the Version Status of a Secret

Function

Queries the version information of a specified secret version status flag.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1085 Path Parameters

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

Request Parameters

Table 4-1086 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1087 Response body parameters

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

Table 4-1088 Stage

Parameter	Type	Description
name	String	Status of a secret version. Constraint: The value can contain up to 64 characters.
update_time	Long	Time when a secret version status was updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
secret_name	String	Secret name.
version_id	String	Version identifier of a secret

Status code: 400

Table 4-1089 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1090 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1091 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1092 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1093 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1094 Response body parameters

Parameter	Type	Description
error_code	String	Error codes

Parameter	Type	Description
error_msg	String	Error description

Status code: 504

Table 4-1095 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "stage" : {
    "name" : "name",
    "version_id" : "v20",
    "update_time" : 1581507580000,
    "secret_name" : "secret-name-demo"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ShowSecretStageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```



```
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSecretStageRequest request = new ShowSecretStageRequest();
        try {
            ShowSecretStageResponse response = client.showSecretStage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecretStageRequest()
        response = client.show_secret_stage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecretStageRequest{}
    response, err := client.ShowSecretStage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.

Status Code	Description
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.3.3 Deleting the Version Status of a Secret

Function

Deletes the status of a specified secret version.

Constraints

SYSCURRENT and **SYSPREVIOUS** are built-in statuses and cannot be deleted.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1096 Path Parameters

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

Request Parameters

Table 4-1097 Request header parameters

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

Response Parameters

Status code: 400

Table 4-1098 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1099 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1100 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1101 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1102 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1103 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1104 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class DeleteSecretStageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteSecretStageRequest request = new DeleteSecretStageRequest();
        try {
            DeleteSecretStageResponse response = client.deleteSecretStage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```



```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecretStageRequest()
        response = client.delete_secret_stage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecretStageRequest{}
    response, err := client.DeleteSecretStage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.4 Secret Tag Management

4.3.4.1 Querying a Secret Instance

Function

This API is used to query secret instances. Filters user secret by tag and returns the secret list.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1105 Path Parameters

Parameter	Mandatory	Type	Description
resource_instances	Yes	String	Its value is **resource_instances**.
project_id	Yes	String	Project ID

Request Parameters

Table 4-1106 Request header parameters

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

Table 4-1107 Request body parameters

Parameter	Mandatory	Type	Description
limit	No	String	Specifies the number of records to be queried. This parameter does not need to be set when action is set to count. If action is set to filter, the default value is 10. The value of limit is in the ranges 1 to 1,000.
offset	No	String	The index location. The query starts from the next resource of the specified location. When querying data on the first page, set this parameter to the value in the response body of the previous page. If action is set to count, you do not need to set this parameter. If the **action** is set to **filter**, **offset** defaults to **0**. The value of offset must be a number and cannot be negative.
action	No	String	Operation type. It can be: - filter - count: Count total records.
tags	No	Array of **Tag** objects	list of tags, including tag keys and tag values. The maximum number is 10.

Parameter	Mandatory	Type	Description
matches	No	Array of **TagItem** objects	Key-value pair to be matched - **key** indicates the search field. Currently, its value is **resource_name**, indicating that only the secret name can be searched. - **value** indicates the field for fuzzy match. It can contain up to 255 characters. If it is left blank, a null value is returned.
sequence	No	String	36-byte sequence number of a request message. Example: 919c82d4-8046-4722-9094-35c3c6524cff.

Table 4-1108 Tag

Parameter	Mandatory	Type	Description
key	No	String	The tag key.
values	No	Array of strings	Indicates the tag value set. Constraints: A maximum of 10 values can be contained. Tag values in a tag list must be unique. If the value list is left empty, any tag value can be matched. Multiple values in the tag list are in the OR relationship. If the key meets the requirements, a value in the request is matched.

Table 4-1109 TagItem

Parameter	Mandatory	Type	Description
key	Yes	String	Name of a tag. The tag keys of a secret cannot have duplicate values. A tag key can be used for multiple secrets. You can add a maximum of 20 tags to a credential. Constraints: The value contains 1 to 128 characters and matches the regular expression "^((?!\s)(?!*sys*)[\p{L}\p{Z}\p{N}_.:=+\-@]*)(?<!\s)$".
value	No	String	Tag value. Constraints: The value contains a maximum of 255 characters and matches the regular expression "^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$".

Response Parameters

Status code: 200

Table 4-1110 Response body parameters

Parameter	Type	Description
resources	Array of **ActionResources** objects	Resource instance list. For details, see the data structure of the **resource** field.
total_count	Integer	Total number of records.

Table 4-1111 ActionResources

Parameter	Type	Description
resource_id	String	Resource ID
resource_detail	**Secret** object	Secret object
resource_name	String	Resource name, which is a null string by default.

Parameter	Type	Description
tags	Array of TagItem objects	Lists of tags. If there is no tag, the array is empty by default.

Table 4-1112 Secret

Parameter	Type	Description
id	String	Resource identifier of a secret
name	String	Secret name
state	String	Secret status. Its value can be: **ENABLED DISABLED PENDING_DELETE FROZEN**
kms_key_id	String	ID of the KMS CMK used to encrypt a secret value.
description	String	Description of a secret
create_time	Long	Secret creation time. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
update_time	Long	Time when a secret was last updated. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time.
scheduled_delete_time	Long	Time when a secret is scheduled to be deleted. The value is a timestamp, that is, the total number of seconds on January 1, 1970 to the current time. If the secret is not in the deletion plan, the value of this parameter is null.
secret_type	String	Secret type The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
auto_rotation	Boolean	Automatic rotation The value can be true (enabled) or false (disabled). The default value is false.

Parameter	Type	Description
rotation_period	String	Rotation period Constraints: 6 hours - 8,760 hours (365 days) Type: Integer[unit]. Integer indicates the time length. unit indicates the time unit, which can be d (day), h (hour), m (minute), or s (second). For example, 1d indicates one day, and 24h also indicates one day. Note: This parameter is mandatory when automatic rotation is enabled.
rotation_config	String	Rotation configuration Constraints: The value contains a maximum of 1,024 characters. If secret_type is set to RDS, set this parameter to {"RDSInstanceId":"","SecretSubType":""}. Note: This parameter is mandatory when secret_type is set to RDS. RDSInstanceId indicates the RDS DB instance ID. SecretSubType indicates the rotation subtype. The value can be SingleUser or MultiUser. SingleUser: The single-user mode is used. The password of a specified account is reset each time the account is rotated. MultiUser: The dual-user mode is used. SYSCURRENT and SYSPREVIOUS reference one of the accounts. During secret rotation, the account password referenced by SYSPREVIOUS is reset to a new random password. Subsequently, secrets exchange SYSCURRENT and SYSPREVIOUS references to the RDS account.
rotation_time	Long	Rotation timestamp
next_rotation_time	Long	Next rotation timestamp
event_subscriptions	Array of strings	List of events subscribed to by secrets. Currently, only one event can be subscribed to. When a basic event contained in an event is triggered, a notification message is sent to the notification topic corresponding to the event.
enterprise_project_id	String	Enterprise project ID

Table 4-1113 TagItem

Parameter	Type	Description
key	String	Name of a tag. The tag keys of a secret cannot have duplicate values. A tag key can be used for multiple secrets. You can add a maximum of 20 tags to a credential. Constraints: The value contains 1 to 128 characters and matches the regular expression "^((?!\s)(?!*sys*)[\p{L}\p{Z}\p{N}_.:=+\-@]*)(?<!\s)$".
value	String	Tag value. Constraints: The value contains a maximum of 255 characters and matches the regular expression "^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$".

Status code: 400

Table 4-1114 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1115 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1116 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1117 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1118 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1119 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1120 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1121 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1122 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1123 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1124 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1125 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1126 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1127 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

Filters user secrets by tag and returns the secret list.

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

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Filters user secrets by tag and returns the secret list.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class ListResourceInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListResourceInstancesRequest request = new ListResourceInstancesRequest();
        ListResourceInstancesRequestBody body = new ListResourceInstancesRequestBody();
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("val1");
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        body.withTags(listbodyTags);
        body.withAction("filter");
        request.withBody(body);
        try {
            ListResourceInstancesResponse response = client.listResourceInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

Filters user secrets by tag and returns the secret list.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListResourceInstancesRequest()
        listValuesTags = [
            "val1"
        ]
        listTagsbody = [
            Tag(
                key="key1",
                values=listValuesTags
            )
        ]
        request.body = ListResourceInstancesRequestBody(
            tags=listTagsbody,
            action="filter"
        )
        response = client.list_resource_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Filters user secrets by tag and returns the secret list.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```



```
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListResourceInstancesRequest{}
    var listValuesTags = []string{
        "val1",
    }
    keyTags:= "key1"
    var listTagsbody = []model.Tag{
        {
            Key: &keyTags,
            Values: &listValuesTags,
        },
    }
    request.Body = &model.ListResourceInstancesRequestBody{
        Tags: &listTagsbody,
        Action: "filter",
    }
    response, err := client.ListResourceInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	Failed to complete the request because the server receives an invalid response from an upstream server.

Status Code	Description
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.4.2 Adding or Deleting Secret Tags in Batches

Function

This API is used to add or delete secret tags in batches.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1128 Path Parameters

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

Request Parameters

Table 4-1129 Request header parameters

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

Table 4-1130 Request body parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of **TagItem** objects	List of tags, including tag keys and tag values.
action	Yes	String	Operation type. It can be: create or delete
sequence	No	String	36-byte sequence number of a request message. Example: 919c82d4-8046-4722-9094-35c3c6524cff.

Table 4-1131 TagItem

Parameter	Mandatory	Type	Description
key	Yes	String	Name of a tag. The tag keys of a secret cannot have duplicate values. A tag key can be used for multiple secrets. You can add a maximum of 20 tags to a credential. Constraints: The value contains 1 to 128 characters and matches the regular expression "^((?!\s)(?!*sys*)[\p{L}\p{Z}\p{N}_.:=+\-@]*)(?<!\s)$".
value	No	String	Tag value. Constraints: The value contains a maximum of 255 characters and matches the regular expression "^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$".

Response Parameters

Status code: 400

Table 4-1132 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1133 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1134 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1135 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1136 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1137 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1138 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1139 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1140 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1141 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1142 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1143 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1144 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1145 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

This API is used to add secret tags in batches.

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

This API is used to add secret tags in batches.


```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateOrDeleteTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateOrDeleteTagsRequest request = new BatchCreateOrDeleteTagsRequest();
        BatchCreateOrDeleteTagsRequestBody body = new BatchCreateOrDeleteTagsRequestBody();
        List<TagItem> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagItem()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new TagItem()
                .withKey("key2")
                .withValue("value2")
        );
        body.withAction("create");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateOrDeleteTagsResponse response = client.batchCreateOrDeleteTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

This API is used to add secret tags in batches.


```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateOrDeleteTagsRequest()
        listTagsbody = [
            TagItem(
                key="key1",
                value="value1"
            ),
            TagItem(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchCreateOrDeleteTagsRequestBody(
            action="create",
            tags=listTagsbody
        )
        response = client.batch_create_or_delete_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

This API is used to add secret tags in batches.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
```



```
    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateOrDeleteTagsRequest{}
    valueTags:= "value1"
    valueTags1:= "value2"
    var listTagsbody = []model.TagItem{
        {
            Key: "key1",
            Value: &valueTags,
        },
        {
            Key: "key2",
            Value: &valueTags1,
        },
    }
    request.Body = &model.BatchCreateOrDeleteTagsRequestBody{
        Action: "create",
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateOrDeleteTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	No Content
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.4.3 Querying Secret Tags

Function

This API is used to query secret tags.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1146 Path Parameters

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

Request Parameters

Table 4-1147 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1148 Response body parameters

Parameter	Type	Description
tags	Array of **TagItem** objects	list of tags, including tag keys and tag values. - key: Tag key. A secret can have a maximum of 20 keys, and each of them is unique and cannot be left blank. A key cannot have duplicate values. The key can contain a maximum of 128 characters. - value: Tag value. Each value can contain a maximum of 255 characters. The relationship between values is AND.
sys_tags	Array of **TagItem** objects	Specifies the system tag list.

Table 4-1149 TagItem

Parameter	Type	Description
key	String	Name of a tag. The tag keys of a secret cannot have duplicate values. A tag key can be used for multiple secrets. You can add a maximum of 20 tags to a credential. Constraints: The value contains 1 to 128 characters and matches the regular expression "^((?!\s)(?!*sys*)[\p{L}\p{Z}\p{N}_.:=+\-@]*)(?<!\s)$".
value	String	Tag value. Constraints: The value contains a maximum of 255 characters and matches the regular expression "^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$".

Status code: 400

Table 4-1150 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1151 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.

Parameter	Type	Description
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1152 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1153 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1154 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1155 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1156 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1157 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1158 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1159 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1160 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1161 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1162 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1163 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;
```



```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ListSecretTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSecretTagsRequest request = new ListSecretTagsRequest();
        try {
            ListSecretTagsResponse response = client.listSecretTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
```



```
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecretTagsRequest()
        response = client.list_secret_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecretTagsRequest{}
    response, err := client.ListSecretTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.

Status Code	Description
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.4.4 Querying Secret Tags

Function

Add a secret tag.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1164 Path Parameters

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

Request Parameters

Table 4-1165 Request header parameters

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

Table 4-1166 Request body parameters

Parameter	Mandatory	Type	Description
tag	Yes	**TagItem** object	

Table 4-1167 TagItem

Parameter	Mandatory	Type	Description
key	Yes	String	Name of a tag. The tag keys of a secret cannot have duplicate values. A tag key can be used for multiple secrets. You can add a maximum of 20 tags to a credential. Constraints: The value contains 1 to 128 characters and matches the regular expression "^((?!\s)(?!*sys*)[\p{L}\p{Z}\p{N}_.:=+\-@]*)(?<!\s)$".
value	No	String	Tag value. Constraints: The value contains a maximum of 255 characters and matches the regular expression "^([\p{L}\p{Z}\p{N}_.:\/=+\-@]*)$".

Response Parameters

Status code: 400

Table 4-1168 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1169 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1170 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1171 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1172 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1173 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1174 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1175 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1176 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1177 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1178 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1179 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1180 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1181 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

Add a secret tag.

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Add a secret tag.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class CreateSecretTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateSecretTagRequest request = new CreateSecretTagRequest();
        CreateSecretTagRequestBody body = new CreateSecretTagRequestBody();
        TagItem tagbody = new TagItem();
        tagbody.withKey("DEV")
            .withValue("DEV1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateSecretTagResponse response = client.createSecretTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Add a secret tag.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
```



```
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecretTagRequest()
        tagbody = TagItem(
            key="DEV",
            value="DEV1"
        )
        request.body = CreateSecretTagRequestBody(
            tag=tagbody
        )
        response = client.create_secret_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Add a secret tag.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
    request := &model.CreateSecretTagRequest{}
    valueTag:= "DEV1"
    tagbody := &model.TagItem{
        Key: "DEV",
        Value: &valueTag,
    }
    request.Body = &model.CreateSecretTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateSecretTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	No Content
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.4.5 Deleting a Secret Tag

Function

This API is used to delete a secret tag.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1182 Path Parameters

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

Request Parameters

Table 4-1183 Request header parameters

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

Response Parameters

Status code: 400

Table 4-1184 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1185 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1186 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1187 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1188 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1189 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1190 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1191 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1192 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1193 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1194 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1195 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1196 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1197 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class DeleteSecretTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
```



```
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteSecretTagRequest request = new DeleteSecretTagRequest();
        try {
            DeleteSecretTagResponse response = client.deleteSecretTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecretTagRequest()
        response = client.delete_secret_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```



```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecretTagRequest{}
    response, err := client.DeleteSecretTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	No Content
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.4.6 Querying Project Tags

Function

This API is used to query all secret tags of a user in a specified project.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/csms/tags

Table 4-1198 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-1199 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1200 Response body parameters

Parameter	Type	Description
tags	Array of **Tag** objects	list of tags, including tag keys and tag values. - key: Tag key. A secret can have a maximum of 20 keys, and each of them is unique and cannot be left blank. A key cannot have duplicate values. The key can contain a maximum of 128 characters. - value: Tag value. Each value can contain a maximum of 255 characters. The relationship between values is AND.

