Simple Message Notification

API Reference

Issue	01
Date	2025-02-13

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

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

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

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

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

Welcome to the *Simple Message Notification API Reference*. Simple Message Notification (SMN) is a reliable and flexible large-scale message notification service. It enables you to efficiently send messages to email addresses, phone numbers, and HTTP/HTTPS servers and connect cloud services through notifications, reducing system complexity.

You can use APIs described in this document to manage SMN resources, such as creating, querying, deleting, and updating resources. For details about all supported operations, see **2 API Overview**.

If you plan to access SMN through an API, ensure that you are familiar with basic SMN concepts. For details, see **Service Overview**.

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**.

Concepts

- Account

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

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

  API authentication requires information such as the account name, username, and password.
- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP

(EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

For details, see **Region and AZ**.

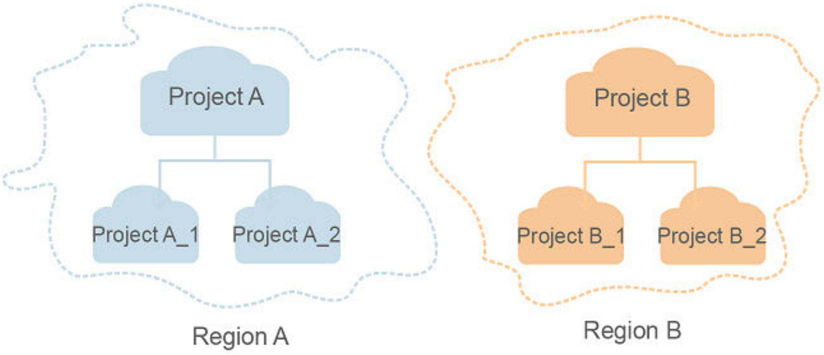
- AZ

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

- Project

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

Figure 1-1 Project isolating model



- Enterprise Project

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

  For details about enterprise projects and about how to obtain enterprise project IDs, see **Enterprise Management User Guide**.

2 API Overview

By calling SMN APIs, you can use all functions provided by SMN, for example, creating a topic and adding a subscription.

Table 2-1 API description

Category	Description
Topics	Create, update, and delete a topic.
Subscriptions	Query subscriptions, add a subscription, and delete a subscription.
Templates	Create, modify, and delete a message template.
Message publishing	Publish a message, publishing a detection message, and obtain the HTTP detection result.
Resource tags	Query resources by tag, add resource tags, and query tags.
Version querying	Query SMN API versions and information about the SMN API v2 version.
Cloud logs	Bind a cloud log to a topic, query cloud logs, and update a cloud log.
Message filter policies of subscribers	Create message filter policies for a subscriber, update message filter policies of subscribers, and delete message filter policies of subscribers.

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

Table 3-1 URI parameter description

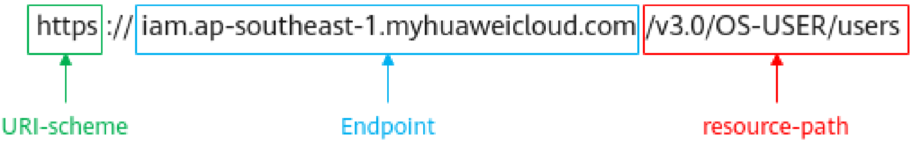
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in region **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**.

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

IAM is a global service. You can create an IAM user using the endpoint of IAM in any region. For example, to create an IAM user in the **CN-Hong Kong** region, obtain the endpoint of IAM (**iam.ap-southeast-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3.0/OS-USER/users**) in the URI of the API for **creating an IAM user**. Then construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

Figure 3-1 Example URI



NOTE

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

Request Methods

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

Table 3-2 HTTP methods

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

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

For example, in the case of the API for **creating an IAM user**, the request method is **POST**. An example request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

Request Header

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

Common request header fields are as follows.

Table 3-3 Common request header fields

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

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **8.1 Obtaining a Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for **obtaining a user token** (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

NOTE

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

For more details, see "Authentication Using AK/SK" in **3.2 Authentication**.

The following shows an example request of the API for **creating an IAM user** when AK/SK authentication is used:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=****************, SignedHeaders=content-type;host;x-sdk-date,
Signature=****************
```

(Optional) Request Body

This part is optional. A request body is generally sent in a structured format (for example, JSON or XML), which is specified by **Content-Type** in the request header. It is used to transfer content other than the request header. If the request body contains full-width characters, these characters must be coded in UTF-8.

The request body varies depending on APIs. Certain APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

The following shows an example request (a request body included) of the API for **creating an IAM user**. You can learn about request parameters and related description from this example. The bold parameters need to be replaced for a real request.

- **accountid**: account ID of an IAM user
- **username**: name of an IAM user
- **email**: email of an IAM user
- **password**: login password of an IAM user

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=****************, SignedHeaders=content-type;host;x-sdk-date,
Signature=****************

{
    "user": {
        "domain_id": "accountid",
        "name": "username",
        "password": "**********",
        "email": "email",
        "description": "IAM User Description"
    }
 }
```

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

3.2 Authentication

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

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

AK/SK Authentication

NOTE

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

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

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

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

NOTE

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

Token Authentication

NOTE

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

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API. You can obtain a token by calling the **Obtaining User Token** API.

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

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",    // IAM user name
                    "password": $ADMIN_PASS,  //IAM user password. You are advised to store it in ciphertext in
the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "domain": {
                        "name": "domainname"  // Name of the account to which the IAM user belongs
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"    // Project name
            }
        }
    }
}
```

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

3.3 Response

Status Code

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

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **A.2.2 Status Codes**.

For example, if status code **201** is returned for calling the API used to **create an IAM user**, the request is successful.

Response Header

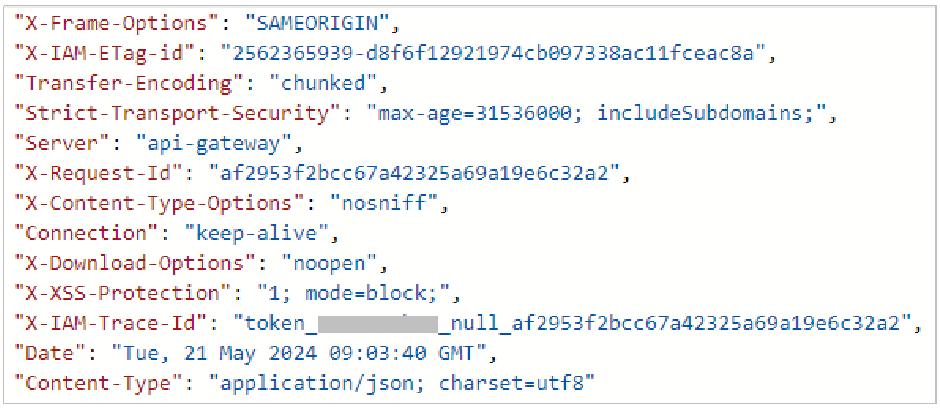
Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 3-2 shows the response header fields for the API used to **create an IAM user**. The **X-Subject-Token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-2 Header fields of the response to the request for creating an IAM user



```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fceac8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_[redacted]_null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to **create an IAM user**.

```
{
    "user": {
        "id": "c131886aec...",
        "name": "IAMUser",
        "description": "IAM User Description",
        "areacode": "",
        "phone": "",
        "email": "***@***.com",
        "status": null,
        "enabled": true,
        "pwd_status": false,
        "access_mode": "default",
        "is_domain_owner": false,
        "xuser_id": "",
        "xuser_type": "",
        "password_expires_at": null,
        "create_time": "2024-05-21T09:03:41.000000",
        "domain_id": "d78cbac1..........",
        "xdomain_id": "30086000........",
        "xdomain_type": "",
        "default_project_id": null
    }
}
```

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

```
{
    "error_msg": "The request message format is invalid.",
    "error_code": "IMG.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Getting Started

Scenarios

This topic describes how to publish a message to a topic by calling APIs.

For details about how to call APIs, see **3 Calling APIs**.

NOTE

The token obtained from IAM is valid for only 24 hours. If you want to use one token for authentication, you can cache it to avoid frequently calling the IAM API.

Involved APIs

If you use a token for authentication, you must obtain the token and add **X-Auth-Token** to the request header of the API when making a call.

- IAM API to obtain a token
- SMN API to create a topic
- SMN API to add subscriptions
- SMN API to publish messages

Procedure

1. Obtain the token by referring to **3.2 Authentication**.
2. Send a request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics** to create a topic.
   a. Add **X-Auth-Token** to the request header.
   b. Specify the following parameters in the request body:
      ```
      {
         "name": "test_topic_v2", //(Mandatory) Topic name
         "display_name": "test"  //(Optional) Display name
      }
      ```
      If the request is successful, a request ID (**request_id**) and topic URN (**topic_urn**) are returned.

      If the request fails, an error code and error message are returned. For details, see **Error Codes**
3. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/subscriptions** to add subscriptions.

Set **topic_urn** based on the response in 2.b.

a. Add **X-Auth-Token** to the request header.

b. Specify the following parameters in the request body:

```
{
   "protocol": "email", //(Mandatory) Subscription protocol
   "endpoint": "xxx@xx.com", //(Mandatory) Email address
   "remark": "Subscription" //(Optional) Description
}
```

If the request is successful, a request ID (**request_id**) and subscription URN (**subscription_urn**) are returned.

If the request fails, an error code and error message are returned. For details, see **Error Codes**.

NOTE

You can add multiple subscriptions to a topic. For details, see **Adding a Subscription**.

c. The subscription endpoint receives a confirmation message, and you must confirm the subscription in the email.

4. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/publish**.

a. Add **X-Auth-Token** to the request header.

b. Specify the following parameters in the request body:

```
{
  "subject":"test message v2", //(Optional) Message subject, which is displayed in email
messages
  "message": "Message test message v2." //(Mandatory) Message to be published
}
```

If the request is successful, a request ID (**request_id**) and message ID (**message_id**) are returned.

If the request fails, an error code and error message are returned. For details, see **A.3 Error Codes**.

NOTE

Only confirmed subscription endpoints can receive the message you publish.

5 APIs

5.1 Topics

5.1.1 Querying Topics

Function

This API is used to query topics by page. The topics are sorted by creation time in descending order. You can specify offset and limit. If no topic has been created, an empty list is returned.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics

Table 5-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Table 5-2 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. If the offset is an integer greater than 0 but less than the number of resources, all resources in excess of this offset will be queried. The default offset is 0.
limit	No	Integer	Specifies the maximum number of records in each query. The number ranges from 1 to 100 and is generally 10, 20, or 50. Specifies the number of resources returned on each page. The default number is 100.
enterprise_project_id	No	String	Specifies the enterprise project ID.
name	No	String	Specifies the name of the topic to be searched for, which must be fully matched.
fuzzy_name	No	String	Specifies the name of the topic to be searched. Fuzzy matching is used.
topic_id	No	String	Specifies the topic ID for exact search.
fuzzy_display_name	No	String	Specifies the display name of the topic to be searched. Fuzzy matching is used. The display name cannot exceed 192 bytes.

Request Parameters

Table 5-3 Request header parameters

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

Response Parameters

Status code: 200

Table 5-4 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
topic_count	Integer	Specifies the number of returned topics. No matter what offset and limit values you have set in the request, this API always returns the total number of topics in your account.
topics	Array of **ListTopicsItem** objects	Specifies the topic structure array.

Table 5-5 ListTopicsItem

Parameter	Type	Description
topic_urn	String	Specifies the resource identifier of the topic, which is unique.
name	String	Specifies the topic name.
display_name	String	Specifies the topic display name, which is presented as the name of the email sender in email messages.

Parameter	Type	Description
push_policy	Integer	Specifies the message push policy. 0 indicates that the message sending fails and the message is cached in the queue. 1 indicates that the failed message is discarded.
enterprise_project_id	String	Specifies the enterprise project ID.
topic_id	String	Specifies the topic ID.

Status code: 400

Table 5-6 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-7 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-8 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Status code: 500

Table 5-9 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying topics

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics?offset=0&limit=100

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "topic_count" : 1,
  "topics" : [ {
    "topic_urn" : "urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:test_topic_v2",
    "display_name" : "testtest",
    "name" : "test_topic_v1",
    "push_policy" : 0,
    "enterprise_project_id" : "0",
    "topic_id" : "84698185946d44d08431aa441d8e2cf2"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized

Status Code	Description
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.2 Creating a Topic

Function

This API is used to create a topic. Each user can create up to 3,000 topics. In high-concurrent scenarios, a user may create a few topics more than 3,000.

The API is idempotent. It returns a successful result after creating a topic. If a topic of the same name already exists, the status code is 200. Otherwise, the status code is 201.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/topics

Table 5-10 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Request Parameters

Table 5-11 Request header parameters

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

Table 5-12 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the topic name. Enter 1 to 255 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The topic name must start with a letter or digit.
display_name	Yes	String	Specifies the topic display name, which is presented as the name of the email sender in email messages. The display name cannot exceed 192 bytes. display_name is left blank by default.
enterprise_project_id	No	String	Specifies the enterprise project ID. It is required only when the enterprise project function is enabled.

Response Parameters

Status code: 201

Table 5-13 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
topic_urn	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see [Querying Topics].

Status code: 400

Table 5-14 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-15 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-16 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-17 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Creating a topic with the name of **test_topic_v2** and display name of **testtest**

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics

{
  "name" : "test_topic_v2",
  "display_name" : "testtest"
}
```