Table 4-1201 Tag

Parameter	Type	Description
key	String	The tag key.
values	Array of strings	Indicates the tag value set. Constraints: A maximum of 10 values can be contained. Tag values in a tag list must be unique. If the value list is left empty, any tag value can be matched. Multiple values in the tag list are in the OR relationship. If the key meets the requirements, a value in the request is matched.

Status code: 400

Table 4-1202 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1203 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1204 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1205 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1206 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1207 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1208 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1209 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1210 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1211 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1212 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1213 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1214 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1215 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ListProjectSecretsTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
```



```
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListProjectSecretsTagsRequest request = new ListProjectSecretsTagsRequest();
        try {
            ListProjectSecretsTagsResponse response = client.listProjectSecretsTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListProjectSecretsTagsRequest()
        response = client.list_project_secrets_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
```



```
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListProjectSecretsTagsRequest{}
    response, err := client.ListProjectSecretsTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.3.5 Event Management

4.3.5.1 Creating an Event

Function

Creates an event that can be configured on one or more secret objects. When an event is enabled and the basic event type contained in the event is triggered on the secret object, the cloud service sends the corresponding event notification to the notification topic specified by the event.

Constraints

The created event notification cannot be used independently and needs to be configured on a specific secret object. The service does not check the validity of the notification object in the event notification. If the notification object does not exist under the user account, the event notification fails when the event is triggered.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/csms/events

Table 4-1216 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-1217 Request header parameters

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

Table 4-1218 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the event notification to be created. Constraints: The value contains 1 to 64 characters and matches the regular expression ^[a-zA-Z0-9_-]{1,64}$.
event_types	Yes	Array of strings	Basic event list of the event notification. The basic event types are as follows: SECRET_VERSION_CREATED: version creation SECRET_VERSION_EXPIRED: The version has expired. SECRET_ROTATED: secret rotation SECRET_DELETED: secret deletion The basic event types contained in the list must be unique.
state	Yes	String	Indicates whether an event takes effect. Basic event types can be triggered only when the event is enabled. Enabled Disabled
notification	Yes	**Notification** object	Notification topic object.

Table 4-1219 Notification

Parameter	Mandatory	Type	Description
target_type	Yes	String	Type of the object to which an event notification is sent.
target_id	Yes	String	ID of the object to which the event notification is sent.
target_name	Yes	String	Name of the object to which the event notification is sent.

Response Parameters

Status code: 200

Table 4-1220 Response body parameters

Parameter	Type	Description
event	**Event** object	Event notification object

Table 4-1221 Event

Parameter	Type	Description
name	String	Event notification name.
event_id	String	Indicates the resource identifier of the event notification.
event_types	Array of strings	Set the basic event type list of the event. Constraint: The array can contain up to 12 elements.
state	String	Event notification status. The options are as follows: **ENABLED DISABLED**
create_time	Long	Time when an event notification is created. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time specified by this parameter.
update_time	Long	Last update time of the event notification. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time.
notification	**Notification** object	Notification topic object.

Table 4-1222 Notification

Parameter	Type	Description
target_type	String	Type of the object to which an event notification is sent.
target_id	String	ID of the object to which the event notification is sent.
target_name	String	Name of the object to which the event notification is sent.

Status code: 400

Table 4-1223 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1224 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1225 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1226 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1227 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1228 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1229 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1230 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1231 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1232 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1233 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1234 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1235 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1236 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

Create an event.

```
{
  "name" : "demo-event",
  "event_types" : [ "SECRET_VERSION_CREATED", "SECRET_VERSION_EXPIRED" ],
  "state" : "ENABLED",
  "notification" : {
    "target_type" : "SMN",
    "target_id" : "urn:smn:cn-north-4:dc3b7c85759141a991da17423c0f2068:test-poc",
    "target_name" : "demo-smn-name"
  }
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "event" : {
    "name" : "event-test",
    "event_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "state" : "ENABLED",
    "event_types" : [ "SECRET_VERSION_CREATED", "SECRET_VERSION_EXPIRED" ],
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "notification" : {
      "target_type" : "SMN",
      "target_id" : "urn:smn:cn-north-4:SecertExpirationTest",
      "target_name" : "SecertExpirationNotificationTest"
    }
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Create an event.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateSecretEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateSecretEventRequest request = new CreateSecretEventRequest();
        CreateSecretEventRequestBody body = new CreateSecretEventRequestBody();
        Notification notificationbody = new Notification();
        notificationbody.withTargetType("SMN")
            .withTargetId("urn:smn:cn-north-4:dc3b7c85759141a991da17423c0f2068:test-poc")
            .withTargetName("demo-smn-name");
        List<String> listbodyEventTypes = new ArrayList<>();
        listbodyEventTypes.add("SECRET_VERSION_CREATED");
```



```
        listbodyEventTypes.add("SECRET_VERSION_EXPIRED");
        body.withNotification(notificationbody);
        body.withState(CreateSecretEventRequestBody.StateEnum.fromValue("ENABLED"));
        body.withEventTypes(listbodyEventTypes);
        body.withName("demo-event");
        request.withBody(body);
        try {
            CreateSecretEventResponse response = client.createSecretEvent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Create an event.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecretEventRequest()
        notificationbody = Notification(
            target_type="SMN",
            target_id="urn:smn:cn-north-4:dc3b7c85759141a991da17423c0f2068:test-poc",
            target_name="demo-smn-name"
        )
        listEventTypesbody = [
            "SECRET_VERSION_CREATED",
            "SECRET_VERSION_EXPIRED"
        ]
        request.body = CreateSecretEventRequestBody(
            notification=notificationbody,
            state="ENABLED",
            event_types=listEventTypesbody,
            name="demo-event"
        )
        response = client.create_secret_event(request)
        print(response)
```



```
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Create an event.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecretEventRequest{}
    notificationbody := &model.Notification{
        TargetType: "SMN",
        TargetId: "urn:smn:cn-north-4:dc3b7c85759141a991da17423c0f2068:test-poc",
        TargetName: "demo-smn-name",
    }
    var listEventTypesbody = []string{
        "SECRET_VERSION_CREATED",
        "SECRET_VERSION_EXPIRED",
    }
    request.Body = &model.CreateSecretEventRequestBody{
        Notification: notificationbody,
        State: model.GetCreateSecretEventRequestBodyStateEnum().ENABLED,
        EventTypes: listEventTypesbody,
        Name: "demo-event",
    }
    response, err := client.CreateSecretEvent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.3.5.2 Querying Events

Function

This API is used to query information about a specified event.

Constraints

The information returned by this API is the metadata information value of the secret event notification.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/csms/events/{event_name}

Table 4-1237 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
event_name	Yes	String	Name of an event notification.

Request Parameters

Table 4-1238 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1239 Response body parameters

Parameter	Type	Description
event	**Event** object	Event notification object

Table 4-1240 Event

Parameter	Type	Description
name	String	Event notification name.
event_id	String	Indicates the resource identifier of the event notification.
event_types	Array of strings	Set the basic event type list of the event. Constraint: The array can contain up to 12 elements.
state	String	Event notification status. The options are as follows: **ENABLED DISABLED**
create_time	Long	Time when an event notification is created. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time specified by this parameter.
update_time	Long	Last update time of the event notification. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time.
notification	**Notification** object	Notification topic object.

Table 4-1241 Notification

Parameter	Type	Description
target_type	String	Type of the object to which an event notification is sent.
target_id	String	ID of the object to which the event notification is sent.
target_name	String	Name of the object to which the event notification is sent.

Status code: 400

Table 4-1242 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1243 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1244 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1245 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1246 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1247 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1248 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1249 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1250 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1251 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1252 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1253 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1254 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1255 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "event" : {
    "name" : "event-test",
    "event_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "state" : "ENABLED",
    "event_types" : [ "SECRET_VERSION_EXPIRED" ],
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "notification" : {
      "target_type" : "SMN",
      "target_id" : "urn:smn:cn-north-4:SecertExpirationTest",
      "target_name" : "SecertExpirationNotificationTest"
    }
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ShowSecretEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSecretEventRequest request = new ShowSecretEventRequest();
        try {
            ShowSecretEventResponse response = client.showSecretEvent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
```



```
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSecretEventRequest()
        response = client.show_secret_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```



```
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSecretEventRequest{}
    response, err := client.ShowSecretEvent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.3.5.3 Querying the Event List

Function

This API is used to query all events created by the current user in the project.

Constraints

The information returned by this API is the metadata information of the event notification, excluding the secret value.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/csms/events

Table 4-1256 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 4-1257 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	String	Specifies the number of records on each page. Default value: 50
marker	No	String	Specifies the resource ID of pagination query. If the parameter is left blank, only resources on the first page are queried.

Request Parameters

Table 4-1258 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1259 Response body parameters

Parameter	Type	Description
events	Array of **Event** objects	Event details list
page_info	**PageInfo** object	Pagination information.

Table 4-1260 Event

Parameter	Type	Description
name	String	Event notification name.
event_id	String	Indicates the resource identifier of the event notification.
event_types	Array of strings	Set the basic event type list of the event. Constraint: The array can contain up to 12 elements.
state	String	Event notification status. The options are as follows: **ENABLED DISABLED**
create_time	Long	Time when an event notification is created. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time specified by this parameter.
update_time	Long	Last update time of the event notification. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time.
notification	**Notification** object	Notification topic object.

Table 4-1261 Notification

Parameter	Type	Description
target_type	String	Type of the object to which an event notification is sent.
target_id	String	ID of the object to which the event notification is sent.
target_name	String	Name of the object to which the event notification is sent.

Table 4-1262 PageInfo

Parameter	Type	Description
next_marker	String	Query address of the next page (name of the secret at the end of the current page, name of the secret at the start of the next page).
previous_marker	String	Name of the start secret on the current page and name of the end secret on the previous page.
current_count	Integer	Number of records returned on this page.

Status code: 400

Table 4-1263 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1264 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1265 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1266 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1267 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1268 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1269 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1270 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1271 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1272 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1273 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1274 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1275 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1276 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.


```
{
  "events" : [ {
    "state" : "ENABLED",
    "name" : "TestEvent1",
    "event_id" : "120eeafd-eca5-4098-8733-c1b369f3a450",
    "event_types" : [ "SECRET_VERSION_CREATED", "SECRET_VERSION_EXPIRED",
"SECRET_VERSION_NEAR_EXPIRY", "SECRET_DELETED" ],
    "create_time" : 1669042498000,
    "update_time" : 1669042498000,
    "notification" : {
      "target_type" : "SMN",
      "target_id" : "urn:smn:cn-north-4:dc3b7c85759141a991da17423c0f2068:smn-target-name",
      "target_name" : "smn-target-name"
    }
  }, {
    "state" : "ENABLED",
    "name" : "TestEvent2",
    "event_id" : "1d251c99-37d2-4396-86ce-f000d3aa9850",
    "event_types" : [ "SECRET_VERSION_CREATED", "SECRET_VERSION_EXPIRED",
"SECRET_VERSION_NEAR_EXPIRY", "SECRET_DELETED" ],
    "create_time" : 1669041491000,
    "update_time" : 1669041491000,
    "notification" : {
      "target_type" : "SMN",
      "target_id" : "urn:smn:cn-north-4:dc3b7c85759141a991da17423c0f2068:smn-target-name",
      "target_name" : "smn-target-name"
    }
  } ],
  "page_info" : {
    "next_marker" : "TestEvent2",
    "previous_marker" : "TestEvent3",
    "current_count" : 2
  }
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ListSecretEventsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```



```
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSecretEventsRequest request = new ListSecretEventsRequest();
        request.withLimit("<limit>");
        request.withMarker("<marker>");
        try {
            ListSecretEventsResponse response = client.listSecretEvents(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecretEventsRequest()
        request.limit = "<limit>"
        request.marker = "<marker>"
        response = client.list_secret_events(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecretEventsRequest{}
    limitRequest:= "<limit>"
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListSecretEvents(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.

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

Error Codes

See **Error Codes**.

4.3.5.4 Update an Event

Function

This API is used to update the metadata of a specified event. Metadata that can be updated includes the event enabling status, basic type list, and notification topic.

Constraints

This API can be used to modify only the metadata of secret event notifications.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v1/{project_id}/csms/events/{event_name}

Table 4-1277 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
event_name	Yes	String	Event notification name

Request Parameters

Table 4-1278 Request header parameters

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

Table 4-1279 Request body parameters

Parameter	Mandatory	Type	Description
state	No	String	Event notification status. The options are as follows: **ENABLED DISABLED**
event_types	No	Array of strings	Basic event list of the event notification. The basic event types are as follows: SECRET_VERSION_CREATED: version creation SECRET_VERSION_EXPIRED: The version has expired. SECRET_ROTATED: secret rotation SECRET_DELETED: secret deletion The basic event types contained in the list must be unique.
notification	No	**Notification** object	Notification topic object.

Table 4-1280 Notification

Parameter	Mandatory	Type	Description
target_type	Yes	String	Type of the object to which an event notification is sent.
target_id	Yes	String	ID of the object to which the event notification is sent.
target_name	Yes	String	Name of the object to which the event notification is sent.

Response Parameters

Status code: 200

Table 4-1281 Response body parameters

Parameter	Type	Description
event	**Event** object	Event notification object

Table 4-1282 Event

Parameter	Type	Description
name	String	Event notification name.
event_id	String	Indicates the resource identifier of the event notification.
event_types	Array of strings	Set the basic event type list of the event. Constraint: The array can contain up to 12 elements.
state	String	Event notification status. The options are as follows: **ENABLED DISABLED**
create_time	Long	Time when an event notification is created. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time specified by this parameter.
update_time	Long	Last update time of the event notification. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time.
notification	**Notification** object	Notification topic object.

Table 4-1283 Notification

Parameter	Type	Description
target_type	String	Type of the object to which an event notification is sent.
target_id	String	ID of the object to which the event notification is sent.
target_name	String	Name of the object to which the event notification is sent.

Status code: 400

Table 4-1284 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1285 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1286 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1287 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1288 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1289 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1290 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1291 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1292 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1293 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1294 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1295 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1296 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1297 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

This API is used to update a specified event.

```
{
  "notification" : {
    "target_type" : "SMN",
    "target_id" : "demo-smn-id",
    "target_name" : "demo-smn-name"
  }
}
```

Example Responses

Status code: 200

Request succeeded.

```
{
  "event" : {
    "name" : "event-test",
    "event_id" : "bb6a3d22-dc93-47ac-b5bd-88df7ad35f1e",
    "state" : "ENABLED",
    "event_types" : [ "SECRET_VERSION_EXPIRED" ],
    "create_time" : 1581507580000,
    "update_time" : 1581507580000,
    "notification" : {
      "target_type" : "SMN",
      "target_id" : "urn:smn:cn-north-4:SecertExpirationTest",
      "target_name" : "SecertExpirationNotificationTest"
    }
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

This API is used to update a specified event.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class UpdateSecretEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateSecretEventRequest request = new UpdateSecretEventRequest();
        UpdateSecretEventRequestBody body = new UpdateSecretEventRequestBody();
        Notification notificationbody = new Notification();
        notificationbody.withTargetType("SMN")
            .withTargetId("demo-smn-id")
            .withTargetName("demo-smn-name");
        body.withNotification(notificationbody);
        request.withBody(body);
        try {
            UpdateSecretEventResponse response = client.updateSecretEvent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```



```
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

This API is used to update a specified event.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSecretEventRequest()
        notificationbody = Notification(
            target_type="SMN",
            target_id="demo-smn-id",
            target_name="demo-smn-name"
        )
        request.body = UpdateSecretEventRequestBody(
            notification=notificationbody
        )
        response = client.update_secret_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

This API is used to update a specified event.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
```



```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSecretEventRequest{}
    notificationbody := &model.Notification{
        TargetType: "SMN",
        TargetId: "demo-smn-id",
        TargetName: "demo-smn-name",
    }
    request.Body = &model.UpdateSecretEventRequestBody{
        Notification: notificationbody,
    }
    response, err := client.UpdateSecretEvent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.

Status Code	Description
502	Failed to complete the request because the server receives an invalid response from an upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

4.3.5.5 Deleting an Event Immediately

Function

The specified event is deleted immediately and cannot be restored. If an event is referenced by a secret, the event cannot be deleted. Disassociate the event from the secret first.

Constraints

Secrets deleted via this API cannot be restored.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v1/{project_id}/csms/events/{event_name}

Table 4-1298 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
event_name	Yes	String	Name of an event notification.

Request Parameters

Table 4-1299 Request header parameters

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

Response Parameters

Status code: 400

Table 4-1300 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1301 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1302 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1303 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.

Parameter	Type	Description
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1304 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1305 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1306 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1307 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1308 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1309 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1310 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1311 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1312 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1313 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class DeleteSecretEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteSecretEventRequest request = new DeleteSecretEventRequest();
        try {
            DeleteSecretEventResponse response = client.deleteSecretEvent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
```



```
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSecretEventRequest()
        response = client.delete_secret_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
```



```
    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSecretEventRequest{}
    response, err := client.DeleteSecretEvent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.3.5.6 Querying Triggered Event Notification Records

Function

Query all event notification records that have been triggered in the last three months.

Constraints

Only event notification records generated within three months are stored.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/csms/notification-records

Table 4-1314 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 4-1315 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1316 Response body parameters

Parameter	Type	Description
records	Array of **Record** objects	Record object
page_info	**PageInfo** object	Pagination information.

Table 4-1317 Record

Parameter	Type	Description
event_name	String	Secret name.

Parameter	Type	Description
trigger_event_type	String	Secret type. The options are as follows: COMMON: common secret; RDS: RDS secret
create_time	Long	Time when an event notification record is created. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time specified by this parameter.
secret_name	String	Secret name
secret_type	String	Secret type. The options are as follows: COMMON: common secret (default) stores sensitive information in the application system. RDS: RDS secrets. RDS secrets are used to store RDS account information.
notification_target_name	String	Name of the object to which the event notification is sent.
notification_target_id	String	ID of the object to which the event notification is sent.
notification_content	String	Description of a secret.
notification_status	String	Secret type. Value: SUCCESS: The event notification is successful. FAIL: The event notification fails. INVALID: The event notification topic is invalid or incorrect. As a result, the notification cannot be triggered.

Table 4-1318 PageInfo

Parameter	Type	Description
next_marker	String	Query address of the next page (name of the secret at the end of the current page, name of the secret at the start of the next page).
previous_marker	String	Name of the start secret on the current page and name of the end secret on the previous page.
current_count	Integer	Number of records returned on this page.

Status code: 400

Table 4-1319 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1320 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 401

Table 4-1321 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1322 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 403

Table 4-1323 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1324 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 404

Table 4-1325 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1326 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 500

Table 4-1327 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1328 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 502

Table 4-1329 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1330 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Status code: 504

Table 4-1331 Response body parameters

Parameter	Type	Description
error	**ErrorDetail** object	Error message.

Table 4-1332 ErrorDetail

Parameter	Type	Description
error_code	String	Error code returned for an error request.
error_msg	String	Error information returned for an error request.

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

```
{
  "records" : [ {
    "event_name" : "demo-event",
    "trigger_event_type" : "SECRET_VERSION_EXPIRED",
    "create_time" : 1581507580000,
    "secret_name" : "demo-secret",
    "secret_type" : "COMMON",
    "notification_target_name" : "SecertExpirationNotificationTest",
```



```
    "notification_target_id" : "urn:smn:cn-north-4:SecertExpirationTest",
    "notification_content" : "{\"eventName\":\"TestEvent20221112\",\"eventType
\":\"SECRET_VERSION_EXPIRED\",\"eventTime\":\"2023-04-14T20:25:10.126Z\",\"data\":{\"secretId
\":\"fde3d6ba-cb31-40b0-b6c4-78757050f8c8\",\"secretName\":\"Secret20230325\",\"createTime
\":\"2023-03-22T20:43:04.000Z\",\"updateTime\":\"2023-04-14T20:18:29.000Z\",\"versionId
\":\"v18\",\"versionExpireTime\":\"2023-03-29T17:07:23.000Z\"}}",
    "notification_status" : "SUCCESS"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class ListNotificationRecordsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListNotificationRecordsRequest request = new ListNotificationRecordsRequest();
        try {
            ListNotificationRecordsResponse response = client.listNotificationRecords(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListNotificationRecordsRequest()
        response = client.list_notification_records(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNotificationRecordsRequest{}
    response, err := client.ListNotificationRecords(request)
```



```
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

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

Error Codes

See **Error Codes**.

4.3.6 Secret version management

4.3.6.1 Creating a Secret Version

Function

Creates a new secret version in the specified secret to encrypt and keep the new secret value. By default, The latest secret version in **SYSCURRENT** state. The previous version is in the **SYSPREVIOUS** state. You can overwrite the default behavior by specifying the **VersionStage** parameter.

Constraints

- On the CSMS console, only the **secret_string** field can be configured. To add binary secrets to the secret_binary field, use an SDK or API. * A secret can have up to 20 versions in CSMS. * You can only add versions to enabled

secrets. * Secret versions are numbered v1, v2, v3, and so on based on their creation time.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1333 Path Parameters

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

Request Parameters

Table 4-1334 Request header parameters

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

Table 4-1335 Request body parameters

Parameter	Mandatory	Type	Description
secret_binary	No	String	Value of a new secret. The value is encrypted and stored in the initial version of the secret. Type: Base64-encoded binary data object Constraint: Either **secret_binary** or **secret_string** must be configured. The maximum size is 32 KB.

Parameter	Mandatory	Type	Description
secret_string	No	String	Value of a new secret. The value is encrypted and stored in the initial version of the secret. Constraint: Either **secret_binary** or **secret_string** must be configured. The maximum size is 32 KB.
version_stages	No	Array of strings	Version status added to a secret version when the version is stored. If you do not specify this parameter, secret manager defaults to SYSCURRENT for the new version. Constraint: The array can contain up to 12 elements. The stage length can be up to 64 bytes.
expire_time	No	Long	Expiration time of the secret version. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the current time. Value based on which the validity period is determined when the secret subscribes to the version expiration event type. This parameter is left blank by default.

Response Parameters

Status code: 200

Table 4-1336 Response body parameters

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

Table 4-1337 VersionMetadata

Parameter	Type	Description
id	String	Secret version ID, which is unique under a secret object.
create_time	Long	Time when a secret version was created. The timestamp indicates the total seconds since January 1, 1970.
expire_time	Long	Expiration time of the secret version. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time. When the secret subscribes to the version expiration event type, the validity period is determined based on this parameter. This parameter is left blank by default.
kms_key_id	String	ID of the KMS CMK used to encrypt a secret version value.
secret_name	String	Secret name
version_stages	Array of strings	Secret version status list. Every version status is unique under a secret. If you add a status tag in use to a new version, the tag will be automatically removed from the old version. If **version_stage** is not specified, the temporary tag **SYSCURRENT** will be added to this version.

Status code: 400

Table 4-1338 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1339 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1340 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1341 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1342 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1343 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 504

Table 4-1344 Response body parameters

Parameter	Type	Description
error_code	String	Error codes

Parameter	Type	Description
error_msg	String	Error description

Example Requests

Create a secret version. The secret value is secret_string.

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

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Create a secret version. The secret value is secret_string.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;


public class CreateSecretVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
```



```
            .withSk(sk);

    CsmsClient client = CsmsClient.newBuilder()
            .withCredential(auth)
            .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
            .build();
    CreateSecretVersionRequest request = new CreateSecretVersionRequest();
    CreateSecretVersionRequestBody body = new CreateSecretVersionRequestBody();
    body.withSecretString("secret_string");
    request.withBody(body);
    try {
        CreateSecretVersionResponse response = client.createSecretVersion(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
  }
}
```

Python

Create a secret version. The secret value is secret_string.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSecretVersionRequest()
        request.body = CreateSecretVersionRequestBody(
            secret_string="secret_string"
        )
        response = client.create_secret_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

Create a secret version. The secret value is secret_string.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSecretVersionRequest{}
    secretStringCreateSecretVersionRequestBody:= "secret_string"
    request.Body = &model.CreateSecretVersionRequestBody{
        SecretString: &secretStringCreateSecretVersionRequestBody,
    }
    response, err := client.CreateSecretVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.
401	A username and password are required.
403	Authentication failed.

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

Error Codes

See **Error Codes**.

4.3.6.2 Querying the Secret Version List

Function

This API is used to query the version list of a specified secret.

Constraints

The information returned by this API is the metadata of secret versions, which does not contain secret values.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-1345 Path Parameters

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

Table 4-1346 Query Parameters

Parameter	Mandatory	Type	Description
marker	No	String	Pagination parameter. The value is the version number of the last record on the previous page.
limit	No	Integer	Specifies the number of items displayed per page. The default value is 50.

Request Parameters

Table 4-1347 Request header parameters

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

Response Parameters

Status code: 200

Table 4-1348 Response body parameters

Parameter	Type	Description
version_metadatas	Array of **VersionMetadata** objects	**version_metadata** object.
page_info	**PageInfo** object	Pagination information.

Table 4-1349 VersionMetadata

Parameter	Type	Description
id	String	Secret version ID, which is unique under a secret object.

Parameter	Type	Description
create_time	Long	Time when a secret version was created. The timestamp indicates the total seconds since January 1, 1970.
expire_time	Long	Expiration time of the secret version. The value is a timestamp, that is, the total number of seconds from January 1, 1970 to the time. When the secret subscribes to the version expiration event type, the validity period is determined based on this parameter. This parameter is left blank by default.
kms_key_id	String	ID of the KMS CMK used to encrypt a secret version value.
secret_name	String	Secret name
version_stages	Array of strings	Secret version status list. Every version status is unique under a secret. If you add a status tag in use to a new version, the tag will be automatically removed from the old version. If **version_stage** is not specified, the temporary tag **SYSCURRENT** will be added to this version.

Table 4-1350 PageInfo

Parameter	Type	Description
next_marker	String	Query address of the next page (name of the secret at the end of the current page, name of the secret at the start of the next page).
previous_marker	String	Name of the start secret on the current page and name of the end secret on the previous page.
current_count	Integer	Number of records returned on this page.

Status code: 400

Table 4-1351 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 401

Table 4-1352 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 403

Table 4-1353 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 404

Table 4-1354 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 500

Table 4-1355 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Status code: 502

Table 4-1356 Response body parameters

Parameter	Type	Description
error_code	String	Error codes

Parameter	Type	Description
error_msg	String	Error description

Status code: 504

Table 4-1357 Response body parameters

Parameter	Type	Description
error_code	String	Error codes
error_msg	String	Error description

Example Requests

None

Example Responses

Status code: 200

Request succeeded.

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

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.csms.v1.region.CsmsRegion;
import com.huaweicloud.sdk.csms.v1.*;
import com.huaweicloud.sdk.csms.v1.model.*;
```



```
public class ListSecretVersionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
                .withAk(ak)
                .withSk(sk);

        CsmsClient client = CsmsClient.newBuilder()
                .withCredential(auth)
                .withRegion(CsmsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSecretVersionsRequest request = new ListSecretVersionsRequest();
        request.withMarker("<marker>");
        request.withLimit(<limit>);
        try {
            ListSecretVersionsResponse response = client.listSecretVersions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcsms.v1.region.csms_region import CsmsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcsms.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = CsmsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CsmsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSecretVersionsRequest()
        request.marker = "<marker>"
        request.limit = <limit>
        response = client.list_secret_versions(request)
```



```
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    csms "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/csms/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := csms.NewCsmsClient(
        csms.CsmsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSecretVersionsRequest{}
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListSecretVersions(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Request succeeded.
400	Invalid request parameter.

Status Code	Description
401	A username and password are required.
403	Authentication failed.
404	The requested resource does not exist or is not found.
500	Internal service error.
502	The request failed to be fulfilled because the server received an invalid response from the upstream server.
504	Gateway timed out.

Error Codes

See **Error Codes**.

5 Historical APIs

5.1 Managing SSH Key Pairs (V2.1)

5.1.1 Querying the List of SSH Key Pairs (V2.1)

Function

This API is used to query the list of SSH key pairs.