Example Responses

Status code: 201

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "topic_urn" : "urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a topic with the name of **test_topic_v2** and display name of **testtest**

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class CreateTopicSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateTopicRequest request = new CreateTopicRequest();
        CreateTopicRequestBody body = new CreateTopicRequestBody();
        body.withDisplayName("testtest");
        body.withName("test_topic_v2");
        request.withBody(body);
        try {
            CreateTopicResponse response = client.createTopic(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

Creating a topic with the name of **test_topic_v2** and display name of **testtest**

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateTopicRequest()
        request.body = CreateTopicRequestBody(
            display_name="testtest",
            name="test_topic_v2"
        )
        response = client.create_topic(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

Creating a topic with the name of **test_topic_v2** and display name of **testtest**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTopicRequest{}
    request.Body = &model.CreateTopicRequestBody{
        DisplayName: "testtest",
        Name: "test_topic_v2",
    }
    response, err := client.CreateTopic(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
201	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.3 Querying Details About a Topic

Function

This API is used to query details about a topic.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics/{topic_urn}

Table 5-18 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-19 Request header parameters

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

Response Parameters

Status code: 200

Table 5-20 Response body parameters

Parameter	Type	Description
update_time	String	Specifies when the topic was updated. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
push_policy	Integer	Specifies the message push policy. 0 indicates that the message sending fails and the message is cached in the queue. 1 indicates that the failed message is discarded.
create_time	String	Specifies when the topic is created. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
name	String	Specifies the topic name.
topic_urn	String	It is the unique identifier of a topic. To obtain the resource identifier, see **Querying Topics**.
display_name	String	Specifies the topic display name, which is presented as the name of the email sender in email messages.
request_id	String	Specifies the request ID, which is unique.
enterprise_project_id	String	Specifies the enterprise project ID.
topic_id	String	Specifies the topic ID.

Status code: 400

Table 5-21 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-22 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-23 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-24 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Example Requests

Querying details about a topic

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:test_create_topic_v2

null
```

Example Responses

Status code: 200

OK

```
{
  "update_time" : "2016-08-01T02:16:38Z",
  "push_policy" : 0,
  "create_time" : "2016-08-01T02:16:38Z",
  "name" : "test_create_topic_v2",
  "topic_urn" : "urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:test_create_topic_v2",
  "display_name" : "test create topic v2",
  "request_id" : "6837531fd3f54550927b930180a706bf",
  "enterprise_project_id" : "0",
  "topic_id" : "84698185946d44d08431aa441d8e2cf2"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.4 Updating a Topic

Function

This API is used to update the topic display name.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/topics/{topic_urn}

Table 5-25 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-26 Request header parameters

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

Table 5-27 Request body parameters

Parameter	Mandatory	Type	Description
display_name	Yes	String	Specifies the topic display name, which is presented as the name of the email sender in email messages. The display name cannot exceed 192 bytes.

Response Parameters

Status code: 200

Table 5-28 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-29 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-30 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-31 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-32 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Modifying the display name of the topic named **test_topic_v2** to **testtest222**

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2

{
  "display_name" : "testtest222"
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying the display name of the topic named **test_topic_v2** to **testtest222**

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class UpdateTopicSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateTopicRequest request = new UpdateTopicRequest();
        request.withTopicUrn("{topic_urn}");
        UpdateTopicRequestBody body = new UpdateTopicRequestBody();
        body.withDisplayName("testtest222");
        request.withBody(body);
        try {
            UpdateTopicResponse response = client.updateTopic(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

Modifying the display name of the topic named **test_topic_v2** to **testtest222**

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateTopicRequest()
        request.topic_urn = "{topic_urn}"
        request.body = UpdateTopicRequestBody(
            display_name="testtest222"
        )
        response = client.update_topic(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

Modifying the display name of the topic named **test_topic_v2** to **testtest222**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTopicRequest{}
    request.TopicUrn = "{topic_urn}"
    request.Body = &model.UpdateTopicRequestBody{
        DisplayName: "testtest222",
    }
    response, err := client.UpdateTopic(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.5 Deleting a Topic

Function

This API is used to delete a topic.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/topics/{topic_urn}

Table 5-33 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-34 Request header parameters

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

Response Parameters

Status code: 200

Table 5-35 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-36 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-37 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-38 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-39 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a topic

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request

Status Code	Description
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.6 Querying a Topic Policy

Function

This API is used to query a topic policy.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics/{topic_urn}/attributes

Table 5-40 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Table 5-41 Query Parameters

Parameter	Mandatory	Type	Description
name	No	String	Specifies the topic policy name.Only specified policy names are supported. For details, see **Topic Attribute List**.

Request Parameters

Table 5-42 Request header parameters

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

Response Parameters

Status code: 200

Table 5-43 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
attributes	**TopicAttribute** object	Specifies the attribute. Its value can be: -access_policy: specifies the access policy of the topic. -introduction: specifies the introduction to the topic.

Table 5-44 TopicAttribute

Parameter	Type	Description
access_policy	String	Specifies the access policy of a topic.
introduction	String	Specifies the introduction to a topic.

Status code: 400

Table 5-45 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-46 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-47 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-48 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Example Requests

Querying a topic policy

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:test_create_topic_v2/attributes?name=access_policy

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6837531fd3f54550927b930180a706bf",
  "attributes" : {
    "access_policy" : "{\"Version\":\"2016-09-07\",\"Id\":\"__default_policy_ID\",\"Statement\":[{\"Sid
\":\"__user_pub_0\",\"Effect\":\"Allow\",\"Principal\":[{\"CSP\":
[\"urn:csp:iam::93dc1b4697ac493d9b7d089569f86b32:root\"]}],\"Action\":[\"SMN:Publish
\",\"SMN:QueryTopicDetail\"],\"Resource\":\"urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:aaa\"},
{\"Sid\":\"__org_path_pub_0\",\"Effect\":\"Allow\",\"Principal\":{\"OrgPath\":[\"o-
bf966fe82ebb4d35d68b791729228788/r-001ebf32880a13eabfc8e1c37eee3ae9/
ou-0dbfffe92fd92ddb35feff9b4079459c\"]},\"Action\":[\"SMN:Publish\",\"SMN:QueryTopicDetail
\"],\"Resource\":\"urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:aaa\"},{\"Sid
\":\"__service_pub_0\",\"Effect\":\"Allow\",\"Principal\":{\"Service\":[\"obs\"]},\"Action\":[\"SMN:Publish
\",\"SMN:QueryTopicDetail\"],\"Resource\":\"urn:smn:regionId:8bad8a40e0f7462f8c1676e3f93a8183:aaa
\"}]}"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.7 Deleting All Topic Policies

Function

This API is used to delete all topic policies.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/topics/{topic_urn}/attributes

Table 5-49 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-50 Request header parameters

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

Response Parameters

Status code: 200

Table 5-51 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-52 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-53 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-54 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-55 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting all topic policies

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/attributes

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6837531fd3f54550927b930180a706bf"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.8 Updating a Topic Policy

Function

This API is used to update a topic policy.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/topics/{topic_urn}/attributes/{name}

Table 5-56 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.
name	Yes	String	Specifies the topic policy name.Only specified policy names are supported. For details, see **Topic Attribute List**.

Request Parameters

Table 5-57 Request header parameters

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

Table 5-58 Request body parameters

Parameter	Mandatory	Type	Description
value	Yes	String	1. If you set name to access_policy, value is the topic attribute, and the maximum length of value is 30 KB. 2. If you set name to introduction, value is the topic introduction, and the maximum length of value is 120 bytes.

Response Parameters

Status code: 200

Table 5-59 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-60 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-61 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-62 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-63 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

- Updating a topic policy

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/attributes/access_policy

{
  "value" : "{\"Version\":\"2016-09-07\",\"Id\":\"__default_policy_ID\",\"Statement\":[{\"Sid
\":\"__user_pub_0\",\"Effect\":\"Allow\",\"Principal\":{\"CSP\":[\"urn:csp:iam::{domainID}:root
\"]},\"Action\":[\"SMN:Publish\",\"SMN:QueryTopicDetail\"],\"Resource\":\"{topic_urn}\"},{\"Sid
\":\"__org_path_pub_0\",\"Effect\":\"Allow\",\"Principal\":{\"OrgPath\":[\"o-
bf966fe82ebb4d35d68b791729228788/r-001ebf32880a13eabfc8e1c37eee3ae9/
ou-0dbfffe92fd92ddb35feff9b4079459c\"]},\"Action\":[\"SMN:Publish\",\"SMN:QueryTopicDetail
\"],\"Resource\":\"{topic_urn}\"},{\"Sid\":\"__service_pub_0\",\"Effect\":\"Allow\",\"Principal\":{\"Service
\":[\"obs\"]},\"Action\":[\"SMN:Publish\",\"SMN:QueryTopicDetail\"],\"Resource\":\"{topic_urn}\"}]}"
}
```

- Updating a topic policy

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/attributes/introduction

{
  "value" : ""
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6837531fd3f54550927b930180a706bf"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Updating a topic policy

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class UpdateTopicAttributeSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateTopicAttributeRequest request = new UpdateTopicAttributeRequest();
        request.withTopicUrn("{topic_urn}");
        request.withName("{name}");
        UpdateTopicAttributeRequestBody body = new UpdateTopicAttributeRequestBody();
        body.withValue("{"Version":"2016-09-07","Id":"__default_policy_ID","Statement":[{"Sid":"__user_pub_0","Effect":"Allow","Principal":{"CSP":["urn:csp:iam::{domainID}:root"]},"Action":["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"},{"Sid":"__org_path_pub_0","Effect":"Allow","Principal":{"OrgPath":["o-bf966fe82ebb4d35d68b791729228788/r-001ebf32880a13eabfc8e1c37eee3ae9/ou-0dbfffe92fd92ddb35feff9b4079459c"]},"Action":["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"},{"Sid":"__service_pub_0","Effect":"Allow","Principal":{"Service":["obs"]},"Action":["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"}]}");
        request.withBody(body);
        try {
            UpdateTopicAttributeResponse response = client.updateTopicAttribute(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());
        }
    }
}
```

- Updating a topic policy

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class UpdateTopicAttributeSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateTopicAttributeRequest request = new UpdateTopicAttributeRequest();
        request.withTopicUrn("{topic_urn}");
        request.withName("{name}");
        UpdateTopicAttributeRequestBody body = new UpdateTopicAttributeRequestBody();
        body.withValue("");
        request.withBody(body);
        try {
            UpdateTopicAttributeResponse response = client.updateTopicAttribute(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 a topic policy

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateTopicAttributeRequest()
        request.topic_urn = "{topic_urn}"
        request.name = "{name}"
        request.body = UpdateTopicAttributeRequestBody(
            value="{"Version":"2016-09-07","Id":"__default_policy_ID","Statement":
[{"Sid":"__user_pub_0","Effect":"Allow","Principal":{"CSP":["urn:csp:iam::{domainID}:root"]},"Action":
["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"},
{"Sid":"__org_path_pub_0","Effect":"Allow","Principal":{"OrgPath":["o-
bf966fe82ebb4d35d68b791729228788/r-001ebf32880a13eabfc8e1c37eee3ae9/
ou-0dbfffe92fd92ddb35feff9b4079459c"]},"Action":
["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"},
{"Sid":"__service_pub_0","Effect":"Allow","Principal":{"Service":["obs"]},"Action":
["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"}]}"
        )
        response = client.update_topic_attribute(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Updating a topic policy

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateTopicAttributeRequest()
        request.topic_urn = "{topic_urn}"
        request.name = "{name}"
        request.body = UpdateTopicAttributeRequestBody(
            value=""
        )
        response = client.update_topic_attribute(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 a topic policy

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTopicAttributeRequest{}
    request.TopicUrn = "{topic_urn}"
    request.Name = "{name}"
    request.Body = &model.UpdateTopicAttributeRequestBody{
        Value: "{"Version":"2016-09-07","Id":"__default_policy_ID","Statement":
```

```
[{"Sid":"__user_pub_0","Effect":"Allow","Principal":{"CSP":["urn:csp:iam::{domainID}:root"]},"Action":
["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"},
{"Sid":"__org_path_pub_0","Effect":"Allow","Principal":{"OrgPath":["o-
bf966fe82ebb4d35d68b791729228788/r-001ebf32880a13eabfc8e1c37eee3ae9/
ou-0dbfffe92fd92ddb35feff9b4079459c"]},"Action":
["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"},
{"Sid":"__service_pub_0","Effect":"Allow","Principal":{"Service":["obs"]},"Action":
["SMN:Publish","SMN:QueryTopicDetail"],"Resource":"{topic_urn}"}]}",
    }
    response, err := client.UpdateTopicAttribute(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Updating a topic policy

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTopicAttributeRequest{}
    request.TopicUrn = "{topic_urn}"
    request.Name = "{name}"
    request.Body = &model.UpdateTopicAttributeRequestBody{
        Value: "",
    }
    response, err := client.UpdateTopicAttribute(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.1.9 Deleting a Specified Topic Policy

Function

This API is used to delete a specified topic policy.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/topics/{topic_urn}/attributes/{name}

Table 5-64 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the topic policy name.Only specified policy names are supported. For details, see **Topic Attribute List**.