URI

- URI format
  GET /v2.1/{project_id}/os-keypairs
- Parameter description

  Table 5-1 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

None

Responses

Table 5-2 Response parameters

Parameter	Mandatory	Type	Description
keypairs	Yes	Array of objects	SSH key pair information list. For details, see **Table 5-3**.

Table 5-3 keypairs field description

Parameter	Mandatory	Type	Description
keypair	Yes	Object	SSH key pair information, see **Table 5-4**.

Table 5-4 keypair field description

Parameter	Mandatory	Type	Description
fingerprint	Yes	String	Fingerprint information about an SSH key pair
name	Yes	String	Name of an SSH key pair
public_key	Yes	String	Public key information about an SSH key pair
is_key_protection	Yes	Boolean	Whether the private key of the SSH key pair is managed and protected.

Examples

- Example request

  None

- Example response

```
{
 "keypairs": [{
   "keypair": {
    "fingerprint": "15:b0:f8:b3:f9:48:63:71:cf:7b:5b:38:6d:44:2d:4a",
    "name": "keypair-601a2305-4f25-41ed-89c6-2a966fc8027a",
    "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQC+Eo/
RZRngaGTkFs7I62ZjsIlO79KklKbMXi8F+KITD4bVQHHn+kV
+4gRgkgCRbdoDqoGfpaDFs877DYX9n4z6FrAIZ4PES8TNKhatifpn9NdQYWA+IkU8CuvlEKGuFpKRi/
k7JLos/gHi2hy7QUwgtRvcefvD/vgQZOVw/mGR9Q== Generated-by-Nova\n",
    "is_key_protection": true
   }
 }
```

```
 ]
}
```

or

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

Status Codes

For details, see **Status Codes**.

5.1.2 Querying Details About an SSH Key Pair (V2.1)

Function

This API is used to query details about an SSH key pair.

URI

- URI format

  GET /v2.1/{project_id}/os-keypairs/{keypair_name}
- Parameter description

  Table 5-5 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Name of an SSH key pair

Requests

None

Responses

Table 5-6 Response parameters

Parameter	Mandatory	Type	Description
keypair	Yes	Object	SSH key pair information. For details, see **Table 5-7**.

Table 5-7 keypair field description

Parameter	Mandatory	Type	Description
public_key	Yes	String	Public key information about an SSH key pair
name	Yes	String	Name of an SSH key pair
fingerprint	Yes	String	Fingerprint information about an SSH key pair
created_at	Yes	String	Time when an SSH key pair is created The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
deleted	Yes	Boolean	Tag that indicates an SSH key pair is deleted
deleted_at	Yes	String	Time when an SSH key pair is deleted The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
id	Yes	String	ID of an SSH key pair
updated_at	Yes	String	Time when an SSH key pair is updated The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
user_id	Yes	String	User to whom an SSH key pair belongs
is_key_protection	Yes	Boolean	Whether the private key of the SSH key pair is managed and protected.
description	Yes	String	Description of an SSH key pair

Examples

- Example request

  None

- Example response

```
{
    "keypair": {
        "created_at": "2014-05-07T12:06:13.681238",
        "deleted": false,
        "deleted_at": null,
        "fingerprint": "9d:00:f4:d7:26:6e:52:06:4c:c1:d3:1d:fd:06:66:01",
        "id": 1,
        "name": "keypair-3582d8b7-e588-4aad-b7f7-f4e76f0e4314",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDYJrTVpcMwFqQy/
oMvtUSRofZdSRHEwrsX8AYkRvn2ZnCXM+b6+GZ2NQuuWj+ocznlnwiGFQDsL/yeE+/
kurqcPJFKKp60mToXIMyzioFxW88fJtwEWawHKAclbHWpR1t4fQ4DS+/sIbX/
Yd9btlVQ2tpQjodGDbM9Tr9/+/3i6rcR+EoLqmbgCgAiGiVV6VbM2Zx79yUwd
+GnQejHX8BlYZoOjCnt3NREsITcmWE9FVFy6TnLmahs3FkEO/
```

```
QGgWGkaohAJlsgaVvSWGgDn2AujKYwyDokK3dXyeX3m2Vmc3ejiqPa/C4nRrCOlko5nSgV/
9IXRx1ERImsqZnE9usB Generated-by-Nova\n",
        "updated_at": null,
        "user_id": "fake",
        "is_key_protection": true,
        "description": "keypair test"
    }
}
```

or

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

Status Codes

For details, see **Status Codes**.

5.1.3 Creating and Importing an SSH Key Pair (V2.1)

Function

This API is used to create and import an SSH key pair, and allows you to manage the private key in the cloud.

URI

- URI format

  POST /v2.1/{project_id}/os-keypairs
- Parameter description

  Table 5-8 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

NOTE

When creating an SSH key pair, you only need to set the **name**. When importing an SSH key pair, you must set the **public_key** parameter.

Table 5-9 Request parameters

Parameter	Mandatory	Type	Description
keypair	Yes	Object	Information about the created or imported SSH key pair. For details, see **Table 5-10**.

Table 5-10 keypair field description

Parameter	Mandatory	Type	Description
public_key	No	String	Character string of a public key to be imported
name	Yes	String	Name of an SSH key pair A new key pair cannot use the same name as an existing one. The value contains a maximum of 64 characters, including only letters, digits, underscores(_), and hyphens (-).
user_id	No	String	ID of the user to whom an SSH key pair belongs
key_protection	No	Object	Management and protection of the private key of an SSH key pair. For details, see **Table 5-11**.

Table 5-11 key_protection field description

Parameter	Mandatory	Type	Description
private_key	No	String	Character string of a private key to be imported
encryption	Yes	Object	Method to encrypt and store private keys. For details, see **Table 5-12**.

Table 5-12 encryption field description

Parameter	Mandatory	Type	Description
type	Yes	String	Value options: **kms** or **default** • **default** refers to the default encryption method. It is applicable to sites where the KMS service is unavailable. • **kms** indicates that the KMS service is leveraged for the encryption. If the KMS service is unavailable, set this parameter to **default**.

Parameter	Mandatory	Type	Description
kms_key_name	No	String	Name of a KMS key. If **type** is set to **kms**, enter the name of the key obtained from KMS.

Responses

Table 5-13 Response parameters

Parameter	Mandatory	Type	Description
keypair	Yes	Object	SSH key pair information. For details, see **Table 5-14**.

Table 5-14 keypair field description

Parameter	Mandatory	Type	Description
fingerprint	Yes	String	Fingerprint information about an SSH key pair
name	Yes	String	Name of an SSH key pair
public_key	Yes	String	Public key information about an SSH key pair
private_key	Yes	String	Private key information about an SSH key pair. • The information about the private key is contained in the response for creating an SSH key pair. • The information about the private key is not contained in the response for importing an SSH key pair.
user_id	Yes	String	User to whom an SSH key pair belongs

Examples

- Creating an SSH key pair
  - Request example for creating an SSH key pair

```
{
    "keypair": {
        "name": "demo1"
```



```
    }
}
```

  - Response example for creating an SSH key pair

```
{
    "keypair": {
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCXKzohKbFOqubYNunFNsrEYlk9NEIJIFvbmTe/
LTeMzFIPKM53Zu2sYr/uuNcziPkWpFchXdkD+O/
Bf2ZzKaR1DYPMWss9TkaqU4RQ7CIBW7ChJF1Qzc1JPRBmKe6e8qs1QBBoS1QvXgSjbuf2Fb1yncSb
phmQV8+5KA8xkxz4XdM1/gSAZZ14rJrMjgp7jCdgxWiHNcDuKxaPt+0eO8rEG/gxR7J0b9Uk53ao/
xjLoKXYdLLiYUaha0fHdW3t6Lw1NdzUMmKnlLqN9O37Tbg7vM0nN4UJt0XXvM45KfnJiMx0HUKXd
WkUj9cE8VBDPw/gBbQzSpJHgQFG7mNDZubN Generated-by-Nova\n",
        "private_key": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAlys6ISmxTqrm2DbpxTbKxGJZPTRCCSBb25k3vy03jMxSDyjO\nd2btrGK/
7rjXM4j5FqRXIV3ZA/jvwX9mcymkdQ2DzFrLPU5GqlOEUOwiAVuwoSRd
\nUM3NST0QZinunvKrNUAQaEtUL14Eo27n9hW9cp3Em6YZkFfPuSgPMZMc+F3TNf4E
\ngGWdeKyazI4Ke4wnYMVohzXA7isWj7ftHjvKxBv4MUeydG/VJOd2qP8Yy6Cl2HSy
\n4mFGoWtHx3Vt7ei8NTXc1DJip5S6jfTt+024O7zNJzeFCbdF17zOOSn5yYjMdB1C\nl3VpFI/
XBPFQQz8P4AW0M0qSR4EBRu5jQ2bmzQIDAQABAoIBAFwm1s3Gi7blCec+
\nOm2lhCAJUBs2UxcbKcQJd+lL2DQ1jcR9/LbfAP34SMZu2Ykp86Zw0kid3CBGzWko
\nj7yeYwmUDocxxfl+USedt+hYujYXvenNsDEE9CK0Xd3ZrAQrLGFOx3G8kfo6FxvG\nDRN/
lzhaorK7o5PRY+icWf6/joZ8Q96scHmm0ob5rtBkUYcek+ckf3mLVIpzzdKA
\ndkSi57M78zwDA89MpVABEoO1DPVxEqrrMQZy5UnAmeGHh16mPS4qMCokPVz36pSG
\nIWSqHnVKzsbxvw5Da9y69NmpSi2E1wqDaU9IzwnLyQpHnE1nXsWmxNqKTHlDBbnb
\nXPGFdcECgYEAxf4IMqYBeBiq
+7RVwcTcT4gpApJmywigwMFaaX35E3O53ja8hk1/\n5OCRnvK7yrt9wnWY8DIh8GPJptKzuTb/l/
14L4kE1MYm7Gpho5SwXV5BqtjgjfZm\nQVNpWruXEugXALcfbHiH+peO
+3AmwgIqgkOLPLxY1Duw6/miDB8bOfECgYEAw3VO\nl9edXExJvJvSzAopSNmw
+ExpUZTgS3L2Pyn21QhfwNHyxPH8fNNKv0/x9ZzBn25U
\nUxXTPPbLFV3cq7kfuYFW0OZkh8QjCPDkIE116E2QvacxqkBuW774xr5msfWdxpcp
\nwccgWKci1vEHLtqj3RTNMFKcXtj4QrCES4ZsPp0CgYEAhzYYux4LWszd188r0Yxz
\nOA0wlIUOlhFqVri02d4hv1sEz/Bphv5eHRwP1pDGFok8NRTCQSa7bsN27uptuksl\n
+e8rqKrWFcjMxB9SVrgwSVMZXWeG0uw2oN4p0MDTRyIgs2hbmWQm2Mev6Z6JmWyJ
\nQzSFXhDySN4K0NxGAiksGsECgYAJvZTXGFWtPdgG4DUXGgKIsOCS3yv4dtTerbH7\n39l/
MFWlRFE242BT0CvPcOp79P3pNhRZt6K5TQs921md7THZisqKypCD+5BLZ8XW\nnkWb
+BGYgfaFp4RYaiH3tZFkmPrt5KaeE5BXGq0vzP8wpJC5+cln+RX13BYzLJzL
\nLr3COQKBgDJjY0qop3aSG+1tSkMJtOFdb2+qXiqaE+Wxv03SOkPx8LBz+taNz+DE
\nR7THDNXSrS85WVc3ozE8hULBX12iIWpr5kb5lRaw3ZgX6wBnSSpdQu49v3nFoGJk\nUWY95pQ/
BWQaX0q4KxzRVxup+7gwT5sKFXU+ktFtsGMYoDoSzbZS\n-----END RSA PRIVATE KEY-----\n",
        "user_id": "e4f380899b1248918f3d37098dc63746",
        "name": "demo123",
        "fingerprint": "49:ef:73:2b:9b:7f:2e:0c:58:d3:e3:42:8e:28:04:3b"
    }
}
```

    or

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

- Creating an SSH key pair and managing the private key in the cloud
  - Request example for creating an SSH key pair and managing the private key in the cloud

```
{
    "keypair": {
        "name": "demo2",
        "key_protection": {
            "encryption": {
                "type": "kms",
                "kms_key_name": "demo"
            }
        }
    }
}
```