Request Parameters

Table 5-65 Request header parameters

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

Response Parameters

Status code: 200

Table 5-66 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-67 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-68 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-69 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-70 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a specified topic policy

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/attributes/access_policy

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6837531fd3f54550927b930180a706bf"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.2 Subscriptions

5.2.1 Querying Subscriptions

Function

This API is used to query the list of all subscriptions by page. The list is sorted by subscription adding time in ascending order. You can specify offset and limit. If no subscription has been added to the topic, an empty list is returned.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/subscriptions

Table 5-71 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Table 5-72 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. If the offset is an integer greater than 0 but less than the number of resources, all resources in excess of this offset will be queried. The default offset is 0.
limit	No	Integer	Specifies the maximum number of records in each query. The number ranges from 1 to 100 and is generally 10, 20, or 50. Specifies the number of resources returned on each page. The default number is 100.
protocol	No	String	Specifies the protocol name. The enumerated values are http, https, sms, email, functionstage.
status	No	Integer	Specifies the subscription status. 0: The subscription has not been confirmed. 1: The subscription has been confirmed. 2: Confirmation is not required. 3: The subscription was canceled. 4: The subscription was deleted.
endpoint	No	String	Specifies the subscription endpoint.
fuzzy_remark	No	String	Specifies the subscription remark for fuzzy match. The value can contain up to 128 bytes.

Request Parameters

Table 5-73 Request header parameters

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

Response Parameters

Status code: 200

Table 5-74 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
subscription_count	Integer	Specifies the number of subscriptions.
subscriptions	Array of **ListSubscriptionsItem** objects	Specifies the subscription structure.

Table 5-75 ListSubscriptionsItem

Parameter	Type	Description
topic_urn	String	Specifies the resource identifier of the topic, which is unique.

Parameter	Type	Description
protocol	String	Specifies the subscription protocol. (Different protocols indicate different types of endpoints to receive messages.) The following protocols are supported: default email: The endpoints are email addresses. sms: The endpoints are phone numbers. functionstage: FunctionGraph (function) transmission protocol. The endpoints are functions. http and https: The endpoints are URLs.
subscription_urn	String	Specifies the resource identifier of a subscription, which is unique.
owner	String	Specifies the project ID of the topic creator.
endpoint	String	Specifies the message receiving endpoint.
remark	String	Specifies the remarks.
status	Integer	0 indicates that the subscription is not confirmed. 1 indicates that the subscription is confirmed. 3 indicates that the subscription is canceled.
filter_polices	Array of **SubscriptionsFilterPolicy** objects	Specifies the filter policy.

Table 5-76 SubscriptionsFilterPolicy

Parameter	Type	Description
name	String	Specifies the filter policy name. **name** can contain 1 to 32 characters, including lowercase letters, digits, and underscores (*). **name** cannot start or end with an underscore, nor contain consecutive underscores. **name** cannot start with **smn**.*

Parameter	Type	Description
string_equals	Array of strings	Specifies the string array for exact match. An array can contain 1 to 10 strings. The array content must be unique. The string cannot be **null** or an empty string "". A string can contain 1 to 32 characters, including letters, digits, and underscores (_).

Status code: 400

Table 5-77 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-78 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-79 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-80 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying subscriptions

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/subscriptions?offset=0&limit=2

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "subscription_count" : 2,
  "subscriptions" : [ {
    "topic_urn" : "urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1",
    "protocol" : "sms",
    "subscription_urn" :
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
    "owner" : "762bdb3251034f268af0e395c53ea09b",
    "endpoint" : "xxxxxxxxxxx",
    "remark" : "",
    "status" : 0,
    "filter_polices" : [ {
      "name" : "alarm",
      "string_equals" : [ "os", "process" ]
    }, {
      "name" : "service",
      "string_equals" : [ "api", "db" ]
    } ]
  }, {
    "topic_urn" : "urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1",
    "protocol" : "email",
    "subscription_urn" :
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:a2d52a9f5c3b47f48c3fafb177a58796",
    "owner" : "762bdb3251034f268af0e395c53ea09b",
    "endpoint" : "xx@example.com",
    "remark" : "",
    "status" : 0,
    "filter_polices" : [ {
      "name" : "alarm",
      "string_equals" : [ "os", "process" ]
    }, {
      "name" : "service",
      "string_equals" : [ "api", "db" ]
    } ]
```

```
 }]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.2.2 Querying Subscriptions of a Specified Topic

Function

This API is used to query subscriptions to a specified topic by page. The list is sorted by subscription adding time in ascending order. You can specify offset and limit. If no subscription has been added to the topic, an empty list is returned.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics/{topic_urn}/subscriptions

Table 5-81 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Table 5-82 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. If the offset is an integer greater than 0 but less than the number of resources, all resources in excess of this offset will be queried. The default offset is 0.
limit	No	Integer	Specifies the maximum number of records in each query. The number ranges from 1 to 100 and is generally 10, 20, or 50. Specifies the number of resources returned on each page. The default number is 100.
fuzzy_remark	No	String	Specifies the subscription remark for fuzzy match. The value can contain up to 128 bytes.

Request Parameters

Table 5-83 Request header parameters

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

Response Parameters

Status code: 200

Table 5-84 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
subscription_count	Integer	Specifies the number of subscriptions.
subscriptions	Array of **ListSubscriptionsItem** objects	Specifies the subscription structure.

Table 5-85 ListSubscriptionsItem

Parameter	Type	Description
topic_urn	String	Specifies the resource identifier of the topic, which is unique.
protocol	String	Specifies the subscription protocol. (Different protocols indicate different types of endpoints to receive messages.) The following protocols are supported: default email: The endpoints are email addresses. sms: The endpoints are phone numbers. functionstage: FunctionGraph (function) transmission protocol. The endpoints are functions. http and https: The endpoints are URLs.
subscription_urn	String	Specifies the resource identifier of a subscription, which is unique.
owner	String	Specifies the project ID of the topic creator.
endpoint	String	Specifies the message receiving endpoint.
remark	String	Specifies the remarks.
status	Integer	0 indicates that the subscription is not confirmed. 1 indicates that the subscription is confirmed. 3 indicates that the subscription is canceled.
filter_polices	Array of **SubscriptionsFilterPolicy** objects	Specifies the filter policy.

Table 5-86 SubscriptionsFilterPolicy

Parameter	Type	Description
name	String	Specifies the filter policy name. **name** can contain 1 to 32 characters, including lowercase letters, digits, and underscores (*). **name** cannot start or end with an underscore, nor contain consecutive underscores. **name** cannot start with **smn**.*
string_equals	Array of strings	Specifies the string array for exact match. An array can contain 1 to 10 strings. The array content must be unique. The string cannot be **null** or an empty string "". A string can contain 1 to 32 characters, including letters, digits, and underscores (_).

Status code: 400

Table 5-87 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-88 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-89 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-90 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying subscriptions of the topic named **test_topic_v2**

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1/subscriptions?offset=0&limit=100

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "subscription_count" : 2,
  "subscriptions" : [ {
    "topic_urn" : "urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1",
    "protocol" : "sms",
    "subscription_urn" :
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
    "owner" : "762bdb3251034f268af0e395c53ea09b",
    "endpoint" : "xxxxxxxxxxx",
    "remark" : "",
    "status" : 0,
    "filter_polices" : [ {
      "name" : "alarm",
      "string_equals" : [ "os", "process" ]
    }, {
      "name" : "service",
      "string_equals" : [ "api", "db" ]
    } ]
  }, {
    "topic_urn" : "urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1",
    "protocol" : "email",
    "subscription_urn" :
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:a2d52a9f5c3b47f48c3fafb177a58796",
    "owner" : "762bdb3251034f268af0e395c53ea09b",
    "endpoint" : "xx@example.com",
    "remark" : "",
    "status" : 0,
    "filter_polices" : [ {
      "name" : "alarm",
      "string_equals" : [ "os", "process" ]
    }, {
      "name" : "service",
      "string_equals" : [ "api", "db" ]
    } ]
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.2.3 Adding a Subscription

Function

This API is used to add a subscription to a specified topic. If the status of the subscription is unconfirmed, a confirmation message is sent to the subscriber. After confirming the subscription, the subscriber can receive notification messages published to the topic. By default, 10,000 subscriptions can be added to a topic. However, in rare high-concurrency scenarios, extra subscriptions may be added successfully. The API is idempotent. If the added subscription already exists, a successful result and status code 200 are returned. Otherwise, the status code is 201.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/topics/{topic_urn}/subscriptions

Table 5-91 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-92 Request header parameters

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

Table 5-93 Request body parameters

Parameter	Mandatory	Type	Description
protocol	Yes	String	Subscription protocol. (Different protocols indicate different types of endpoints to receive messages.) The following protocols are supported: **email**: The endpoints are email addresses. **sms**: SMS transmission protocol. The endpoints are phone numbers. **functionstage**: FunctionGraph(function) transmission protocol. The endpoints are functions. **functiongraph**: FunctionGraph(workflow) transmission protocol. The endpoints are FunctionGraph workflows orchestrated by the FunctionGraph function. **http** and **https**: The endpoints are URLs. **callnotify**: Voice notification protocol. The endpoints are phone numbers. **wechat**: WeCom group chatbot transmission protocol. **dingding**: DingTalk group chatbot transmission protocol. **feishu**: Lark group chatbot transmission protocol. **welink**: WeLink group chatbot transmission protocol. **dingTalkBot**: DingTalk transmission protocol.

Parameter	Mandatory	Type	Description
endpoint	Yes	String	Note: If you use the HTTP protocol, start the endpoint with **http://**. If you use the HTTPS protocol, start the endpoint with **https://**. If you use the email protocol, enter an email address. If you use the SMS protocol, enter a phone number. If you use the FunctionGraph(function) protocol, enter a function. If you use the FunctionGraph (workflow) protocol, enter a workflow. If you use the voice notification protocol, enter a phone number. If you use the DingTalk chatbot protocol, enter the webhook URL of a custom DingTalk group chatbot or **openConversationId** of a group to which an enterprise DingTalk chatbot is added. When you add **openConversationId**, do not start the endpoint with **http://** or **https://**. If you use the **wechat** protocol, enter the webhook URL of a WeCom group chatbot. If you use the**feishu** protocol, enter the webhook URL of a Lark group chatbot. If you use the **welink** protocol, enter a WeLink group number. If you use the **dingTalkBot** protocol, enter the user ID of a DingTalk enterprise user.

Parameter	Mandatory	Type	Description
remark	No	String	Specifies the remarks. The remarks must be UTF-8-coded and can contain up to 128 bytes.
extension	No	**SubscriptionExtension** object	Specifies the extension information.
subscriptions	No	Array of **BatchAddSubscriptionsRequestBody** objects	This parameter is mandatory when subscriptions need to be created in batches. SMN allows you to create a maximum of 50 subscriptions at a time.

Table 5-94 BatchAddSubscriptionsRequestBody

Parameter	Mandatory	Type	Description
protocol	Yes	String	Subscription protocol. (Different protocols indicate different types of endpoints to receive messages.) The following protocols are supported: **email**: The endpoints are email addresses. **sms**: SMS transmission protocol. The endpoints are phone numbers. **functionstage**: FunctionGraph(function) transmission protocol. The endpoints are functions. **functiongraph**: FunctionGraph(workflow) transmission protocol. The endpoints are FunctionGraph workflows orchestrated by the FunctionGraph function. **http** and **https**: The endpoints are URLs. **callnotify**: Voice notification protocol. The endpoints are phone numbers. **wechat**: WeCom group chatbot transmission protocol. **dingding**: DingTalk group chatbot transmission protocol. **feishu**: Lark group chatbot transmission protocol. **welink**: WeLink group chatbot transmission protocol.

Parameter	Mandatory	Type	Description
endpoint	Yes	String	Note: If you use the HTTP protocol, start the endpoint with **http://**. If you use the HTTPS protocol, start the endpoint with **https://**. If you use the email protocol, enter an email address. If you use the SMS protocol, enter a phone number. If you use the FunctionGraph (function) protocol, enter a function. If you use the FunctionGraph (workflow) protocol, enter a workflow. If you use the voice notification protocol, enter a phone number. If you use the DingTalk chatbot protocol, enter the webhook URL of a custom DingTalk group chatbot or **openConversationId** of a group to which an enterprise DingTalk chatbot is added. When you add **openConversationId**, do not start the endpoint with **http://** or **https://**. If you use the **wechat** protocol, enter the webhook URL of a WeCom group chatbot. If you use the**feishu** protocol, enter the webhook URL of a Lark group chatbot. If you use the **welink** protocol, enter a WeLink group number.
remark	No	String	Remarks. The remarks must be UTF-8-coded and can contain up to 128 bytes.
extension	No	**SubscriptionExtension** object	Extended subscription information.

Table 5-95 SubscriptionExtension

Parameter	Mandatory	Type	Description
client_id	No	String	Specifies the application ID. This field is mandatory when you set **protocol** to **welink**. Obtain the value from WeLink.
client_secret	No	String	Specifies the application secret. This field is mandatory when you set **protocol** to **welink**. Obtain the value from WeLink.
keyword	No	String	Specifies the keyword. When you set **protocol** to **feishu**, you must set either **keyword** or **sign_secret**. When you add a keyword for security verification on the Lark chatbot or a custom DingTalk chatbot, **keyword** must be one of the keywords you entered for the Lark chatbot or the custom DingTalk chatbot.
sign_secret	No	String	Specifies the key. When you set **protocol** to **feishu**, you must set either this parameter or **keyword**. The value of this parameter must be the same as that you customized for the Lark chatbot or the custom DingTalk chatbot. For example, if you set only **key** for the Lark chatbot, enter the value of **key**. If you set only **keyword** for the Lark chatbot, skip this field.

Parameter	Mandatory	Type	Description
header	No	Map<String,String>	This is an HTTP header field, which can be customized within the field range. The field content exists in the form of key/value pairs. When a topic is used to send messages, confirmed subscription messages carry the user-defined HTTP header. Header for subscriptions over HTTP or HTTPS. The header must meet the following requirements: **key** can contain letters, digits, and hyphens (-). **key** cannot end with a hyphen (-) nor contain consecutive hyphens (-). You can specify up to 10 key/value pairs. **key** must start with **x-**but not **x-smn**. Examples: **x-abc-cba** or **x-abc** The total length of all key/value pairs cannot exceed 1,024 characters. **key** is case insensitive. **key** must be unique. **value** must be an ASCII code. Unicode characters are not supported. Spaces are allowed.
app_key	No	String	The **appKey** field of DingTalk can contain a maximum of 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. This parameter is mandatory when the protocol is set to **dingTalkBot**.
app_secret	No	String	The **appSecret** field of DingTalk can contain a maximum of 128 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. This parameter is mandatory when the protocol is set to **dingTalkBot**.

Parameter	Mandatory	Type	Description
robot_code	No	String	The **robotCode** field of DingTalk can contain a maximum of 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. Generally, the value is the same as that of **appKey**. This parameter is mandatory when the protocol is set to **dingTalkBot**.

Response Parameters

Status code: 201

Table 5-96 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
subscription_urn	String	Specifies the resource identifier of a subscription, which is unique.

Status code: 400

Table 5-97 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-98 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-99 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-100 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Adding a subscription to the topic named **test_topic_v1**

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1/subscriptions

{
  "protocol" : "email",
  "endpoint" : "xxx@example.com",
  "remark" : "O&M"
}
```

Example Responses

Status code: 201

OK

```
[ {
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "subscription_urn" :
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
"
} ]
```

SDK Sample Code

The SDK sample code is as follows.

Java

Adding a subscription to the topic named **test_topic_v1**

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class AddSubscriptionSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        AddSubscriptionRequest request = new AddSubscriptionRequest();
        request.withTopicUrn("{topic_urn}");
        AddSubscriptionRequestBody body = new AddSubscriptionRequestBody();
        body.withRemark("O&M");
        body.withEndpoint("xxx@example.com");
        body.withProtocol("email");
        request.withBody(body);
        try {
            AddSubscriptionResponse response = client.addSubscription(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

Adding a subscription to the topic named **test_topic_v1**

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddSubscriptionRequest()
        request.topic_urn = "{topic_urn}"
        request.body = AddSubscriptionRequestBody(
            remark="O&M",
            endpoint="xxx@example.com",
            protocol="email"
        )
        response = client.add_subscription(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

Adding a subscription to the topic named **test_topic_v1**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddSubscriptionRequest{}
    request.TopicUrn = "{topic_urn}"
    remarkAddSubscriptionRequestBody:= "O&M"
    request.Body = &model.AddSubscriptionRequestBody{
        Remark: &remarkAddSubscriptionRequestBody,
        Endpoint: "xxx@example.com",
        Protocol: "email",
    }
    response, err := client.AddSubscription(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
201	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.2.4 Deleting a Subscription

Function

This API is used to delete a specified subscription.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/subscriptions/{subscription_urn}

Table 5-101 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
subscription_urn	Yes	String	Specifies the resource identifier of the subscriber, which is unique. To obtain the resource identifier, see **Query Subscriptions**.

Request Parameters

Table 5-102 Request header parameters

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

Response Parameters

Status code: 200

Table 5-103 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-104 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-105 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-106 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-107 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a subscription with the URN **urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837**

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/subscriptions/
urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.2.5 Updating a Subscriber

Function

This API is used to update the subscriber remarks.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/topics/{topic_urn}/subscriptions/{subscription_urn}

Table 5-108 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.
subscription_urn	Yes	String	Specifies the resource identifier of the subscriber, which is unique. To obtain the resource identifier, see **Query Subscriptions**.

Request Parameters

Table 5-109 Request header parameters

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

Table 5-110 Request body parameters

Parameter	Mandatory	Type	Description
remark	Yes	String	Specifies the subscriber remarks. The subscriber remarks can contain up to 128 bytes.

Response Parameters

Status code: 200

Table 5-111 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique request ID.
subscription_urn	String	Specifies the resource identifier of a subscription, which is unique.

Status code: 400

Table 5-112 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-113 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-114 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-115 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Updating a subscriber

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1/subscriptions/
urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837

{
  "remark" : "test remark"
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "fdbabe38ead6482b8574f82a3d1168e9",
  "subscription_urn" :
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating a subscriber

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class UpdateSubscriptionSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateSubscriptionRequest request = new UpdateSubscriptionRequest();
        request.withTopicUrn("{topic_urn}");
        request.withSubscriptionUrn("{subscription_urn}");
        UpdateSubscriptionRequestBody body = new UpdateSubscriptionRequestBody();
        body.withRemark("test remark");
        request.withBody(body);
        try {
            UpdateSubscriptionResponse response = client.updateSubscription(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 a subscriber

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateSubscriptionRequest()
        request.topic_urn = "{topic_urn}"
        request.subscription_urn = "{subscription_urn}"
        request.body = UpdateSubscriptionRequestBody(
            remark="test remark"
        )
        response = client.update_subscription(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 a subscriber

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSubscriptionRequest{}
    request.TopicUrn = "{topic_urn}"
    request.SubscriptionUrn = "{subscription_urn}"
    request.Body = &model.UpdateSubscriptionRequestBody{
        Remark: "test remark",
    }
    response, err := client.UpdateSubscription(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.3 Templates

5.3.1 Querying Message Templates

Function

This API is used to query the template list by page. The list is sorted by template creation time in ascending order. You can specify offset and limit. If no template has been created, an empty list is returned. The parameters message_template_name and protocol are added.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/message_template

Table 5-116 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Table 5-117 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. If the offset is an integer greater than 0 but less than the number of resources, all resources in excess of this offset will be queried. The default offset is 0.
limit	No	Integer	The number ranges from 1 to 100 and is generally 10, 20, or 50. Specifies the number of resources returned on each page. The default number is 100.
message_template_name	No	String	Specifies the template name. Enter 1 to 64 characters, and start with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Mandatory	Type	Description
protocol	No	String	Specifies the protocol supported by the template. The following protocols are supported: default email sms functionstage http and https

Request Parameters

Table 5-118 Request header parameters

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

Response Parameters

Status code: 200

Table 5-119 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
message_template_count	Integer	Specifies the number of returned templates.
message_templates	Array of **MessageTemplate** objects	Specifies the Message_template structure array.

Table 5-120 MessageTemplate

Parameter	Type	Description
message_template_id	String	Specifies the template ID.
message_template_name	String	Specifies the template name.
protocol	String	Specifies the protocol supported by the template. The following protocols are supported: email sms functionstage http and https
tag_names	Array of strings	Specifies the variable list.
create_time	String	Specifies when the template was created. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
update_time	String	Specifies the last time when the template was updated. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.

Status code: 400

Table 5-121 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-122 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-123 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-124 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying message templates

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/message_template?
offset=0&limit=2&message_template_name=test1&protocol=email

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "message_template_count" : 3,
  "message_templates" : [ {
    "message_template_name" : "confirm_message",
    "protocol" : "sms",
    "update_time" : "2016-08-02T08:22:18Z",
    "create_time" : "2016-08-02T08:22:18Z",
    "tag_names" : [ "topic_urn" ],
    "message_template_id" : "79227dfdf88d4e52a1820ca1eb411635"
  }, {
    "message_template_name" : "confirm_message",
    "protocol" : "email",
    "update_time" : "2016-08-02T08:22:19Z",
    "create_time" : "2016-08-02T08:22:19Z",
    "tag_names" : [ "topic_id" ],
    "message_template_id" : "ecf63465804a4b10a0573980be78ffba"
  }, {
    "message_template_name" : "confirm_message",
    "protocol" : "https",
    "update_time" : "2016-08-02T08:22:20Z",
    "create_time" : "2016-08-02T08:22:20Z",
    "tag_names" : [ "topic_id" ],
    "message_template_id" : "57ba8dcecda844878c5dd5815b65d10f"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.3.2 Creating a Message Template

Function

This API is used to create a message template for quick message sending and reduced request data volume.

By default, a user can create a maximum of 100 message templates. However, in rare high-concurrency scenarios, extra templates may be successfully created.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/message_template

Table 5-125 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Request Parameters

Table 5-126 Request header parameters

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

Table 5-127 Request body parameters

Parameter	Mandatory	Type	Description
message_template_name	Yes	String	Specifies the template name. Enter 1 to 64 characters, and start with a letter or digit. Only letters, digits, hyphens (-), and underscores (_) are allowed.
protocol	Yes	String	Specifies the protocol supported by the template. The following protocols are supported: default email sms functionstage http and https
content	Yes	String	Specifies the template content, which supports plain text only. The template content cannot be left blank or larger than 256 KB.

Response Parameters

Status code: 201

Table 5-128 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
message_template_id	String	Specifies the resource identifier of the template, which is unique.

Status code: 400

Table 5-129 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-130 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-131 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-132 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Creating an HTTPS message template with the name of **confirm_message**

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/message_template

{
  "message_template_name" : "confirm_message",
  "protocol" : "https",
  "content" : "(1/2)You are invited to subscribe to topic({topic_id}). Click the following URL to confirm subscription:(If you do not want to subscribe to this topic, ignore this message.)"
}
```

Example Responses

Status code: 201

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "message_template_id" : "57ba8dcecda844878c5dd5815b65d10f"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating an HTTPS message template with the name of **confirm_message**

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class CreateMessageTemplateSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateMessageTemplateRequest request = new CreateMessageTemplateRequest();
        CreateMessageTemplateRequestBody body = new CreateMessageTemplateRequestBody();
        body.withContent("(1/2)You are invited to subscribe to topic({topic_id}). Click the following URL to
confirm subscription:(If you do not want to subscribe to this topic, ignore this message.)");
        body.withProtocol("https");
        body.withMessageTemplateName("confirm_message");
        request.withBody(body);
        try {
            CreateMessageTemplateResponse response = client.createMessageTemplate(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

Creating an HTTPS message template with the name of **confirm_message**

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateMessageTemplateRequest()
        request.body = CreateMessageTemplateRequestBody(
            content="(1/2)You are invited to subscribe to topic({topic_id}). Click the following URL to confirm
subscription:(If you do not want to subscribe to this topic, ignore this message.)",
            protocol="https",
            message_template_name="confirm_message"
        )
        response = client.create_message_template(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

Creating an HTTPS message template with the name of **confirm_message**

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateMessageTemplateRequest{}
    request.Body = &model.CreateMessageTemplateRequestBody{
        Content: "(1/2)You are invited to subscribe to topic({topic_id}). Click the following URL to confirm
subscription:(If you do not want to subscribe to this topic, ignore this message.)",
        Protocol: "https",
        MessageTemplateName: "confirm_message",
    }
    response, err := client.CreateMessageTemplate(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
201	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.3.3 Querying Details About a Message Template

Function

This API is used to query template details, including the template content.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/message_template/{message_template_id}

Table 5-133 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
message_template_id	Yes	String	Specifies the unique resource ID of a template. To obtain the ID, see **Querying Message Templates**.