  - Response example for creating an SSH key pair and managing the private key in the cloud


```
{
   "keypair": {
      "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCXKzohKbFOqubYNunFNsrEYlk9NEIJIFvbmTe/
LTeMzFIPKM53Zu2sYr/uuNcziPkWpFchXdkD+O/
Bf2ZzKaR1DYPMWss9TkaqU4RQ7CIBW7ChJF1Qzc1JPRBmKe6e8qs1QBBoS1QvXgSjbuf2Fb1yncSb
phmQV8+5KA8xkxz4XdM1/gSAZZ14rJrMjgp7jCdgxWiHNcDuKxaPt+0eO8rEG/gxR7J0b9Uk53ao/
xjLoKXYdLLiYUaha0fHdW3t6Lw1NdzUMmKnlLqN9O37Tbg7vM0nN4UJt0XXvM45KfnJiMx0HUKXd
WkUj9cE8VBDPw/gBbQzSpJHgQFG7mNDZubN Generated-by-Nova\n",
      "private_key": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAlys6ISmxTqrm2DbpxTbKxGJZPTRCCSBb25k3vy03jMxSDyjO\nd2btrGK/
7rjXM4j5FqRXIV3ZA/jvwX9mcymkdQ2DzFrLPU5GqlOEUOwiAVuwoSRd
\nUM3NST0QZinunvKrNUAQaEtUL14Eo27n9hW9cp3Em6YZkFfPuSgPMZMc+F3TNf4E
\ngGWdeKyazI4Ke4wnYMVohzXA7isWj7ftHjvKxBv4MUeydG/VJOd2qP8Yy6Cl2HSy
\n4mFGoWtHx3Vt7ei8NTXc1DJip5S6jfTt+024O7zNJzeFCbdF17zOOSn5yYjMdB1C\nl3VpFI/
XBPFQQz8P4AW0M0qSR4EBRu5jQ2bmzQIDAQABAoIBAFwm1s3Gi7bICec+
\nOm2lhCAJUBs2UxcbKcQJd+lL2DQ1jcR9/LbfAP34SMZu2Ykp86Zw0kid3CBGzWko
\nj7yeYwmUDocxxfl+USedt+hYujYXvenNsDEE9CK0Xd3ZrAQrLGFOx3G8kfo6FxvG\nDRN/
lzhaorK7o5PRY+icWf6/joZ8Q96scHmm0ob5rtBkUYcek+ckf3mLVIpzzdKA
\ndkSi57M78zwDA89MpVABEoO1DPVxEqrrMQZy5UnAmeGHh16mPS4qMCokPVz36pSG
\nIWSqHnVKzsbxvw5Da9y69NmpSi2E1wqDaU9IzwnLyQpHnE1nXsWmxNqKTHlDBbnb
\nXPGFdcECgYEAxf4IMqYBeBiq
+7RVwcTcT4gpApJmywigwMFaaX35E3O53ja8hk1/\n5OCRnvK7yrt9wnWY8DIh8GPJptKzuTb/l/
14L4kE1MYm7Gpho5SwXV5BqtjgjfZm\nQVNpWruXEugXALcfbHiH+peO
+3AmwgIqgkOLPLxY1Duw6/miDB8bOfECgYEAw3VO\nl9edXExJvJvSzAopSNmw
+ExpUZTgS3L2Pyn21QhfwNHyxPH8fNNKv0/x9ZzBn25U
\nUxXTPPbLFV3cq7kfuYFW0OZkh8QjCPDkIE116E2QvacxqkBuW774xr5msfWdxpcp
\nwccgWKci1vEHLtqj3RTNMFKcXtj4QrCES4ZsPp0CgYEAhzYYux4LWszd188r0Yxz
\nOA0wlIUOlhFqVri02d4hv1sEz/Bphv5eHRwP1pDGFok8NRTCQSa7bsN27uptuksl\n
+e8rqKrWFcjMxB9SVrgwSVMZXWeG0uw2oN4p0MDTRyIgs2hbmWQm2Mev6Z6JmWyJ
\nQzSFXhDySN4K0NxGAiksGsECgYAJvZTXGFWtPdgG4DUXGgKIsOCS3yv4dtTerbH7\n39l/
MFWlRFE242BT0CvPcOp79P3pNhRZt6K5TQs921md7THZisqKypCD+5BLZ8XW\nnkWb
+BGYgfaFp4RYaiH3tZFkmPrt5KaeE5BXGq0vzP8wpJC5+cln+RX13BYzLJzL
\nLr3COQKBgDJjY0qop3aSG+1tSkMJtOFdb2+qXiqaE+Wxv03SOkPx8LBz+taNz+DE
\nR7THDNXSrS85WVc3ozE8hULBX12iIWpr5kb5lRaw3ZgX6wBnSSpdQu49v3nFoGJk\nUWY95pQ/
BWQaX0q4KxzRVxup+7gwT5sKFXU+ktFtsGMYoDoSzbZS\n-----END RSA PRIVATE KEY-----\n",
      "user_id": "e4f380899b1248918f3d37098dc63746",
      "name": "demo123",
      "fingerprint": "49:ef:73:2b:9b:7f:2e:0c:58:d3:e3:42:8e:28:04:3b"
   }
}
```

- Importing the public key of the SSH key pair
  - Request example for importing the public key of an SSH key pair

    ```
    {
       "keypair": {
          "public_key": "ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAABAQCyNtFZM04PFxERvZU5OBKTKr3mtRZABe5/+zX81lTgDF
    CBfq6OXia47M4qXOa3ciBEKKZF+fMfs8U2UNB9aK1R/
    uORsoEFtxSgZnWG6p4Ct1vnrqwDD934VaDFPEn+h3JeAfvTB
    +Ag1YQ9zh9uYyE9Z3qZcC9+Ui93BDGdBtQeav4odxdwXcr2mT2jJV0nsocV0O4UcKM8BaIm8eqbcro
    ZEkyxqT3mUoSbmGx1hrngjBsP1ufgwJ6D85LFGQC1SjIOLvsR9i6v41BaLF8/
    kygvKOh2HlNVSMx38g52sTqoQ/xb3f8vR1VDXliAuD0frrG2Fy5wK4rOAnjuX9nh0bC9 Generated-
    by-Nova\n",
          "name": "demo3"
       }
    }
    ```

  - Response example for importing the public key of an SSH key pair

    ```
    {
       "keypair": {
          "public_key": "ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAABAQCyNtFZM04PFxERvZU5OBKTKr3mtRZABe5/+zX81lTgDF
    CBfq6OXia47M4qXOa3ciBEKKZF+fMfs8U2UNB9aK1R/
    uORsoEFtxSgZnWG6p4Ct1vnrqwDD934VaDFPEn+h3JeAfvTB
    +Ag1YQ9zh9uYyE9Z3qZcC9+Ui93BDGdBtQeav4odxdwXcr2mT2jJV0nsocV0O4UcKM8BaIm8eqbcro
    ZEkyxqT3mUoSbmGx1hrngjBsP1ufgwJ6D85LFGQC1SjIOLvsR9i6v41BaLF8/
    kygvKOh2HlNVSMx38g52sTqoQ/xb3f8vR1VDXliAuD0frrG2Fy5wK4rOAnjuX9nh0bC9 Generated-
    by-Nova\n",
          "user_id": "e4f380899b1248918f3d37098dc63746",
    ```



```
        "name": "demo1",
        "fingerprint": "b4:9a:c3:12:c4:90:bf:8e:7a:e2:70:10:c3:00:55:3f"
    }
}
```

or

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

- Importing the public key of the SSH key pair and managing the private key in the cloud
  - Request example for importing the public key of an SSH key pair and managing the private key in the cloud

```
{
    "keypair": {
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDb26UrW0htRbE/
Ygf8EPhzanBCc+5yEhkgmeSb2hTe48YRE5FdJKH6tueyj+vw5guoKjAITLjqZCqffGYXz/
7aXpFt244b9tTzh2l43uNtEZC+XZtc6KiBgfWupFI8O2i9YjJqdadsr+4Ad4AtIBbF++qsSJN4YycPX//
Gl8ja6AGPy4sdv8DZ40Gr8d+dMQ4pAsnUEtZ3jJ6NLdQU2CE1JhBdg3hbVbeh44gqQtSjhxWaSTlr
+NbVxSERtXXpsQWsid6qM1RhqH2+02cqXq5oNs4JLdu56pcTgSO5azTsGYJi6j5qp5BAdjMrFtHjbaeV
VWtkO1XQxfpueCJ470lx Generated-by-Nova\n",
        "name": "demo4",
        "key_protection": {
            "private_key": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEpQIBAAKCAQEA29ulK1tIbUWxP2IH/BD4c2pwQnPuchIZIJnkm9oU3uPGEROR\nXSSh
+rbnso/r8OYLqCowCEy46mQqn3xmF8/+2l6RbduOG/bU84dpeN7jbRGQvl2b
\nXOiogYH1rqRSPDtovWIyanWnbK/uAHeALSAWxfvqrEiTeGMnD1//xpfI2ugBj8uL\nHb/A2eNBq/
HfnTEOKQLJ1BLWd4yejS3UFNghNSYQXYN4W1W3oeOIKkLUo4cVmkk5\na/
jW1cUhEbV16bEFrIneqjNUYah9vtNnKl6uaDbOCS3bueqXE4EjuWs07BmCYuo+
\naqeQQHYzKxbR422nlVVrZDtV0MX6bngieO9JcQIDAQABAoIBAAVSEXM1KFGMqDdy
\ndeMBviF85+6Tw6d7DKSfVMr4whyKwpZTNOdeJZVdq8nEdd9Eke+l5bets6PofKeT
\naR0WaYJ7W2WfNJc0p/6kvkawjixrimcw+LuM3dcUgA+T5nGStnwuzi2JX13f/BCC
\n09VDu4lbCVjWAMufCqjyl8wEjFXP0Amhu8fpDvqHuhGvDkoVWRm9vDEeyz71P25K\n/
UUs7kXw5Qv0VRcm15b+2jO6tii3RTo+JaTvkYXoI/qrOjiQhQD88geiOPQVuffa
\nzJhDw4/2GdHaCwEN6mzwKCYCfcPTRbM503F1YlceiP9w2qScToao
+5B2okN9cIE8\nTV4vmIkCgYEA+vo/TKqFep2D6DPY3dNRu2bHIYikBMtYMCIKJ1bgQ1xS/FjEJfSj
\notdDcBEik+0VEV05BCHNduTiMt6rTUD9fpqDduV8PfsskAAZZjndbyUEGP/KrCvS\nJsd2BFa2G7In/
3wz3zw+3P77Aegb+zHJfDyqYWReNkYy5tSsaZ9Oz38CgYEA4EH5\nIXQPhJt9683JmK/
INiyhW6WqKOAZkvjKpUpdUpGdVHJI/9dfUYI9wxnybgAkOZVL\noErxMTdRCDev8nVAq/OwC/
4jI5YnBGkN3ZRxIkCrepzwxIXgWeJiwWqsvdDbosfM
\nN0Q6PvUnPTXEly8RGcH9ABTQvQ9Nq4rQyjQvXw8CgYEAuK8hmWb55iq3AE32zfVM\n9ZxB
+Jk2KRkBghnqYtx5Fth/cJZZcJy/NXs2cucJDNWvZSG2bSX6rfzrvwdAAw9J\n
+rn0968TaADG4AhSqHj4S2tvwn2oRF3SvRqV68drJqJI8KYS/bi1gaZYSyTkQkp2\nu
+dgcV6MPWAW4OmrHeZO9j0CgYEAkY8Az4/vipkKkIWP9oUSJOevDDdpTQK4VscZ
\ncVPlYvSU8/0CGN2IRvWMdRhgXLnGyYF4BuDd0J4hAH50u6ETiwivGfmogS6ywJAX\nqdzx2dOz
+e/n7wceafkhNH2zBbmM9glNKgok7DxfbcF6is7IALoDJ4xbOHu4ZEHD
\n55sbrE0CgYEAuOuiISgfbujENFFPW0nvUmNqbkAH5YW1oUIWYA+64z7wcWyvzxRS
\nYml2XLWyrJy3JNzHpLoe4mCBxz+HGrftZ0/qfQ/WDZrY/Djp7/xIkPyI9EwsRTYC
\nr1PtWvVws3y3hgdo6WVQMaeUqtLSiTugyuuPqidH+/QtwxObunNH6Ns=\n-----END RSA
PRIVATE KEY-----\n",
            "encryption": {
                "type": "kms",
                "kms_key_name": "testName"
            }
        }
    }
}
```

  - Response example for importing the public key of an SSH key pair and managing the private key in the cloud

```
{
    "keypair": {
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCyNtFZM04PFxERvZU5OBKTKr3mtRZABe5/+zX81lTgDF
CBfq6OXia47M4qXOa3ciBEKKZF+fMfs8U2UNB9aK1R/
uORsoEFtxSgZnWG6p4Ct1vnrqwDD934VaDFPEn+h3JeAfvTB
+Ag1YQ9zh9uYyE9Z3qZcC9+Ui93BDGdBtQeav4odxdwXcr2mT2jJV0nsocV0O4UcKM8BaIm8eqbcro
```



```
ZEkyxqT3mUoSbmGx1hrngjBsP1ufgwJ6D85LFGQC1SjIOLvsR9i6v41BaLF8/
kygvKOh2HlNVSMx38g52sTqoQ/xb3f8vR1VDXliAuD0frrG2Fy5wK4rOAnjuX9nh0bC9 Generated-
by-Nova\n",
        "user_id": "e4f380899b1248918f3d37098dc63746",
        "name": "demo1",
        "fingerprint": "b4:9a:c3:12:c4:90:bf:8e:7a:e2:70:10:c3:00:55:3f"
    }
}
```


or


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


Status Codes

For details, see **Status Codes**.

5.1.4 Deleting an SSH Key Pair (V2.1)

Function

This API is used to delete a specified SSH key pair based on the key pair name.

URI

- URI format

  DELETE /v2.1/{project_id}/os-keypairs/{keypair_name}

- Parameter description

  Table 5-15 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Name of an SSH key pair

Requests

None

Responses

None

Status Codes

For details, see **Status Codes**.

5.1.5 Modifying the Description of a Key Pair (V2.1)

Function

This API is used to modify the description of a specified SSH key pair based on the name of the SSH key pair.

URI

- URI format

  PUT /v2.1/{project_id}/os-keypairs/{keypair_name}

- Parameter description

  Table 5-16 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Name of an SSH key pair

Requests

Table 5-17 Request parameters

Parameter	Mandatory	Type	Description
description	Yes	String	Description of the SSH key pair (The value ranges from 0 to 255 characters.)

Responses

None

Examples

- Example request

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

- Example response

  None

Status Codes

For details, see **Status Codes**.

5.2 Managing SSH Key Pairs (V2)

5.2.1 Querying the List of SSH Key Pairs (V2)

Function

This API is used to query the SSH key pair list.

URI

- URI format
  GET /v2/{project_id}/os-keypairs
- Parameter description

Table 5-18 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

None

Responses

Table 5-19 Response parameters

Parameter	Mandatory	Type	Description
keypairs	Yes	Array of objects	SSH key pair information list. For details, see **Table 5-20**.

Table 5-20 keypairs field description

Parameter	Mandatory	Type	Description
keypair	Yes	Object	SSH key pair information, see **Table 5-21**.

Table 5-21 keypair field description

Parameter	Mandatory	Type	Description
fingerprint	Yes	String	Fingerprint information about an SSH key pair
name	Yes	String	Name of an SSH key pair
public_key	Yes	String	Public key information about an SSH key pair

Examples

The following example describes how to query the SSH key pair list.

- Example request

  None

- Example response

```
{
   "keypairs": [
      {
         "keypair": {
            "fingerprint": "15:b0:f8:b3:f9:48:63:71:cf:7b:5b:38:6d:44:2d:4a",
            "name": "keypair-601a2305-4f25-41ed-89c6-2a966fc8027a",
            "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQC+Eo/
RZRngaGTkFs7I62ZjsIlO79KklKbMXi8F+KITD4bVQHHn+kV
+4gRgkgCRbdoDqoGfpaDFs877DYX9n4z6FrAIZ4PES8TNKhatifpn9NdQYWA+IkU8CuvlEKGuFpKRi/
k7JLos/gHi2hy7QUwgtRvcefvD/vgQZOVw/mGR9Q== Generated-by-Nova\n"
         }
      }
   ]
}
```

Status Codes

For details, see **Status Codes**.

5.2.2 Querying Details About an SSH Key Pair (V2)

Function

This API is used to query a specified SSH key pair based on the key pair name.

URI

- URI format

  GET /v2/{project_id}/os-keypairs/{keypair_name}

- Parameter description

Table 5-22 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Name of an SSH key pair

Requests

None

Responses

Table 5-23 Response parameters

Parameter	Mandatory	Type	Description
keypair	Yes	Object	SSH key pair information. For details, see **Table 5-24**.