Request Parameters

Table 5-134 Request header parameters

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

Response Parameters

Status code: 200

Table 5-135 Response body parameters

Parameter	Type	Description
message_template_id	String	Specifies the template ID.
message_template_name	String	Specifies the template name.
protocol	String	Specifies the protocol supported by the template. The following protocols are supported: default email sms functionstage http and https

Parameter	Type	Description
tag_names	Array of strings	Specifies the variable list. The variable name will be quoted in braces ({}) in the template. When you use a template to send messages, you can replace the variable with any content.
create_time	String	Specifies when the template was created. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
update_time	String	Specifies the last time when the template was updated. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
content	String	Specifies the template content.
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-136 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-137 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-138 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-139 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying details about a message template

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/message_template/
57ba8dcecda844878c5dd5815b65d10f

null
```

Example Responses

Status code: 200

OK

```
{
  "message_template_id" : "57ba8dcecda844878c5dd5815b65d10f",
  "message_template_name" : "confirm_message",
  "protocol" : "https",
  "tag_names" : [ "topic_id_id4" ],
  "create_time" : "2016-08-02T08:22:20Z",
  "update_time" : "2016-08-02T08:22:25Z",
  "content" : "(1/24)You are invited to subscribe to topic({topic_id_id4}). Click the following URL to confirm 
subscription:(If you do not want to subscribe to this topic, ignore this message.)",
  "request_id" : "ba79ca8f794f4f50985ce7b98a401b47"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.3.4 Modifying a Message Template

Function

This API is used to modify the message template content.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/message_template/{message_template_id}

Table 5-140 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
message_template_id	Yes	String	Specifies the unique resource ID of a template. To obtain the ID, see **Querying Message Templates**.

Request Parameters

Table 5-141 Request header parameters

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

Table 5-142 Request body parameters

Parameter	Mandatory	Type	Description
content	Yes	String	Specifies the template content.

Response Parameters

Status code: 200

Table 5-143 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-144 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-145 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-146 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-147 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Modifying the content of a message template

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/message_template/
b3ffa2cdda574168826316f0628f774f

{
  "content" : "(1/22)You are invited to subscribe to topic({topic_id_id1}). Click the following URL to confirm
subscription:(If you do not want to subscribe to this topic, ignore this message.)"
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying the content of a message template

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class UpdateMessageTemplateSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateMessageTemplateRequest request = new UpdateMessageTemplateRequest();
        request.withMessageTemplateId("{message_template_id}");
        UpdateMessageTemplateRequestBody body = new UpdateMessageTemplateRequestBody();
        body.withContent("(1/22)You are invited to subscribe to topic({topic_id_id1}). Click the following URL to confirm subscription:(If you do not want to subscribe to this topic, ignore this message.)");
        request.withBody(body);
        try {
            UpdateMessageTemplateResponse response = client.updateMessageTemplate(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

Modifying the content of a message template

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateMessageTemplateRequest()
        request.message_template_id = "{message_template_id}"
        request.body = UpdateMessageTemplateRequestBody(
            content="(1/22)You are invited to subscribe to topic({topic_id_id1}). Click the following URL to confirm subscription:(If you do not want to subscribe to this topic, ignore this message.)"
        )
        response = client.update_message_template(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

Modifying the content of a message template

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateMessageTemplateRequest{}
    request.MessageTemplateId = "{message_template_id}"
    request.Body = &model.UpdateMessageTemplateRequestBody{
        Content: "(1/22)You are invited to subscribe to topic({topic_id_id1}). Click the following URL to 
confirm subscription:(If you do not want to subscribe to this topic, ignore this message.)",
    }
    response, err := client.UpdateMessageTemplate(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.3.5 Deleting a Message Template

Function

This API is used to delete a message template. After you delete the template, you can no longer use it to publish messages.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/message_template/{message_template_id}

Table 5-148 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
message_template_id	Yes	String	Specifies the unique resource ID of a template. To obtain the ID, see **Querying Message Templates**.

Request Parameters

Table 5-149 Request header parameters

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

Response Parameters

Status code: 200

Table 5-150 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-151 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-152 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-153 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-154 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a message template

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/message_template/
b3ffa2cdda574168826316f0628f774e

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.4 Message Publishing

5.4.1 Publishing a Message

Function

This API is used to publish messages to a topic. After a message ID is returned, the message has been saved and is to be pushed to the subscribers of the topic. To send messagesto subscribers successfully, ensure that your message content complies with local laws and regulations.

The following three message formats are supported:

message

message_structure

message_template_name

If the three formats are specified at the same time, they take effect in the following sequence:

message_structure > **message_template_name** > **message**

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/topics/{topic_urn}/publish

Table 5-155 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-156 Request header parameters

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

Table 5-157 Request body parameters

Parameter	Mandatory	Type	Description
subject	No	String	Specifies the message subject, which is used as the email subject when you publish email messages. The length of the subject cannot exceed 512 bytes.
message	No	String	Message content. The message content must be UTF-8-coded and can be no more than 256 KB. If the endpoint is a phone number, the maximum length of an SMS message is 490 characters, or the message may be intercepted. The SMS message cannot contain square brackets ([]).Note:The message sending modes inclue **message**, **message_structure**, and **message_template_name**.Set at least one of them. If you set all of them, they take effect in the following sequence:**message_structure** > **message_template_name** > **message**

Parameter	Mandatory	Type	Description
message_structure	No	String	Message structure, which is a JSON string. The options are **email**, **sms**, **http**, **https**, **functiongraph**, **functionstage**, **dingding**, **wechat**, **feishu**, and **welink**. The **default** protocol is mandatory. If the system fails to match any other protocols, the message using the **default** protocol is sent. The **msgType** field must be specified for messages of the DingTalk chatbot, WeCom chatbot, Lark chatbot, or WeLink protocol. The DingTalk, WeCom, or Lark chatbot protocol support messages in **text** or **markdown** format. WeLink and WeLink (red version) chatbots support messages in **text** format.For the DingTalk chatbot protocol, SMN supports the @ group member function through the **at** fields. When you need to @ group members, set **isAtAll** to **true**. You can enter the phone numbers of people who need to be @ for the **atMobiles** field or enter the DingTalk user IDs of people who need to be @ for the **atUserIds** field. When you use the **atMobiles** or **atUserIds** field, add **@** before each phone number or user ID in the message content. For details about the display effect, see the official DingTalk document.Note:The message sending modes inclue **message**, **message_structure**, and **message_template_name**.Set at least one of them. If you set all of them, they take effect in the following sequence:**message_structure**

Parameter	Mandatory	Type	Description
			> **message_template_name** > **message**
message_template_name	No	String	Message template name. To obtain the name, see [Querying Message Templates](ListMessageTemplates.xml).Note:The message sending modes inclue **message**, **message_structure**, and **message_template_name**.Set at least one of them. If you set all of them, they take effect in the following sequence:**message_structure** > **message_template_name** > **message**
tags	No	Map<String,String>	Specifies the dictionary consisting of variable parameters and values. The value cannot be left blank. This parameter is mandatory when you use a message template to publish messages. The key in the dictionary is the parameter name in the message template and contains a maximum of 21 characters. The value in the dictionary is the value after the key in the message template is replaced. The value cannot exceed 1 KB.
time_to_live	No	String	Specifies the maximum retention period of a message in SMN. After the retention period expires, SMN does not send this message. Unit: second Default value: 3600s (one hour) The retention period must be a positive integer less than or equal to 86,400 (3600 x 24).
message_attributes	No	Array of **MessageAttribute** objects	Specifies the message attribute list.

Parameter	Mandatory	Type	Description
locale	No	String	Language of the additional system content in messages sent by SMN. If this parameter is not specified, language settings for the account are used. The value is the language supported by the site, for example, zh-cn or en-us.

Table 5-158 MessageAttribute

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the attribute name. **name** can contain 1 to 32 characters, including only lowercase letters, digits, and underscores (_). **name** cannot start or end with an underscore, nor contain consecutive underscores.
type	Yes	String	Specifies the attribute type. The value can be one of the following: **STRING** **STRING_ARRAY** **PROTOCOL**
value	Yes	Object	Specifies the attribute value. When you set **type** to **STRING**, **value** can contain 1 to 32 characters, including only letters, digits, and underscores (*). When you set **type** to **STRING_ARRAY**, **value** is an array containing 1 to 10 strings. Each string in the array must be unique. Each string in the array must contain 1 to 32 characters, including only letters, digits, and underscores (*). When you set **type** to **PROTOCOL**, **value** is a string array of supported protocol types.

Response Parameters

Status code: 200

Table 5-159 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
message_id	String	Specifies the message ID, which is unique.

Status code: 400

Table 5-160 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-161 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-162 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 429

Table 5-163 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-164 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

- Publishing a message

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_create_topic_v2/publish

{
  "subject" : "test message v2",
  "message" : "Message test message v2",
  "time_to_live" : "3600",
  "message_attributes" : [ {
    "name" : "smn_protocol",
    "type" : "PROTOCOL",
    "value" : [ "email", "sms" ]
  } ]
}
```

- Publishing messages using a message structure

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_create_topic_v2/publish
```

```
{
  "subject" : "test message v2",
  "message_structure" : "{   \"default\": \"Message structure test message v2\",   \"sms\": \"Message 
structure test message v2\",   \"email\": \"Message structure test message v2\",   \"wechat\": \"{\\
\"msgtype\\\":\\\"text\\\",\\\"text\\\":{\\\"content\\\":\\\"Message structure test message v2\\\"}}\",   
\"dingding\": \"{\\\"msgtype\\\":\\\"markdown\\\",\\\"at\\\":{\\\"atMobiles\\\": [\\\"15XXXXXX\\\"],\\
\"atUserIds\\\": [\\\"1234567XXX\\\"],\\\"isAtAll\\\": true},\\\"markdown\\\":{\\\"content\\\":\\
\"Message structure test message v2 @15XXXXXX @1234567XXX\\\",\\\"title\\\":\\\"Message structure 
title\\\"}}\",   \"feishu\": \"{\\\"msgtype\\\":\\\"text\\\",\\\"text\\\":{\\\"content\\\":\\\"Message 
structure test message v2\\\"}}\",   \"welinkRed\": \"{\\\"msgtype\\\":\\\"text\\\",\\\"text\\\":{\\
\"content\\\":\\\"Message structure test message v2\\\"}}\",   \"http\": \"Message structure test 
message v2\",   \"https\": \"Message structure test message v2\",   \"dingTalkBot\": \"{\\\"msgtype\\
\":\\\"markdown\\\",\\\"markdown\\\":{\\\"content\\\":\\\"Message structure test message v2 
dingTalkBot markdown\\\",\\\"title\\\":\\\"Message structure title\\\"}}\",   \"functionstage\": 
\"Message structure test message v2\" }",
  "time_to_live" : "3600",
  "message_attributes" : [ {
    "name" : "smn_protocol",
    "type" : "PROTOCOL",
    "value" : [ "email", "sms" ]
  } ]
}
```

- Publishing messages using a template

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_create_topic_v2/publish

{
  "subject" : "test message template v2",
  "message_template_name" : "confirm_message",
  "time_to_live" : "3600",
  "tags" : {
    "topic_urn" : "topic_urn3331",
    "topic_id" : "topic_id3332"
  },
  "message_attributes" : [ {
    "name" : "smn_protocol",
    "type" : "PROTOCOL",
    "value" : [ "email", "sms" ]
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "message_id" : "bf94b63a5dfb475994d3ac34664e24f2"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Publishing a message

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class PublishMessageSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        PublishMessageRequest request = new PublishMessageRequest();
        request.withTopicUrn("{topic_urn}");
        PublishMessageRequestBody body = new PublishMessageRequestBody();
        List<MessageAttribute> listbodyMessageAttributes = new ArrayList<>();
        listbodyMessageAttributes.add(
            new MessageAttribute()
                .withName("smn_protocol")
                .withType(MessageAttribute.TypeEnum.fromValue("PROTOCOL"))
                .withValue("[email, sms]")
        );
        body.withMessageAttributes(listbodyMessageAttributes);
        body.withTimeToLive("3600");
        body.withMessage("Message test message v2");
        body.withSubject("test message v2");
        request.withBody(body);
        try {
            PublishMessageResponse response = client.publishMessage(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());
        }
    }
}
```

- Publishing messages using a message structure

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class PublishMessageSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        PublishMessageRequest request = new PublishMessageRequest();
        request.withTopicUrn("{topic_urn}");
        PublishMessageRequestBody body = new PublishMessageRequestBody();
        List<MessageAttribute> listbodyMessageAttributes = new ArrayList<>();
        listbodyMessageAttributes.add(
            new MessageAttribute()
                .withName("smn_protocol")
                .withType(MessageAttribute.TypeEnum.fromValue("PROTOCOL"))
                .withValue("[email, sms]")
        );
        body.withMessageAttributes(listbodyMessageAttributes);
        body.withTimeToLive("3600");
        body.withMessageStructure("{   "default": "Message structure test message v2",   "sms":
"Message structure test message v2",   "email": "Message structure test message v2",   "wechat":
"{\"msgtype\":\"text\",\"text\":{\"content\":\"Message structure test message v2\"}}",   "dingding":
"{\"msgtype\":\"markdown\",\"at\":{\"atMobiles\": [\"15XXXXXX\"],\"atUserIds\": [\"1234567XXX
\"],\"isAtAll\": true},\"markdown\":{\"content\":\"Message structure test message v2 @15XXXXXX
@1234567XXX\",\"title\":\"Message structure title\"}}",   "feishu": "{\"msgtype\":\"text\",\"text\":
{\"content\":\"Message structure test message v2\"}}",   "welinkRed": "{\"msgtype\":\"text\",\"text\":
{\"content\":\"Message structure test message v2\"}}",   "http": "Message structure test message
v2",   "https": "Message structure test message v2",   "dingTalkBot": "{\"msgtype\":\"markdown
\",\"markdown\":{\"content\":\"Message structure test message v2 dingTalkBot markdown\",\"title
\":\"Message structure title\"}}",   "functionstage": "Message structure test message v2" }");
        body.withSubject("test message v2");
        request.withBody(body);
        try {
            PublishMessageResponse response = client.publishMessage(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());
        }
    }
}
```

- Publishing messages using a template

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class PublishMessageSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        PublishMessageRequest request = new PublishMessageRequest();
        request.withTopicUrn("{topic_urn}");
        PublishMessageRequestBody body = new PublishMessageRequestBody();
        List<MessageAttribute> listbodyMessageAttributes = new ArrayList<>();
        listbodyMessageAttributes.add(
            new MessageAttribute()
                .withName("smn_protocol")
                .withType(MessageAttribute.TypeEnum.fromValue("PROTOCOL"))
                .withValue("[email, sms]")
        );
        Map<String, String> listbodyTags = new HashMap<>();
        listbodyTags.put("topic_urn", "topic_urn3331");
        listbodyTags.put("topic_id", "topic_id3332");
        body.withMessageAttributes(listbodyMessageAttributes);
        body.withTimeToLive("3600");
        body.withTags(listbodyTags);
        body.withMessageTemplateName("confirm_message");
        body.withSubject("test message template v2");
        request.withBody(body);
        try {
            PublishMessageResponse response = client.publishMessage(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

- Publishing a message

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = PublishMessageRequest()
        request.topic_urn = "{topic_urn}"
        listMessageAttributesbody = [
            MessageAttribute(
                name="smn_protocol",
                type="PROTOCOL",
                value="[email, sms]"
            )
        ]
        request.body = PublishMessageRequestBody(
            message_attributes=listMessageAttributesbody,
            time_to_live="3600",
            message="Message test message v2",
            subject="test message v2"
        )
        response = client.publish_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Publishing messages using a message structure

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = PublishMessageRequest()
        request.topic_urn = "{topic_urn}"
        listMessageAttributesbody = [
            MessageAttribute(
                name="smn_protocol",
                type="PROTOCOL",
                value="[email, sms]"
            )
        ]
        request.body = PublishMessageRequestBody(
            message_attributes=listMessageAttributesbody,
            time_to_live="3600",
            message_structure="{   "default": "Message structure test message v2",   "sms": "Message structure test message v2",   "email": "Message structure test message v2",   "wechat": "{\"msgtype\":\"text\",\"text\":{\"content\":\"Message structure test message v2\"}}",   "dingding": "{\"msgtype\":\"markdown\",\"at\":{\"atMobiles\": [\"15XXXXXX\"],\"atUserIds\": [\"1234567XXX\"],\"isAtAll\": true},\"markdown\":{\"content\":\"Message structure test message v2 @15XXXXXX @1234567XXX\",\"title\":\"Message structure title\"}}",   "feishu": "{\"msgtype\":\"text\",\"text\":{\"content\":\"Message structure test message v2\"}}",   "welinkRed": "{\"msgtype\":\"text\",\"text\":{\"content\":\"Message structure test message v2\"}}",   "http": "Message structure test message v2",   "https": "Message structure test message v2",   "dingTalkBot": "{\"msgtype\":\"markdown\",\"markdown\":{\"content\":\"Message structure test message v2 dingTalkBot markdown\",\"title\":\"Message structure title\"}}",   "functionstage": "Message structure test message v2" }",
            subject="test message v2"
        )
        response = client.publish_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Publishing messages using a template

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = PublishMessageRequest()
        request.topic_urn = "{topic_urn}"
        listMessageAttributesbody = [
            MessageAttribute(
                name="smn_protocol",
                type="PROTOCOL",
                value="[email, sms]"
            )
        ]
        listTagsbody = {
            "topic_urn": "topic_urn3331",
            "topic_id": "topic_id3332"
        }
        request.body = PublishMessageRequestBody(
            message_attributes=listMessageAttributesbody,
            time_to_live="3600",
            tags=listTagsbody,
            message_template_name="confirm_message",
            subject="test message template v2"
        )
        response = client.publish_message(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

- Publishing a message

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.PublishMessageRequest{}
    request.TopicUrn = "{topic_urn}"
    valueMessageAttributes:= "[email, sms]"
    var valueMessageAttributesInterface interface{} = valueMessageAttributes
    var listMessageAttributesbody = []model.MessageAttribute{
        {
            Name: "smn_protocol",
            Type: model.GetMessageAttributeTypeEnum().PROTOCOL,
            Value: &valueMessageAttributesInterface,
        },
    }
    timeToLivePublishMessageRequestBody:= "3600"
    messagePublishMessageRequestBody:= "Message test message v2"
    subjectPublishMessageRequestBody:= "test message v2"
    request.Body = &model.PublishMessageRequestBody{
        MessageAttributes: &listMessageAttributesbody,
        TimeToLive: &timeToLivePublishMessageRequestBody,
        Message: &messagePublishMessageRequestBody,
        Subject: &subjectPublishMessageRequestBody,
    }
    response, err := client.PublishMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Publishing messages using a message structure

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.PublishMessageRequest{}
    request.TopicUrn = "{topic_urn}"
    valueMessageAttributes:= "[email, sms]"
    var valueMessageAttributesInterface interface{} = valueMessageAttributes
    var listMessageAttributesbody = []model.MessageAttribute{
        {
            Name: "smn_protocol",
            Type: model.GetMessageAttributeTypeEnum().PROTOCOL,
            Value: &valueMessageAttributesInterface,
        },
    }
    timeToLivePublishMessageRequestBody:= "3600"
    messageStructurePublishMessageRequestBody:= "{   "default": "Message structure test message
v2",   "sms": "Message structure test message v2",   "email": "Message structure test message v2",
"wechat": "{\"msgtype\":\"text\",\"text\":{\"content\":\"Message structure test message v2\"}}",
"dingding": "{\"msgtype\":\"markdown\",\"at\":{\"atMobiles\": [\"15XXXXXX\"],\"atUserIds\":
[\"1234567XXX\"],\"isAtAll\": true},\"markdown\":{\"content\":\"Message structure test message v2
@15XXXXXX @1234567XXX\",\"title\":\"Message structure title\"}}",   "feishu": "{\"msgtype\":\"text
\",\"text\":{\"content\":\"Message structure test message v2\"}}",   "welinkRed": "{\"msgtype\":\"text
\",\"text\":{\"content\":\"Message structure test message v2\"}}",   "http": "Message structure test
message v2",   "https": "Message structure test message v2",   "dingTalkBot": "{\"msgtype
\":\"markdown\",\"markdown\":{\"content\":\"Message structure test message v2 dingTalkBot
markdown\",\"title\":\"Message structure title\"}}",   "functionstage": "Message structure test
message v2" }"
    subjectPublishMessageRequestBody:= "test message v2"
    request.Body = &model.PublishMessageRequestBody{
        MessageAttributes: &listMessageAttributesbody,
        TimeToLive: &timeToLivePublishMessageRequestBody,
        MessageStructure: &messageStructurePublishMessageRequestBody,
        Subject: &subjectPublishMessageRequestBody,
    }
    response, err := client.PublishMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Publishing messages using a template

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
    request := &model.PublishMessageRequest{}
    request.TopicUrn = "{topic_urn}"
    valueMessageAttributes:= "[email, sms]"
    var valueMessageAttributesInterface interface{} = valueMessageAttributes
    var listMessageAttributesbody = []model.MessageAttribute{
        {
            Name: "smn_protocol",
            Type: model.GetMessageAttributeTypeEnum().PROTOCOL,
            Value: &valueMessageAttributesInterface,
        },
    }
    var listTagsbody = map[string]string{
        "topic_urn": "topic_urn3331",
        "topic_id": "topic_id3332",
    }
    timeToLivePublishMessageRequestBody:= "3600"
    messageTemplateNamePublishMessageRequestBody:= "confirm_message"
    subjectPublishMessageRequestBody:= "test message template v2"
    request.Body = &model.PublishMessageRequestBody{
        MessageAttributes: &listMessageAttributesbody,
        TimeToLive: &timeToLivePublishMessageRequestBody,
        Tags: listTagsbody,
        MessageTemplateName: &messageTemplateNamePublishMessageRequestBody,
        Subject: &subjectPublishMessageRequestBody,
    }
    response, err := client.PublishMessage(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	Bad Request
403	Unauthorized
404	Not Found
429	Too Many Requests
500	Internal Server Error

Error Codes

See **Error Codes**.

5.4.2 Publishing a Detection Message

Function

Specifies the HTTP or HTTPS detection messages sent based on topics. These messages are used to check whether the current HTTP or HTTPS endpoint is available and whether the SMN egress can access the endpoint.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/topics/{topic_urn}/detection

Table 5-165 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-166 Request header parameters

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

Table 5-167 Request body parameters

Parameter	Mandatory	Type	Description
protocol	Yes	String	Specifies the protocol type. Only **http** and **https** are supported. This field cannot be left blank.
endpoint	Yes	String	Specifies the endpoint address to be detected. The address must start with **http://** or **https://** and cannot be left blank.
extension	No	**extension** object	Specifies the extended information for subscriptions over HTTP or HTTPS. You can specify multiple key/value pairs for the HTTP or HTTPS protocol. These key/value pairs will be carried as the request header when HTTP or HTTPS messages are sent. **extension** can be left blank. **header** can be left blank.

Table 5-168 extension

Parameter	Mandatory	Type	Description
header	No	Map<String,String>	Specifies the header for subscriptions over HTTP or HTTPS. The header must meet the following requirements: 1. **key** can contain letters, digits, and hyphens (-). **key** cannot end with a hyphen (-) nor contain consecutive hyphens (-). 2. You can specify up to 10 key/value pairs. 3. **key** must start with **x-**but not **x-smn**. Examples: **x-abc-cba** or **x-abc** 4. The total length of all key/value pairs cannot exceed 1,024 characters. 5. **key** is case insensitive. 6. **key** must be unique. 7. **value** must be an ASCII code. Unicode characters are not supported. Spaces are allowed.

Response Parameters

Status code: 200

Table 5-169 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
task_id	String	Specifies the detection task ID.

Status code: 400

Table 5-170 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-171 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-172 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 429

Table 5-173 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-174 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Example

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_create_topic_v2/detect

{
  "protocol" : "http",
  "endpoint" : "http://172.20.0.51:32000/http_notify",
  "extension" : {
    "header" : {
      "x-abc" : "1213",
      "x-abc-cba" : "test2"
    }
  }
}
```

Example Responses

Status code: 200

Detection succeeded.

```
{
  "request_id" : "adc82efa7e0042899722531681b1959a",
  "task_id" : "adc82efa7e0042899722531681b1959a"
}
```

Status code: 400

Detection failed.