Table 5-24 keypair field description

Parameter	Mandatory	Type	Description
public_key	Yes	String	Public key information about an SSH key pair
name	Yes	String	Name of an SSH key pair
fingerprint	Yes	String	Fingerprint information about an SSH key pair
created_at	Yes	String	Time when an SSH key pair is created The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
deleted	Yes	Boolean	Tag that indicates an SSH key pair is deleted
deleted_at	Yes	String	Time when an SSH key pair is deleted The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
id	Yes	String	ID of an SSH key pair

Parameter	Mandatory	Type	Description
updated_at	Yes	String	Time when an SSH key pair is updated The value is a timestamp expressed in the number of seconds since 00:00:00 UTC on January 1, 1970.
user_id	Yes	String	User to whom an SSH key pair belongs

Examples

The following example describes how to query the SSH key pair details.

- Example request

  None

- Example response

```
{
    "keypair": {
        "created_at": "2014-05-07T12:06:13.681238",
        "deleted": false,
        "deleted_at": null,
        "fingerprint": "9d:00:f4:d7:26:6e:52:06:4c:c1:d3:1d:fd:06:66:01",
        "id": 1,
        "name": "keypair-3582d8b7-e588-4aad-b7f7-f4e76f0e4314",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDYJrTVpcMwFqQy/
oMvtUSRofZdSRHEwrsX8AYkRvn2ZnCXM+b6+GZ2NQuuWj+ocznlnwiGFQDsL/yeE+/
kurqcPJFKKp60mToXIMyzioFxW88fJtwEWawHKAclbHWpR1t4fQ4DS+/sIbX/
Yd9btlVQ2tpQjodGDbM9Tr9/+/3i6rcR+EoLqmbgCgAiGiVV6VbM2Zx79yUwd
+GnQejHX8BlYZoOjCnt3NREsITcmWE9FVFy6TnLmahs3FkEO/
QGgWGkaohAJlsgaVvSWGgDn2AujKYwyDokK3dXyeX3m2Vmc3ejiqPa/C4nRrCOlko5nSgV/
9IXRx1ERImsqZnE9usB Generated-by-Nova\n",
        "updated_at": null,
        "user_id": "fake"
    }
}
```

Status Codes

For details, see **Status Codes**.

5.2.3 Creating and Importing an SSH Key Pair (V2)

Function

This API is used to create an SSH key pair or import a public key to HUAWEI CLOUD to generate a key pair.

After an SSH key pair is created, you need to download the private key to a local directory. Then, you can use this private key to log in to an ECS. For ECS security purposes, the private key can be downloaded only once. Keep it secure.

URI

- URI format

  POST /v2/{project_id}/os-keypairs

- Parameter description

Table 5-25 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

NOTE

When creating an SSH key pair, you only need to set the **name**. When importing an SSH key pair, you must set the **public_key** parameter.

Table 5-26 Request parameters

Parameter	Mandatory	Type	Description
keypair	Yes	Object	Information about the created or imported SSH key pair. For details, see Table 5-27.

Table 5-27 keypair field description

Parameter	Mandatory	Type	Description
public_key	No	String	Character string of a public key to be imported
type	No	String	Type of a key pair. The value is **ssh** or **x509**.
name	Yes	String	Name of an SSH key pair. A new key pair cannot use the same name as an existing one. The value contains a maximum of 64 characters, including only letters, digits, underscores(_), and hyphens (-).
user_id	No	String	ID of the user to whom an SSH key pair belongs

Responses

Table 5-28 Response parameters

Parameter	Mandatory	Type	Description
keypair	Yes	Object	SSH key pair information. For details, see **Table 5-29**.

Table 5-29 keypair field description

Parameter	Mandatory	Type	Description
fingerprint	Yes	String	Fingerprint information about an SSH key pair
name	Yes	String	Name of an SSH key pair
public_key	Yes	String	Public key information about an SSH key pair
private_key	No	String	Private key information about an SSH key pair. • The information about the private key is contained in the response for creating an SSH key pair. • The information about the private key is not contained in the response for importing an SSH key pair.
user_id	Yes	String	ID of the user to whom an SSH key pair belongs

Examples

- Creating an SSH key pair
  - Request example for creating an SSH key pair

```
{    "keypair": {
        "type": "ssh",
        "name": "demo"
    }
}
```

  - Response example for creating an SSH key pair

```
{
   "keypair": {
      "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCyNtFZM04PFxERvZU5OBKTKr3mtRZABe5/+zX81lTgDF
CBfq6OXia47M4qXOa3ciBEKKZF+fMfs8U2UNB9aK1R/
uORsoEFtxSgZnWG6p4Ct1vnrqwDD934VaDFPEn+h3JeAfvTB
+Ag1YQ9zh9uYyE9Z3qZcC9+Ui93BDGdBtQeav4odxdwXcr2mT2jJV0nsocV0O4UcKM8BaIm8eqbcro
ZEkyxqT3mUoSbmGx1hrngjBsP1ufgwJ6D85LFGQC1SjIOLvsR9i6v41BaLF8/
```

```
kygvKOh2HlNVSMx38g52sTqoQ/xb3f8vR1VDXliAuD0frrG2Fy5wK4rOAnjuX9nh0bC9 Generated-by-Nova\n",
        "private_key": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEpAIBAAKCAQEAsjbRWTNODxcREb2VOTgSkyq95rUWQAXuf/s1/NZU4AxQgX6u
\njl4muOzOKlzmt3IgRCimRfnzH7PFNlDQfWitUf7jkbKBBbcUoGZ1huqeArdb566s\nAw/d
+FWgxTxJ/odyXgH70wfgINWEPc4fbmMhPWd6mXAvfllvdwQxnQbUHmr+KHcX
\ncF3K9pk9oyVdJ7KHFdDuFHCjPAWiJvHqm3K6GRJMsak95lKEm5hsdYa54IwbD9bn\n4MCeg/
OSxRkAtUoyDi77EfYur+NQWixfP5MoLyjodh5TVUjMd/
IOdrE6qEP8W93/\nL0dVQ15YgLg9H66xthcucCuKzgJ47l/Z4dGwvQIDAQABAoIBAQCdTjXL/
rVQLJQs\njKNDNnNu47NsCTvyl0nGPf+Rhb61ZSlKpH9/uyuC38O7MPWVx28jup3J9q7btNrG
\n7t6ZU+RpFAvbdyzb1pamXsoupLmEvESrZEsBCOhtY2fdsTG/Md+Ji0a1J6Z2VQG9\nbEviLC4S/
VwCRDwnzHOJInKIoJZroZv6SdK+KonQBS0Rq9bZrlvtBUUhaSGjBcJx\nmWKO78ikNOXP/
5Yl92SAw2vOYWhZdMZQrkp1EUFMG18Akuj+jC9QKXXfsLYYfzsQ
\nIGgpRdf6zYIV84QVMZ7NhQABM5DNmQfxrSIUSdbvOzOJzmShp41tH3sn9d+XS+bS
\nLloyuaQhAoGBAN7tpwgkcKddKI/Lp/CPqjkxP6lfO+xHEXjtnZd1Y//BavPSgq4v
\nWuFHgx1sPQK49KcSLZfF6UxkPw0KHBc5R9RkfYBAIdGNwENF2xyoYLLdnUtF4hRq
\n1q2DC3oklBZibH2tc6+hQ2aCWSeMvQbIvxTYV70EFzwR5f4O5LIskCm/AoGBAMyn
\nA7DOQdvcf4aexSYL4kGp70ERMOCtwr/d+O5RswARoyAQOxp4a7/TyFuGjnlT//bR
\nEYacXV1AieIdeJF3PgeUIR1QnUlNYD9Rufs14fs+5idQ7Evn1gvXv0HpBYTY7wNu
\nWTrWbsznY0fNIrGT4bQR6QpdvIuR5TBJf6HIAKyDAoGAFhKf3D2HbfraXkqC6V3A
\nNAN9Uy7bxwxOXZPha7Ky4QrspRGt4MNNk0q6X7ps3A0mJDi3jPSKoga2+3qJx37j
\nbtM4Xe97qb0IUWDkThUZ5fvtbBuSRAVEFlAIXeKrSwAZz+PRtY0ZGFhFrZXQzZAo
\n4058eXmjN05qYFpnKIEjEQ8CgYEAwELzW6oaAzR+dfk428p0UB4W0HkXAy0a9efS
\nUgpc8Oag6qF09SRGjdunshySQvegU78MCPtjVxUntE7dk0OD+di213SBn3jawAHG
\niHORjtkDndIPfCwcUdnpK0GAVtL6kK2dlIIZa9TB15WnT07Pzry4w21WkYSJ3Thf
\neJyNzYMCgYA8OvpKMdaEXFeNZWHDE1Q2VmpxvP/D6u6s4SBuyy8eac1qqku/s7zc\nsuFd/
o9wbBgzsf4eN8tNJ4bxrArRXvf9WyH7xd4PE3DvVJnz5S+8Nqj2Z0KCAqPD
\nibDbFxBYHcMIdwC2JBGQZIXpkST2jG9wZho5KghX4yiHSOPr2V25/g==\n-----END RSA PRIVATE KEY-----\n",
        "user_id": "6fc0d2cbbfab40b199874b97097e913d",
        "name": "demo",
        "fingerprint": "b4:9a:c3:12:c4:90:bf:8e:7a:e2:70:10:c3:00:55:3f"
    }
}
```

- Importing an SSH key pair
  - Request example for importing an SSH key pair

```
{
    "keypair": {
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCyNtFZM04PFxERvZU5OBKTKr3mtRZABe5/+zX81lTgDF
CBfq6OXia47M4qXOa3ciBEKKZF+fMfs8U2UNB9aK1R/
uORsoEFtxSgZnWG6p4Ct1vnrqwDD934VaDFPEn+h3JeAfvTB
+Ag1YQ9zh9uYyE9Z3qZcC9+Ui93BDGdBtQeav4odxdwXcr2mT2jJV0nsocV0O4UcKM8BaIm8eqbcro
ZEkyxqT3mUoSbmGx1hrngjBsP1ufgwJ6D85LFGQC1SjIOLvsR9i6v41BaLF8/
kygvKOh2HlNVSMx38g52sTqoQ/xb3f8vR1VDXliAuD0frrG2Fy5wK4rOAnjuX9nh0bC9 Generated-by-Nova\n",
        "type": "ssh",
        "name": "demo1",
        "user_id": "fake"
    }
}
```

  - Response example for importing an SSH key pair

```
{
    "keypair": {
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCyNtFZM04PFxERvZU5OBKTKr3mtRZABe5/+zX81lTgDF
CBfq6OXia47M4qXOa3ciBEKKZF+fMfs8U2UNB9aK1R/
uORsoEFtxSgZnWG6p4Ct1vnrqwDD934VaDFPEn+h3JeAfvTB
+Ag1YQ9zh9uYyE9Z3qZcC9+Ui93BDGdBtQeav4odxdwXcr2mT2jJV0nsocV0O4UcKM8BaIm8eqbcro
ZEkyxqT3mUoSbmGx1hrngjBsP1ufgwJ6D85LFGQC1SjIOLvsR9i6v41BaLF8/
kygvKOh2HlNVSMx38g52sTqoQ/xb3f8vR1VDXliAuD0frrG2Fy5wK4rOAnjuX9nh0bC9 Generated-by-Nova\n",
        "user_id": "6fc0d2cbbfab40b199874b97097e913d",
        "name": "demo1",
        "fingerprint": "b4:9a:c3:12:c4:90:bf:8e:7a:e2:70:10:c3:00:55:3f"
    }
}
```


Status Codes

For details, see **Status Codes**.

5.2.4 Deleting an SSH Key Pair (V2)

Function

This API is used to delete a specified SSH key pair based on the key pair name.

URI

- URI format
  DELETE /v2/{project_id}/os-keypairs/{keypair_name}
- Parameter description

  Table 5-30 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
keypair_name	Yes	String	Name of an SSH key pair

Requests

None

Responses

None

Status Codes

For details, see **Status Codes**.

5.2.5 Copying an SSH Key Pair (V2)

Function

A tenant may contain multiple users. This API is used to copy the key pair from the target user to the current user under the same tenant account.

URI

- URI format
  POST /v2/{project_id}/os-keypairs/copy
- Parameter description

Table 5-31 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Requests

Table 5-32 Request parameters

Parameter	Mandatory	Type	Description
user_name	Yes	String	Name of the target user under the same tenant account
force	No	Boolean	Indicates whether to forcibly overwrite an existing key pair.

Responses

Table 5-33 Response parameters

Parameter	Mandatory	Type	Description
changed	Yes	Integer	Number of copied key pairs.
success	Yes	Array of objects	List of key pairs that were successfully copied. For details, see **Table 5-34**.
failed	Yes	Array of objects	List of key pairs that failed to be copied. For details, see **Table 5-34**.

Table 5-34 success/failed field description

Parameter	Mandatory	Type	Description
keypair	Yes	String	Key pair name
message	Yes	String	Task information

Examples

- Example request


```
{
 "user_name": "kpsuser"
}
```

- Example response

```
{
    "changed": 2,
    "success": [
        {
            "keypair": "KeyPair-test1",
            "message": "imported"
        },
        {
            "keypair": "KeyPair-test2",
            "message": "imported"
        }
    ],
    "failed": [
        {
            "keypair": "KeyPair-test3",
            "message": "exist"
        }
    ]
}
```

or

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

Status Codes

For details, see **Status Codes**.

6 Application Examples

6.1 Example 1: Encrypting or Decrypting Small Volumes of Data

Scenario

Encrypt or decrypt data not larger than 4 KB, such as passwords, certificates, and phone numbers, by using a tool on the console or calling an API. This section describes how to call a KMS API and use a CMK to encrypt or decrypt data.

Process:

1. Create a CMK in KMS.
2. Call the encrypt-data API of KMS to encrypt plaintext data by using a CMK.
3. Deploy ciphertext certificates on your servers.
4. When your servers need to use a certificate, they call the decrypt-data API of KMS to decrypt the ciphertext data and obtain the ciphertext certificate.

Operations

APIs are called to perform the following operations:

- **Create a CMK**
- **Encrypt a DEK**
- **Decrypt a DEK**

Procedure

Step 1 Create a CMK.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/create-key

  For details, see **Creating a CMK**.

> **NOTE**
>
> Default Master Keys are created by services integrated with KMS. Names of Default Master Keys end with **/default**. Do not end your CMK names with **/default**.

- Example request

  POST: https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/create-key

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:

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

- Example response

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

Step 2 Encrypt data.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/encrypt-data

  For details, see **Encrypt data**.

- Example request

  POST https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/encrypt-data

  Obtain *{endpoint}* from **Regions and Endpoints**.

  You can use the API for **Querying the List of CMKs** to check key information, including key_id.

  Body:

```
{
   "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=="
}
```

Step 3 Decrypt data.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/decrypt-data

  For details, see **Decrypt data**.

- Example request

  POST https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/decrypt-data

Obtain *{endpoint}* from **Regions and Endpoints**.

You can use the API for **Querying the List of CMKs** to check key information, including key_id.

Body:

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

- Example response

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

----End

6.2 Example 2: Encrypting or Decrypting Large Volumes of Data

Scenario

Encrypt or decrypt a large amount of data.