```
{
  "request_id" : "373cdb9696e44454832ea02e61a2c689",
  "error_code" : "SMN.00011004",
  "error_msg" : "Parameter: Extension is invalid. Http header key is invalid."
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Example

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class PublishHttpDetectSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        PublishHttpDetectRequest request = new PublishHttpDetectRequest();
        request.withTopicUrn("{topic_urn}");
        HttpDetectRequestBody body = new HttpDetectRequestBody();
        Map<String, String> listExtensionHeader = new HashMap<>();
        listExtensionHeader.put("x-abc", "1213");
        listExtensionHeader.put("x-abc-cba", "test2");
        HttpDetectRequestBodyExtension extensionbody = new HttpDetectRequestBodyExtension();
        extensionbody.withHeader(listExtensionHeader);
        body.withExtension(extensionbody);
        body.withEndpoint("http://172.20.0.51:32000/http_notify");
        body.withProtocol(HttpDetectRequestBody.ProtocolEnum.fromValue("http"));
        request.withBody(body);
        try {
            PublishHttpDetectResponse response = client.publishHttpDetect(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

Example

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = PublishHttpDetectRequest()
        request.topic_urn = "{topic_urn}"
        listHeaderExtension = {
            "x-abc": "1213",
            "x-abc-cba": "test2"
        }
        extensionbody = HttpDetectRequestBodyExtension(
            header=listHeaderExtension
        )
        request.body = HttpDetectRequestBody(
            extension=extensionbody,
            endpoint="http://172.20.0.51:32000/http_notify",
            protocol="http"
        )
        response = client.publish_http_detect(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

Example

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.PublishHttpDetectRequest{}
    request.TopicUrn = "{topic_urn}"
    var listHeaderExtension = map[string]string{
        "x-abc": "1213",
        "x-abc-cba": "test2",
    }
    extensionbody := &model.HttpDetectRequestBodyExtension{
        Header: listHeaderExtension,
    }
    request.Body = &model.HttpDetectRequestBody{
        Extension: extensionbody,
        Endpoint: "http://172.20.0.51:32000/http_notify",
        Protocol: model.GetHttpDetectRequestBodyProtocolEnum().HTTP,
    }
    response, err := client.PublishHttpDetect(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	Detection succeeded.
400	Detection failed.
403	Unauthorized
404	Not Found
429	Too Many Requests
500	Internal Server Error

Error Codes

See **Error Codes**.

5.4.3 Obtaining the HTTP Detection Result

Function

This API is used to query the detection result based on **task_id** returned by the HTTP detection.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics/{topic_urn}/detection/{task_id}

Table 5-175 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.
task_id	Yes	String	Specifies the ID of the HTTP detection task.

Request Parameters

Table 5-176 Request header parameters

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

Response Parameters

Status code: 200

Table 5-177 Response body parameters

Parameter	Type	Description
detail	**detail** object	Specifies the endpoint detection details.
status	Integer	Specifies the status of the HTTP detection task. **0** indicates that the task was successful and the endpoint was available. **1** indicates that the task was not executed. **2** indicates that the task failed and the endpoint was unavailable.
request_id	String	Specifies the request ID.

Table 5-178 detail

Parameter	Type	Description
httpCode	Integer	Specifies the HTTP code returned by the endpoint detected. **0** indicates that the user was blacklisted. **-1** indicates that the endpoint did not respond within 5 seconds. **-2** indicates that the queue was full and the HTTP detection task was not executed. Other HTTP codes are the actual values returned by the endpoint.
httpResponse	String	Specifies the response body for endpoint detection. If the HTTP code is **0**, **-1**, **-2**, or **2***xx*, the response body content is fixed and defined by SMN. If the HTTP code is not one of the values mentioned above, the response body content is the content returned by the endpoint.

Status code: 400

Table 5-179 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Status code: 403

Table 5-180 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-181 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-182 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Example

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_create_topic_v2/detect/
c414cfd4326a488e9c6a315ce8dc5fe3
```

Example Responses

Status code: 200

OK

```
{
  "detail" : {
    "httpCode" : 200,
    "httpResponse" : "Http endpoint is available."
  },
  "status" : 0,
  "request_id" : "e4bde8f67bd64402804011160c9f8772"
}
```

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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class ShowHttpDetectResultSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowHttpDetectResultRequest request = new ShowHttpDetectResultRequest();
        request.withTopicUrn("{topic_urn}");
        request.withTaskId("{task_id}");
        try {
            ShowHttpDetectResultResponse response = client.showHttpDetectResult(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 huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowHttpDetectResultRequest()
        request.topic_urn = "{topic_urn}"
        request.task_id = "{task_id}"
        response = client.show_http_detect_result(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"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowHttpDetectResultRequest{}
    request.TopicUrn = "{topic_urn}"
    request.TaskId = "{task_id}"
    response, err := client.ShowHttpDetectResult(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.5 Tags

5.5.1 Querying Resources by Tag

Function

This API is used to query resources by tag.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/{resource_type}/resource_instances/action

Table 5-183 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
resource_type	Yes	String	Specifies the resource type. Possible values are: smn_topic: topic smn_sms: SMS smn_application: mobile push

Request Parameters

Table 5-184 Request header parameters

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

Table 5-185 Request body parameters

Parameter	Mandatory	Type	Description
tags	No	Array of **ResourceTags** objects	A maximum of 10 tag keys are allowed in each query operation. Each tag key can have up to 10 tag values. The structure body is mandatory. The tag key cannot be left blank or be an empty string. Each tag key and tag values of one key must be unique. Resources identified by different keys are in AND relationship.
tags_any	No	Array of **ResourceTags** objects	A maximum of 10 tag keys are allowed in each query operation. Each tag key can have up to 10 tag values. The structure body is mandatory. The tag key cannot be left blank or be an empty string. Each tag key and tag values of one key must be unique. Resources identified by different keys are in OR relationship.
not_tags	No	Array of **ResourceTags** objects	A maximum of 10 tag keys are allowed in each query operation. Each tag key can have up to 10 tag values. The structure body is mandatory. The tag key cannot be left blank or be an empty string. Each tag key and tag values of one key must be unique. Resources identified by different keys are in NAND relationship.

Parameter	Mandatory	Type	Description
not_tags_any	No	Array of **ResourceTags** objects	A maximum of 10 tag keys are allowed in each query operation. Each tag key can have up to 10 tag values. The structure body is mandatory. The tag key cannot be left blank or be an empty string. Each tag key and tag values of one key must be unique. Resources identified by different keys are in NOR relationship.
offset	No	String	Specifies the index position. The query starts from the next data record indexed by this parameter. You do not need to specify this parameter when you query resources on the first page. When you query resources on subsequent pages, set this parameter to the location returned in the response body for the previous query.If action is set to count, this parameter does not take effect.If you set action to filter, the offset value is 0 by default. The offset value cannot be a negative integer.
limit	No	String	Specifies the maximum number of resources to be queried. If action is set to count, this parameter does not take effect. If you set action to filter, the value of this parameter is 1000 by default. The maximum value of limit is 1000 and the minimum value is 1.

Parameter	Mandatory	Type	Description
action	Yes	String	Specifies the operation to be performed. The value can be filter (filtering) or count (querying the total number). filter indicates pagination query. count indicates that the total number of query results meeting the search criteria will be returned.
matches	No	Array of **TagMatch** objects	Specifies the key-value pair to be matched. The key can only be resource_name. The value will be exactly matched.
without_any_tag	No	Boolean	No tag is contained. If this parameter is set to **true**, all resources without tags are queried. In this case, the **tags**, **tags_any**, **not_tags**, and **not_tags_any** fields are ignored.

Table 5-186 ResourceTags

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. ● A key can contain a maximum of 127 Unicode characters. ● key must be specified.
values	Yes	Array of strings	Specifies the tag values. ● Each tag contains a maximum of 10 values. ● Values of the same tag must be unique. ● It can contain a maximum of 255 Unicode characters. ● If this parameter is left blank, any value can be used. ● All values of a tag key are in the OR relationship.

Table 5-187 TagMatch

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the key field to be matched. key can only be resource_name. In such a case, value is the ECS name. ● The key must be unique, and the value is used for matching. ● The key field is a fixed dictionary value. ● key cannot be left blank.
value	Yes	String	Specifies the value. key can only be resource_name. In such a case, value is the ECS name. ● It can contain a maximum of 255 Unicode characters. ● This field cannot be left blank.

Response Parameters

Status code: 200

Table 5-188 Response body parameters

Parameter	Type	Description
resources	Array of **TagResource** objects	Specifies the returned resource list.
total_count	Integer	Specifies the total number of records.

Table 5-189 TagResource

Parameter	Type	Description
resource_id	String	Specifies the resource ID.

Parameter	Type	Description
resource_detail	**ResourceDetail** object	Specifies the resource details.
tags	Array of **ResourceTag** objects	Specifies the list of tags.
resource_name	String	Specifies the resource name.

Table 5-190 ResourceDetail

Parameter	Type	Description
enterprise_project_id	String	Specifies the enterprise project ID.
detailId	String	Specifies the details ID.
topic_urn	String	Specifies the unique identifier of the topic.
display_name	String	Specifies the display name.

Table 5-191 ResourceTag

Parameter	Type	Description
key	String	Specifies the key field to be matched. key can only be resource_name. In such a case, value is the ECS name. ● The key must be unique, and the value is used for matching. ● The key field is a fixed dictionary value. ● key cannot be left blank.
value	String	Specifies the value. key can only be resource_name. In such a case, value is the ECS name. ● It can contain a maximum of 255 Unicode characters. ● This field cannot be left blank.

Status code: 400

Table 5-192 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-193 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-194 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-195 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

- Using tags to filter resources

```
POST https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/resource_instances/action

{
  "offset" : "100",
  "limit" : "100",
  "action" : "filter",
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  }, {
    "key" : "key2",
    "values" : [ "*value21", "value22" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ]
}
```

- Using tags to query the total number of resources

```
POST https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/resource_instances/action

{
  "action" : "count",
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "*value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resouurce"
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "resources" : {
    "resource_detail" : {
      "topic_urn" : "urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:resouece1",
      "display_name" : "testtest",
      "enterprise_project_id" : "0"
    },
    "resource_id" : "cffe4fc4c9a54219b60dbaf7b586e132",
    "resource_name" : "resouece1",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ]
  },
  "total_count" : 1000
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Using tags to filter resources

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        ListResourceInstancesRequest request = new ListResourceInstancesRequest();
        request.withResourceType("{resource_type}");
        ListInstanceRequestBody body = new ListInstanceRequestBody();
        List<TagMatch> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new TagMatch()
                .withKey("resource_name")
                .withValue("resource1")
        );
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<ResourceTags> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("*value21");
        listNotTagsValues.add("value22");
        List<String> listNotTagsValues1 = new ArrayList<>();
        listNotTagsValues1.add("*value1");
        listNotTagsValues1.add("value2");
        List<ResourceTags> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listNotTagsValues1)
        );
        listbodyNotTags.add(
            new ResourceTags()
                .withKey("key2")
                .withValues(listNotTagsValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<ResourceTags> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("*value1");
        listTagsValues.add("value2");
        List<ResourceTags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        body.withMatches(listbodyMatches);
        body.withAction("filter");
        body.withLimit("100");
        body.withOffset("100");
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withNotTags(listbodyNotTags);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        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());
        }
    }
}
```

- Using tags to query the total number of resources

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        ListResourceInstancesRequest request = new ListResourceInstancesRequest();
        request.withResourceType("{resource_type}");
        ListInstanceRequestBody body = new ListInstanceRequestBody();
        List<TagMatch> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new TagMatch()
                .withKey("resource_name")
                .withValue("resouurce")
        );
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<ResourceTags> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("value1");
        listNotTagsValues.add("*value2");
        List<ResourceTags> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<ResourceTags> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("value1");
        listTagsValues.add("value2");
        List<ResourceTags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTags()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        body.withMatches(listbodyMatches);
        body.withAction("count");
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withNotTags(listbodyNotTags);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        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

- Using tags to filter resources

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListResourceInstancesRequest()
        request.resource_type = "{resource_type}"
        listMatchesbody = [
            TagMatch(
                key="resource_name",
                value="resource1"
            )
        ]
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            ResourceTags(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesNotTags = [
            "*value21",
            "value22"
        ]
        listValuesNotTags1 = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            ResourceTags(
                key="key1",
                values=listValuesNotTags1
            ),
            ResourceTags(
                key="key2",
                values=listValuesNotTags
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            ResourceTags(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "*value1",
            "value2"
        ]
        listTagsbody = [
            ResourceTags(
                key="key1",
                values=listValuesTags
            )
        ]
        request.body = ListInstanceRequestBody(
            matches=listMatchesbody,
            action="filter",
            limit="100",
            offset="100",
            not_tags_any=listNotTagsAnybody,
            not_tags=listNotTagsbody,
            tags_any=listTagsAnybody,
            tags=listTagsbody
        )
        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)
```

- Using tags to query the total number of resources

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListResourceInstancesRequest()
        request.resource_type = "{resource_type}"
        listMatchesbody = [
            TagMatch(
                key="resource_name",
                value="resouurce"
            )
        ]
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            ResourceTags(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesNotTags = [
            "value1",
            "*value2"
        ]
        listNotTagsbody = [
            ResourceTags(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            ResourceTags(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            ResourceTags(
                key="key1",
                values=listValuesTags
            )
        ]
        request.body = ListInstanceRequestBody(
            matches=listMatchesbody,
            action="count",
            not_tags_any=listNotTagsAnybody,
            not_tags=listNotTagsbody,
            tags_any=listTagsAnybody,
            tags=listTagsbody
        )
        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

- Using tags to filter resources

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListResourceInstancesRequest{}
    request.ResourceType = "{resource_type}"
    var listMatchesbody = []model.TagMatch{
        {
            Key: "resource_name",
            Value: "resource1",
        },
    }
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesNotTags = []string{
        "*value21",
        "value22",
    }
    var listValuesNotTags1 = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesNotTags1,
        },
        {
            Key: "key2",
            Values: listValuesNotTags,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value1",
        "value2",
    }
    var listTagsbody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    limitListInstanceRequestBody:= "100"
    offsetListInstanceRequestBody:= "100"
    request.Body = &model.ListInstanceRequestBody{
        Matches: &listMatchesbody,
        Action: "filter",
        Limit: &limitListInstanceRequestBody,
        Offset: &offsetListInstanceRequestBody,
        NotTagsAny: &listNotTagsAnybody,
        NotTags: &listNotTagsbody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
    }
    response, err := client.ListResourceInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Using tags to query the total number of resources

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListResourceInstancesRequest{}
    request.ResourceType = "{resource_type}"
    var listMatchesbody = []model.TagMatch{
        {
            Key: "resource_name",
            Value: "resouurce",
        },
    }
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesNotTags = []string{
        "value1",
        "*value2",
    }
    var listNotTagsbody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    var listTagsbody = []model.ResourceTags{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    request.Body = &model.ListInstanceRequestBody{
        Matches: &listMatchesbody,
        Action: "count",
        NotTagsAny: &listNotTagsAnybody,
        NotTags: &listNotTagsbody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
    }
    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	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.5.2 Adding or Deleting Resource Tags in Batches

Function

This API is used to add or delete tags in batches for a specified resource. You can add a maximum of 10 tags to a resource.