- Encryption process:
  a. Create a CMK in KMS.
  b. Call the create-datakey API of the KMS to create a DEK. A plaintext DEK and a ciphertext DEK will be generated. The ciphertext DEK was generated by using a CMK to encrypt the plaintext DEK.
  c. Use the plaintext DEK to encrypt a plaintext file, generating a ciphertext file.
  d. Store the ciphertext DEK and the ciphertext file together in a permanent storage device or a storage service.
- Decryption process:
  a. Read the ciphertext DEK and the ciphertext file from the permanent storage device or storage service.
  b. Call the decrypt-datakey API and use the encryption CMK to decrypt the ciphertext DEK. The plaintext DEK will be generated.

     If the CMK is deleted, the decryption will fail. Properly keep your CMKs.
  c. Use the plaintext DEK to decrypt the ciphertext file.

Involved APIs

APIs used for the following operations are involved:

- **Create a CMK**
- **Create a DEK**

- **Encrypt a DEK**
- **Decrypt a DEK**

Procedure

Step 1 Create a CMK.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/create-key

  For details, see **Creating a CMK**.

  NOTE

  Default Master Keys are created by services integrated with KMS. Names of Default Master Keys end with **/default**. Do not end your CMK names with **/default**.

- Example request

  POST: https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/create-key

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:

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

- Example response

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

Step 2 Create a DEK.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/create-datakey

  For details, see **Creating a DEK**.

- Example request

  POST https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/create-datakey

  Obtain *{endpoint}* from **Regions and Endpoints**.

  You can use the API for **Querying the List of CMKs** to check key information, including key_id.

  Body:

```
{
    "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
3662302D653732372D346439632D623335642D66383462623437346133376600000000045B05321483B
D9F9561865EE7DFE9BE267A42EB104E98C16589CE46940B18E52"
}
```

Step 3 Encrypt the DEK.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/encrypt-datakey

  For details, see **Encrypting a DEK**.

- Example request

  POST https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/encrypt-datakey

  Obtain *{endpoint}* from **Regions and Endpoints**.

  You can use the API for **Querying the List of CMKs** to check key information, including key_id.

  Body:

```
{
    "key_id": "0d0466b0-e727-4d9c-b35d-f84bb474a37f",
    "plain_text":
"00000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000F5A5FD42D16A20302798EF6ED309979B43003
D2320D9F0E8EA9831A92759FB4B",
    "datakey_plain_length": "64"
}
```

- Example response

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

Step 4 Decrypt the DEK.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/decrypt-datakey

  For details, see **Decrypting a DEK**.

- Example request

  POST https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/decrypt-datakey

  Obtain *{endpoint}* from **Regions and Endpoints**.

  You can use the API for **Querying the List of CMKs** to check key information, including key_id.

  Body:

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

```
62302D653732372D346439632D623335642D66383462623437346133376600000000D34457984F9730
D57F228C210FD22CA6017913964B21D4ECE45D81092BB9112E"
}
```

- Example response

```
{
    "data_key":
"00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000",
    "datakey_length": "64",
    "datakey_dgst": "F5A5FD42D16A20302798EF6ED309979B43003D2320D9F0E8EA9831A92759FB4B"
}
```

----End

6.3 Example 3: Querying Information About Keys

Scenario

Use KMS APIs to obtain the list of keys, key information, key instances, and key tags.

Involved APIs

- **Query key list**
- **Query key information**
- **Query key instance**
- **Query key tags** Tag Management Service (TMS) needs to query all the tags of a specified CMK.

Procedure

Step 1 Query the list of keys.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/list-keys

  For details, see "Querying Key List".

- Example request

  POST: https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/list-keys

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:

```
{
   "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
}
```


Step 2 Query the information about keys.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/describe-key

  For details, see "Querying Key Details".

- Example request

  POST: https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/describe-key

  Obtain *{endpoint}* from **Regions and Endpoints**.

  You can use the API for **Querying the List of CMKs** to check key information, including key_id.

  Body:

  ```
  {
    "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",
      "key_rotation_enabled":"false",
      "sys_enterprise_project_id ": "0",
```



```
    }
}
```

Step 3 Query CMK instances.

- API information

  URI format: POST /v1.0/*{project_id}*/kms/resource_instances/action

  For details, see "Querying Key Instances".

- Example request

  POST: https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms//resource_instances/action

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:

```
{
      "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,
   "key_rotation_enabled" : false,
   "expiration_time" : "",
   "origin" : "kms",
   "sys_enterprise_project_id" : "0",
   "realm" :  "cn-hongkong-7"
  } ],
  "resource_name" : "tagTest_xmdmi",
  "tags" : [ {
   "key" : "key",
   "value" : "testValue!"
  }, {
   "key" : "haha",
   "value" : "testValue"
  } ]
 } ],
 "total_count" : 1
}
```


Step 4 Query the tags of a key.

- API information

  URI format: GET /v1.0/{project_id}/kms/{key_id}/tags

  For details, see "Querying Key Tags".

- Example request

  GET: https://*{endpoint}*/v1.0/53d1aefc533f4ce9a59c26b01667cbcf/kms/94752282-805e-4032-ada8-34966f70e02f/tags

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:

  None

- Example response

```
{
       "tags": [
        {
         "key": "key1",
         "value": "value1"
         },
         {
         "key": "key2",
         "value": "value3"
         }
         ],
         "existTagsNum":2
}
```

----End

7 Permissions Policies and Supported Actions

7.1 Introduction

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

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

Permissions are classified into **roles** and **policies** based on the authorization granularity. 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. For example, if an IAM user wants to use an API to query the SSH keys of account key pairs, the user must be granted permissions that allow the **kps:domainKeypairs:list** action.

Supported Actions

DEW provides system-defined policies that can be directly used in IAM. You can also create custom policies and use them to supplement system-defined policies, implementing more 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.

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

- **Manage keys**, such as creating keys, querying keys, and creating grants.
- **Manage key pairs**, such as creating, querying, and deleting key pairs.

7.2 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	-	√	√
Creating a grant	POST /v1.0/{project_id}/kms/create-grant	kms:grant:create	-	√	√
Revoking a grant	POST /v1.0/{project_id}/kms/revoke-grant	kms:grant:revoke	-	√	√
Retiring a grant	POST /v1.0/{project_id}/kms/retire-grant	kms:grant:retire	-	√	√
Querying the grant list of a CMK	POST /v1.0/{project_id}/kms/list-grants	kms:grant:list	-	√	√
Querying the list of grants that can be retired	POST /v1.0/{project_id}/kms/list-retirable-grants	kms:grant:list	-	√	√
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	-	√	√

Permission	API	Action	Dependent Permission	IAM Project (Project)	Enterprise Project (Enterprise Project)
Deleting key material	POST /v1.0/{project_id}/kms/delete-imported-key-material	kms:cmk:deleteMaterial	-	√	√
Enabling key rotation	POST /v1.0/{project_id}/kms/enable-key-rotation	kms:cmk:enableRotation	-	√	√
Modifying the rotation interval	POST /v1.0/{project_id}/kms/update-key-rotation-interval	kms:cmk:updateRotation	-	√	√
Disabling key rotation	POST /v1.0/{project_id}/kms/disable-key-rotation	kms:cmk:disableRotation	-	√	√
Querying the key rotation status	POST /v1.0/{project_id}/kms/get-key-rotation-status	kms:cmk:getRotation	-	√	√
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	-	√	√
Adding tags to a key	POST /v1.0/{project_id}/kms/{key_id}/tags	kms:cmkTag:create	-	√	√

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

7.3 Key Pair Management

Permission	API	Action	Dependent Permission	IAM Project	Enterprise Project
Creating and importing an SSH key pair (native OpenStack API)	POST /v2.1/{project_id}/os-keypairs	ecs:serverKeypairs:create	-	√	x
Querying an SSH key pair (native OpenStack API)	GET /v2.1/{project_id}/os-keypairs/{keypair_name}	ecs:serverKeypairs:get	-	√	x
Querying SSH key pairs (native OpenStack API)	GET /v2.1/{project_id}/os-keypairs	ecs:serverKeypairs:list	-	√	x
Deleting an SSH key pair (native OpenStack API)	DELETE /v2.1/{project_id}/os-keypairs/{keypair_name}	ecs:serverKeypairs:delete	-	√	x

A Appendix

A.1 Status Codes

Status Code	Status	Description
200	OK	Request processed successfully.
202	Accept	The job was successfully delivered. However, it will be postponed because the system is busy currently.
204	No Content	The request is processed successfully and no content is returned.
300	multiple choices	The requested resource has multiple available responses.
400	Bad Request	The request parameter is incorrect.
401	Unauthorized	You need to enter the username and password to access the requested page.
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.
405	Method Not Allowed	The method specified in the request is not allowed.
406	Not Acceptable	The response generated by the server cannot be accepted by the client.
407	Proxy Authentication Required	You must use the proxy server for authentication. Then, the request can be processed.

Status Code	Status	Description
408	Request Timeout	The request timed out.
409	Conflict	The request cannot be processed due to a conflict.
500	Internal Server Error	Internal service error.
501	Not Implemented	Failed to complete the request. The server does not support the requested function.
502	Bad Gateway	Failed to complete the request, because the server receives an invalid request.
503	Service Unavailable	Failed to complete the request due to system exception.
504	Gateway Timeout	A gateway timeout error occurs.

A.2 Error Code

Status Code	Error Code	Message	Description	Measure
400	CSMS.0002	The value of parameter is invalid.	The XX parameter is invalid.	Enter a valid parameter.
400	CSMS.0003	The request is invalid.	Invalid request.	Enter a valid URL.
400	CSMS.0004	The requested body format is wrong.	The request body format is incorrect.	Enter a valid body.
400	CSMS.0005	The resource does not exist.	The resource does not exist.	Enter valid secrets.
400	CSMS.0103	The secret is in the scheduled deletion state.	The secret is in Pending deletion state and cannot be used.	Ensure that the secret is in Enabled or Disabled state.
400	CSMS.0105	Can not delete the system internal stage.	The built-in version status of the system cannot be deleted.	Do not delete the built-in version.

Status Code	Error Code	Message	Description	Measure
400	CSMS.0106	The secret name not found int the db.	The secret name does not exist in the database.	Enter a valid secret name.
400	CSMS.0107	The secret is in schedule delete state.	The secret is in Pending deletion state.	Enter other secrets.
400	CSMS.0108	The secret is not in schedule delete state.	The secret is not in Pending deletion state.	Enter other secrets.
400	CSMS.0202	The number of secret has reached the upper limit.	The number of secrets reaches the upper limit.	Delete other secrets.
400	CSMS.0203	The number of stage has reached the upper limit.	The number of secret version statuses has reached the upper limit.	Delete other version statuses.
400	CSMS.0204	The number of stage is greater than the quota limit.	The number of secret version statuses has exceeded the upper limit.	Delete other version statuses.
400	CSMS.0301	Invalid X-Auth-Token.	Invalid **X-Auth-Token**.	Obtain the token again and ensure the token string is complete.
400	CSMS.0401	Can not get the protected secret value using the provided KMS key.	Failed to encrypt or decrypt the secret value by using KMS.	Ensure that the key is enabled.
403	CSMS.0109	The secret state is not enabled.	The secret is not enabled.	Enable the secret.

Status Code	Error Code	Message	Description	Measure
403	CSMS.0302	The user role has no permission to access the interface.	The user role has no permission to access the API.	Contact the administrator to grant CSMS full access permissions.
404	CSMS.0205	Version quota not found for secret	No version quota is found for secret XX.	The secret has been deleted. Select another secret.
404	CSMS.0206	Stage quota not found for secret	No version status quota is found for secret XX.	The secret has been deleted. Select another secret.
404	CSMS.0207	Secret quota not found.	No secret quota found.	The secret has been deleted. Select another secret.
409	CSMS.0101	The secret name already exists.	The secret name is in use.	Enter another valid secret name.
409	CSMS.0102	The version id already exists.	The secret version number already exists.	Enter another valid secret version number.
409	CSMS.0104	The stage name already exists.	The version status name already exists.	Enter another valid version status name.
500	CSMS.0001	An internal error occurred.	Internal service error.	Try again later or contact customer support.
500	CSMS.0006	AES encrypt secret value occurred an error.	An error occurs while the secret value is encrypted using AES.	Try again later or contact customer support.
500	CSMS.0201	The number of secret has reached the upper limit.	The quota exceeds the upper limit.	Delete other secrets.

Status Code	Error Code	Message	Description	Measure
400	KMS.0105	A system exception occurred. Contact technical support.	A system exception occurs. Contact technical support.	Contact technical support.
400	KMS.0106	It is replica service, readonly api allowed.	The backup service allows only read-only operations.	The backup service allows only read-only operations.
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.
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 scheduled to be deleted and is unavailable.	Enable the key.

Status Code	Error Code	Message	Description	Measure
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.0212	Invalid resource type.	Invalid resource type.	Use the correct resource type.
400	KMS.0214	The request format is invalid.	Invalid request format.	Use the correct request format.
400	KMS.0308	Invalid parameter.	Invalid parameter.	Enter valid parameters.
400	KMS.0309	External keys required.	External keys required.	Use an imported key.
400	KMS.0310	The key is not in Pending import state.	The key is not scheduled to be imported.	Ensure the key is in the **Pending** 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 support.
400	KMS.0312	External keys cannot be rotated.	External keys cannot be rotated.	Use a common CMK.
400	KMS.0313	Key rotation is not enabled.	Key rotation is not enabled.	Enable key rotation.
400	KMS.0315	Invalid partition_id.	Invalid partition type.	Enter the correct partition type.
400	KMS.0317	The cmk partition is not enabled.	The key partition type is not enabled.	Enter the type of the partition to be enabled.
400	KMS.0318	Partition name has exist.	The partition type name already exists.	Enter a correct partition type name.
400	KMS.0319	Rotation not supported in the current KMS version.	Rotation is not supported in the current KMS version.	Try again later or contact customer support.

Status Code	Error Code	Message	Description	Measure
400	KMS.0320	Resource frozen.	Resource frozen.	Renew the service and try again.
400	KMS.0323	Failed to obtain the partition of the key.	Failed to obtain the partition of the key.	Try again later or contact customer support.
400	KMS.0324	RSA keys cannot be rotated.	RSA keys cannot be rotated.	Use a common CMK.
400	KMS.0325	The asymmetric key is not support this operation.	Asymmetric keys do not support this operation.	Try again using an asymmetric key.
400	KMS.0327	Failed to obtain user permissions.	Failed to obtain user permissions.	Contact the administrator to grant the KMS CMK full access permission to the account.
400	KMS.0329	Hash algorithm does not match the digest length.	Hash algorithm does not match the digest length.	Enter valid parameters or contact customer support.
400	KMS.0331	The symmetric key is not support this operation.	Symmetric keys do not support this operation.	Use a correct key type and try again.
400	KMS.0332	This key is not support the signing algorithm.	The key does not support the signature algorithm.	Use a correct signature algorithm or contact technical support.
400	KMS.0333	The signing algorithm SM2DSA_SM3 not support RAW signing.	The signature algorithm SM2DSA_SM3 does not support RAW signatures.	Use a correct signature algorithm or contact technical support.

Status Code	Error Code	Message	Description	Measure
400	KMS.0334	The cmk is used to encrypt and decrypt.	This key is used for encryption and decryption.	This key is used for encryption and decryption.
400	KMS.0335	The cmk is used to sign and verify.	This key is used for signature and verification.	This key is used for signature and verification.
400	KMS.0336	The current region does not support SM4.	SM4 is not supported in the current region.	Replace the algorithm or contact technical support.
400	KMS.0337	The current region does not support SM2.	SM2 is not supported in the current region.	Replace the algorithm or contact technical support.
400	KMS.0338	The custom keystore do not support rotation.	The user-defined keystore does not support rotation.	Contact technical support.
400	KMS.0339	The custom keystore do not support import key.	Keys cannot be imported to a custom keystore.	Contact technical support.
400	KMS.0340	not support keystore.	Currently, the keystore is not supported.	Contact technical support.
400	KMS.0401	Tag list cannot be empty.	Tag list cannot be empty.	Enter valid parameters.
400	KMS.0402	Invalid match value.	Invalid match value.	Enter valid parameters.
400	KMS.0403	Invalid match key.	Invalid match key.	Enter valid parameters.
400	KMS.0404	Invalid action.	Invalid action.	Enter valid parameters.
400	KMS.0405	Invalid tag value.	Invalid tag value.	Enter valid parameters.

Status Code	Error Code	Message	Description	Measure
400	KMS.0406	Invalid tag key.	Invalid tag key.	Enter valid parameters.
400	KMS.0407	Invalid tag list size.	Invalid tag list size.	Enter valid parameters.
400	KMS.0408	Invalid resourceType.	Invalid resourceType.	Enter valid parameters.
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 valid parameters.
400	KMS.0411	Invalid tag key length.	Invalid tag key length.	Enter valid parameters.
400	KMS.0412	Invalid tag list.	Invalid tag list.	Enter valid parameters.
400	KMS.0413	Too many tag values.	The length of the values list in the tag exceeded the upper limit.	Enter valid parameters.
400	KMS.0414	Invalid tags.	Invalid tags.	Enter valid parameters.
400	KMS.0415	Invalid matches.	Invalid matches.	Enter valid parameters.
400	KMS.0417	Invalid offset.	Invalid offset.	Enter valid parameters.
400	KMS.0418	Offset is not required.	No offset is required.	Enter valid parameters.
400	KMS.1101	Invalid key_alias.	Invalid key_alias.	Enter valid parameters.
400	KMS.1102	Invalid realm.	Invalid realm.	Enter valid parameters.
400	KMS.1103	Invalid key_description.	Invalid key_description.	Enter valid parameters.
400	KMS.1104	Duplicate key aliases.	The key alias already exists.	Use another alias.

Status Code	Error Code	Message	Description	Measure
400	KMS.1105	Too many keys.	The number of keys has reached the upper limit.	Increase key quota or delete unnecessary keys.
400	KMS.1108	Failed to create the default partition for the key.	Failed to create the default partition for the key.	Try again later or contact customer support.
400	KMS.1109	Failed to create the route for the key.	Failed to create the route for the key.	Try again later or contact customer support.
400	KMS.1110	Invalid alg_type.	The algorithm type is invalid.	Use a correct algorithm type.
400	KMS.1114	EC keys do not support to encrypt/ decrypt.	The key cannot be encrypted or decrypted.	Use the correct key for encryption and decryption.
400	KMS.1115	Symmetric keys do not support to sign/verify.	Symmetric keys do not support signature/ verification.	Try again using an asymmetric key.
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.	The scheduled deletion time ranges from 7 days to 1,096 days.	Enter valid parameters.
400	KMS.1402	The key is already in Pending deletion state.	The key is already scheduled to be deleted.	No further operation required.

Status Code	Error Code	Message	Description	Measure
400	KMS.1404	This region is not the original region of the key, and it is not allowed to import-key-material.	The key material cannot be imported because this region is not the original region of the CMK.	The current site does not support the import of key materials.
400	KMS.1501	The key is not in Pending deletion state.	Basic-edition KMS cannot be deleted.	Schedule deletion the key.
400	KMS.1601	Invalid limit.	Invalid limit.	Enter valid parameters.
400	KMS.1602	marker must be greater than or equals 0.	The value of **marker** must be greater than or equal to **0**.	Enter valid parameters.
400	KMS.1603	Invalid offset.	Invalid offset.	Enter valid parameters.
400	KMS.1801	random_data_length must be 512 bits.	random_data_length must be 512 bits.	Enter valid parameters.
400	KMS.1802	random_data_length must be a multiple of 8.	**random_data_length** must be a multiple of 8.	Enter valid parameters.
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 valid parameters.
400	KMS.1902	key_spec can only be AES_128 or AES_256.	The value of **key_spec** must be **AES_128** or **AES_256**.	Enter valid parameters.
400	KMS.1903	datakey_length must be a multiple of 8.	**datakey_length** must be a multiple of 8.	Enter valid parameters.

Status Code	Error Code	Message	Description	Measure
400	KMS.1904	The rsa wrapping data key is over length 1520 bits.	The RSA wrapping data key exceeds 1,520 bytes.	Enter valid parameters.
400	KMS.2001	datakey_length must be 512 bits.	**datakey_length** must be 512 bits.	Enter valid parameters.
400	KMS.2101	Invalid plain_text.	Invalid plain text.	Enter valid parameters.
400	KMS.2102	datakey_plain_length must be 64 bytes.	**datakey_plain_length** must be 64 bytes.	Enter valid parameters.
400	KMS.2103	Failed to verify the DEK hash.	Failed to verify the DEK hash.	Ensure the DEK is valid and try again, or contact customer support.
400	KMS.2104	The length of plain_text does not match datakey_plain_length.	The length of **plain_text** does not match **datakey_plain_length**.	Enter valid parameters.
400	KMS.2105	The symmetric key not support this encryption algorithm.	The symmetric key does not support this encryption algorithm.	Enter valid parameters.
400	KMS.2106	The rsa key not support this encryption algorithm.	The rsa key does not support this encryption algorithm.	Enter valid parameters.
400	KMS.2107	The sm2 key not support this encryption algorithm.	The SM2 key does not support this encryption algorithm.	Enter valid parameters.

Status Code	Error Code	Message	Description	Measure
400	KMS.2108	The rsa encryption plain_text is over length.	The length of rsa encryption plain_text is too long.	Enter valid parameters.
400	KMS.2201	Invalid cipher_text.	Invalid cipher text.	Enter valid parameters.
400	KMS.2202	datakey_cipher_length must be 64 bytes.	datakey_cipher_length must be 64 bytes.	Enter valid parameters.
400	KMS.2203	Failed to verify the DEK hash.	Failed to verify the DEK hash.	Ensure the DEK is valid and try again, or contact customer support.
400	KMS.2204	The length of cipher_text does not match datakey_cipher_length.	The length of **cipher_text** does not match **datakey_cipher_length**.	Enter valid parameters.
400	KMS.2301	The quota value is beyond the maximum configurable limit.	The quota value exceeds the upper limit.	Enter valid parameters.
400	KMS.2302	New quota value must not less than old.	The new quota cannot be smaller than the old quota.	Enter valid parameters.
400	KMS.2303	The quota type is not supported.	The quota type is not supported.	Enter valid parameters.
400	KMS.2304	Can not update grant quota when no grant has created.	If no grant is created, the grant quota cannot be updated.	Create a grant and try again.
400	KMS.2305	Can not update cmk quota when no cmk has created.	The CMK quota cannot be updated if no CMK is created.	Create a key and try again.

Status Code	Error Code	Message	Description	Measure
400	KMS.2401	Specify an operation in addition to create-grant.	The operation cannot contain only create-grant.	Enter valid parameters.
400	KMS.2402	Invalid user ID.	Invalid user ID.	Enter valid parameters.
400	KMS.2403	Failed to create the grant.	Failed to create the grant.	Try again later or contact customer support.
400	KMS.2404	Too many CMK grants.	No more permissions can be generated for the CMK.	Increase grant quota or delete unnecessary grants.
400	KMS.2405	Too many grants.	Too many grants.	Increase grant quota or delete unnecessary grants.
400	KMS.2406	The basic partition has no right to create grant.	The basic partition does not have the permission to create authorization.	Contact technical support.
400	KMS.2501	Invalid grant ID.	Invalid grant ID.	Enter a valid grant ID.
400	KMS.2502	grant_id and key_id do not match.	grant_id and key_id do not match.	Ensure input grant_id matches key_id.
400	KMS.2601	Token expired.	Token is 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.

Status Code	Error Code	Message	Description	Measure
400	KMS.2604	The external key plaintext length must be 32 bits.	The external key plaintext length must be 32 bits.	Enter valid parameters.
400	KMS.2605	Token verification failed.	Failed to verify token.	Obtain a new token.
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.2607	The import sm4 plain key length must 16.	The length of the imported SM4 pure key must be 16.	Enter valid parameters.
400	KMS.2608	The imported asymmetric private key can not be null.	The imported asymmetric private key cannot be empty.	Enter valid parameters.
400	KMS.2609	The imported asymmetric private is invalid.	The imported asymmetric private is invalid.	Enter valid parameters.
400	KMS.2610	The temporary key length must be 16 or 32.	The length of the temporary key must be 16 or 32.	Enter valid parameters.
400	KMS.2701	Key material is not in Enabled or Disabled state and cannot be deleted.	Key materials can be deleted only when they are enabled or disabled.	Ensure that the key is in Enabled or Disabled state.

Status Code	Error Code	Message	Description	Measure
400	KMS.2702	The imported private key material can not be deleted.	The imported private key material cannot be deleted.	The imported private key material cannot be deleted.
400	KMS.2801	End_time must bigger than start_time.	End time must be later than start time.	Enter valid parameters.
400	KMS.2901	Key rotation is not disabled.	Key rotation is not disabled.	Disable key rotation.
400	KMS.3001	Invalid rotation_interval.	Invalid rotation interval.	Enter valid parameters.
400	KMS.3103	Invalid id of tenant.	Invalid tenant ID.	Enter a valid tenant ID.
400	KMS.3201	Generate order id failed.	Failed to generate the order.	Try again or contact technical support.
400	KMS.3202	KMS error.	The order resource is incorrect.	Try again or contact technical support.
400	KMS.3702	Invalid kms status parameter.	The KMS status parameter is invalid.	Enter valid parameters.
400	KMS.3801	KMS key sync enable and configuration failed.	Failed to synchronize and configure KMS keys.	Try again or contact technical support.
400	KMS.3802	Failed to sync operation convert to born region.	Currently, the keystore is not supported.	Contact technical support.
400	KMS.3803	The key of born region is turn off sync ,so no operating can be performed.	Keystore is not supported.	Contact technical support.

Status Code	Error Code	Message	Description	Measure
400	KMS.3804	AttestationDocument parsing failed.	Failed to parse the document.	Try again or contact technical support.
400	KMS.3902	Invalid Service-Transaction-Id in request header.	The **Service-Transaction-Id** parameter in the request header is invalid.	Contact customer support.
400	KMS.3903	Service-Transaction-Id is missing in request header.	The request header does not contain the Service-Transaction-Id parameter.	Contact customer support.
400	KMS.4001	Service name to notify is illegal.	The service name used for notification is invalid.	Contact customer support.
400	KMS.4002	Service url to notify is illegal.	The service URL used for notification is invalid.	Contact customer support.
400	KMS.5022	Tags are not compliant.	The tags are not compliant.	Try again or contact technical support.
400	KMS.5023	Pdp5 Header is invalid.	The Header parameter is invalid.	Try again or contact technical support.
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.

Status Code	Error Code	Message	Description	Measure
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 administrator to grant the KMS CMKFullaccess permission to the account.
403	KMS.0307	No access permissions.	No access permissions.	Contact the administrator to grant the KMS CMKFullaccess permission to the account.
403	KMS.0314	Real-name authentication is required to access the API.	Real-name authentication is required to access the API.	Complete real-name authentication and try again.
403	KMS.0321	URIs in URL and X-Auth-Token do not match.	URIs in URL and X-Auth-Token do not match.	Ensure the URI in the URL matches that in X-Auth-Token.
403	KMS.0322	The user has no permission to access the partition.	The user does not have the permission on the partition type.	Configure the permission and try again.
403	KMS.0326	No access permissions.	No access permissions.	Contact the administrator to grant the KMS CMKFullaccess permission to the account.

Status Code	Error Code	Message	Description	Measure
403	KMS.0328	KMS has been frozen. Renew it and try again.	KMS has been frozen. Renew it and try again.	Renew the service and try again.
403	KMS.0330	User has no permission.	The user does not have related permissions.	Configure the KMS CMKFullaccess permission and try again.
404	KMS.0207	The key does not exist.	The key does not exist.	Choose a valid key or create a key.
404	KMS.0316	No exist partition_id.	The partition type does not exist.	Enter the partition type.
404	KMS.0416	Invalid tag ID.	Invalid tag ID.	Enter a valid key tag.
404	KMS.3901	The requested jobId cannot be found.	The task ID is not found.	Contact technical support.
405	KMS.0215	Request method not supported.	Invalid request method.	Use a supported request method.
500	KMS.0100	Get-status error.	Failed to obtain the service status.	Contact technical support.
500	KMS.0101	KMS error.	KMS service error.	Try again later or contact customer support.
500	KMS.0102	Abnormal KMS I/O.	Abnormal KMS I/O.	Try again later or contact customer support.
500	KMS.0103	A system exception occurred. Contact technical support.	A system exception occurs. Contact technical support.	Contact technical support.
500	KMS.3805	Failed to build digital envelope.	The digital envelope cannot be constructed.	Try again or contact technical support.

Status Code	Error Code	Message	Description	Measure
500	KMS.3806	Unable to obtain enclave root certificate, please contact technical support.	If the root certificate cannot be obtained, contact technical support.	Contact technical support.
500	KMS.5012	Request PDP service failed.	Failed to access the PDP service.	Try again or contact technical support.
500	KMS.5021	Check pdp5 auth failed.	PDP5 authentication fails.	Try again or contact technical support.
503	KMS.0104	A system exception occurred. Contact technical support.	A system exception occurs. Contact technical support.	Contact technical support.

A.3 Obtaining a Project ID

Obtaining a Project ID by Calling an API

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

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 **Basic Information** from the drop-down list.
3. On the **Account Info** page, click **Manage** next to **Security Credentials**.
   On the **My Credentials** page, view project IDs in the project list.

Figure A-1 Viewing project IDs