The API is idempotent. When you are creating tags, if there are duplicate keys in the request body, an error is reported.

If a to-be-created tag has the same key as an existing tag, the tag will be created and overwrite the existing one.

During tag deletion, if some tags do not exist, the deletion is considered to be successful by default. The character set of the tags will not be checked.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/{resource_type}/{resource_id}/tags/action

Table 5-196 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
resource_type	Yes	String	Specifies the resource type. Possible values are: smn_topic: topic smn_sms: SMS smn_application: mobile push
resource_id	Yes	String	Specifies the resource ID. To obtain the value of resource_id, perform the following steps: When resource_type is set to smn_topic: Add X-SMN-RESOURCEID-TYPE=name in the request header and set the resource ID to the topic name. Call the GetResourceInstances API to obtain the resource ID. If you set resource_type to smn_sms, the resource ID is the signature ID, which can be obtained on the console.

Request Parameters

Table 5-197 Request header parameters

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

Table 5-198 Request body parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of **ResourceTag** objects	Specifies the tag list. For details, see table 1. When you delete tags, the tag structure cannot be missing, and the key cannot be left blank or be an empty string. The system does not check the character set when deleting a tag.
action	Yes	String	Specifies the operation to be performed. The value can be create or delete only.

Table 5-199 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the key field to be matched. key can only be resource_name. In such a case, value is the ECS name. ● The key must be unique, and the value is used for matching. ● The key field is a fixed dictionary value. ● key cannot be left blank.

Parameter	Mandatory	Type	Description
value	Yes	String	Specifies the value. key can only be resource_name. In such a case, value is the ECS name. ● It can contain a maximum of 255 Unicode characters. ● This field cannot be left blank.

Response Parameters

Status code: 400

Table 5-200 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-201 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-202 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-203 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

- Batch adding tags to resources

```
POST https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags/action

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

- Batch deleting resource tags

```
POST https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags/action

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

- Batch adding tags to resources

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class BatchCreateOrDeleteResourceTagsSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateOrDeleteResourceTagsRequest request = new BatchCreateOrDeleteResourceTagsRequest();
        request.withResourceType("{resource_type}");
        request.withResourceId("{resource_id}");
        BatchCreateOrDeleteResourceTagsRequestBody body = new BatchCreateOrDeleteResourceTagsRequestBody();
        List<ResourceTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new ResourceTag()
                .withKey("key")
                .withValue("value3")
        );
        body.withAction("create");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateOrDeleteResourceTagsResponse response = client.batchCreateOrDeleteResourceTags(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());
        }
    }
}
```

- Batch deleting resource tags

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class BatchCreateOrDeleteResourceTagsSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateOrDeleteResourceTagsRequest request = new 
BatchCreateOrDeleteResourceTagsRequest();
        request.withResourceType("{resource_type}");
        request.withResourceId("{resource_id}");
        BatchCreateOrDeleteResourceTagsRequestBody body = new 
BatchCreateOrDeleteResourceTagsRequestBody();
        List<ResourceTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTag()
                .withKey("key1")
        );
        listbodyTags.add(
            new ResourceTag()
                .withKey("key2")
                .withValue("value3")
        );
        body.withAction("delete");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateOrDeleteResourceTagsResponse response = 
client.batchCreateOrDeleteResourceTags(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

- Batch adding tags to resources

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchCreateOrDeleteResourceTagsRequest()
        request.resource_type = "{resource_type}"
        request.resource_id = "{resource_id}"
        listTagsbody = [
            ResourceTag(
                key="key1",
                value="value1"
            ),
            ResourceTag(
                key="key",
                value="value3"
            )
        ]
        request.body = BatchCreateOrDeleteResourceTagsRequestBody(
            action="create",
            tags=listTagsbody
        )
        response = client.batch_create_or_delete_resource_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)
```

- Batch deleting resource tags

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchCreateOrDeleteResourceTagsRequest()
        request.resource_type = "{resource_type}"
        request.resource_id = "{resource_id}"
        listTagsbody = [
            ResourceTag(
                key="key1"
            ),
            ResourceTag(
                key="key2",
                value="value3"
            )
        ]
        request.body = BatchCreateOrDeleteResourceTagsRequestBody(
            action="delete",
            tags=listTagsbody
        )
        response = client.batch_create_or_delete_resource_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

- Batch adding tags to resources

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateOrDeleteResourceTagsRequest{}
    request.ResourceType = "{resource_type}"
    request.ResourceId = "{resource_id}"
    var listTagsbody = []model.ResourceTag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key",
            Value: "value3",
        },
    }
    request.Body = &model.BatchCreateOrDeleteResourceTagsRequestBody{
        Action: "create",
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateOrDeleteResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Batch deleting resource tags

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateOrDeleteResourceTagsRequest{}
    request.ResourceType = "{resource_type}"
    request.ResourceId = "{resource_id}"
    var listTagsbody = []model.ResourceTag{
        {
            Key: "key1",
        },
        {
            Key: "key2",
            Value: "value3",
        },
    }
    request.Body = &model.BatchCreateOrDeleteResourceTagsRequestBody{
        Action: "delete",
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateOrDeleteResourceTags(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.5.3 Querying Resource Tags

Function

This API is used to query tags of a specified resource.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/{resource_type}/{resource_id}/tags

Table 5-204 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
resource_type	Yes	String	Specifies the resource type. The type can be the following: smn_topic: topic smn_sms: SMS smn_application: mobile push
resource_id	Yes	String	Specifies the resource ID. To obtain the value of resource_id, perform the following steps: When resource_type is set to smn_topic: Add X-SMN-RESOURCEID-TYPE=name in the request header and set the resource ID to the topic name. Call the GetResourceInstances API to obtain the resource ID. If you set resource_type to smn_sms, the resource ID is the signature ID, which can be obtained on the console.

Request Parameters

Table 5-205 Request header parameters

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

Response Parameters

Status code: 200

Table 5-206 Response body parameters

Parameter	Type	Description
tags	Array of **ResourceTag** objects	Specifies resource tags.

Table 5-207 ResourceTag

Parameter	Type	Description
key	String	Specifies the key field to be matched. key can only be resource_name. In such a case, value is the ECS name. ● The key must be unique, and the value is used for matching. ● The key field is a fixed dictionary value. ● key cannot be left blank.
value	String	Specifies the value. key can only be resource_name. In such a case, value is the ECS name. ● It can contain a maximum of 255 Unicode characters. ● This field cannot be left blank.

Status code: 400

Table 5-208 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-209 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-210 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-211 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying resource tags

```
GET https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags

null
```

Example Responses

Status code: 200

OK

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

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.5.4 Adding a Resource Tag

Function

You can add a maximum of 10 tags to a resource. The API is idempotent. If a to-be-created tag has the same key as an existing tag, the tag will be created and overwrite the existing one.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/{resource_type}/{resource_id}/tags

Table 5-212 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
resource_type	Yes	String	Specifies the resource type. Possible values are: smn_topic: topic smn_sms: SMS smn_application: mobile push
resource_id	Yes	String	Specifies the resource ID. To obtain the value of resource_id, perform the following steps: When resource_type is set to smn_topic: Add X-SMN-RESOURCEID-TYPE=name in the request header and set the resource ID to the topic name. Call the GetResourceInstances API to obtain the resource ID. If you set resource_type to smn_sms, the resource ID is the signature ID, which can be obtained on the console.

Request Parameters

Table 5-213 Request header parameters

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

Table 5-214 Request body parameters

Parameter	Mandatory	Type	Description
tag	Yes	**ResourceTag** object	Specifies the resource tag structure.

Table 5-215 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the key field to be matched. key can only be resource_name. In such a case, value is the ECS name. ● The key must be unique, and the value is used for matching. ● The key field is a fixed dictionary value. ● key cannot be left blank.
value	Yes	String	Specifies the value. key can only be resource_name. In such a case, value is the ECS name. ● It can contain a maximum of 255 Unicode characters. ● This field cannot be left blank.

Response Parameters

Status code: 400

Table 5-216 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-217 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-218 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-219 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Adding a resource tag

```
POST https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Adding a resource 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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class CreateResourceTagSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateResourceTagRequest request = new CreateResourceTagRequest();
        request.withResourceType("{resource_type}");
        request.withResourceId("{resource_id}");
        CreateResourceTagRequestBody body = new CreateResourceTagRequestBody();
        ResourceTag tagbody = new ResourceTag();
        tagbody.withKey("DEV")
            .withValue("DEV1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateResourceTagResponse response = client.createResourceTag(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

Adding a resource tag

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateResourceTagRequest()
        request.resource_type = "{resource_type}"
        request.resource_id = "{resource_id}"
        tagbody = ResourceTag(
            key="DEV",
            value="DEV1"
        )
        request.body = CreateResourceTagRequestBody(
            tag=tagbody
        )
        response = client.create_resource_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

Adding a resource tag

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateResourceTagRequest{}
    request.ResourceType = "{resource_type}"
    request.ResourceId = "{resource_id}"
    tagbody := &model.ResourceTag{
        Key: "DEV",
        Value: "DEV1",
    }
    request.Body = &model.CreateResourceTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateResourceTag(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.5.5 Querying Project Tags

Function

This API is used to query all tags of a resource type in a specified region.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/{resource_type}/tags

Table 5-220 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
resource_type	Yes	String	Specifies the resource type. The type can be the following: smn_topic: topic smn_sms: SMS smn_application: mobile push

Request Parameters

Table 5-221 Request header parameters

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

Response Parameters

Status code: 200

Table 5-222 Response body parameters

Parameter	Type	Description
tags	Array of **ResourceTags** objects	Specifies resource tags.

Table 5-223 ResourceTags

Parameter	Type	Description
key	String	Specifies the tag key. ● A key can contain a maximum of 127 Unicode characters. ● key must be specified.
values	Array of strings	Specifies the tag values. ● Each tag contains a maximum of 10 values. ● Values of the same tag must be unique. ● It can contain a maximum of 255 Unicode characters. ● If this parameter is left blank, any value can be used. ● All values of a tag key are in the OR relationship.

Status code: 400

Table 5-224 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-225 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-226 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-227 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Querying project tags

```
GET https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/tags

null
```

Example Responses

Status code: 200

OK

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

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.5.6 Deleting a Resource Tag

Function

This API is used to delete resource tags. The API is idempotent. When deleting a tag, the system does not check its character set. The tag key cannot be left blank or be an empty string. If the key of the tag to be deleted does not exist, 404 will be returned.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/{resource_type}/{resource_id}/tags/{key}

Table 5-228 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Specifies the resource type. Possible values are: smn_topic: topic smn_sms: SMS smn_application: mobile push
resource_id	Yes	String	Specifies the resource ID. To obtain the value of resource_id, perform the following steps: When resource_type is set to smn_topic: Add X-SMN-RESOURCEID-TYPE=name in the request header and set the resource ID to the topic name. Call the GetResourceInstances API to obtain the resource ID. If you set resource_type to smn_sms, the resource ID is the signature ID, which can be obtained on the console.
key	Yes	String	Specifies the key value of the tag to be deleted.

Request Parameters

Table 5-229 Request header parameters

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

Response Parameters

Status code: 400

Table 5-230 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-231 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-232 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-233 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a resource tag

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags/{key}

null
```

Example Responses

None

Status Codes

Status Code	Description
204	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.6 Version Querying

5.6.1 Querying All SMN API Versions

Function

This API is used to list all SMN API versions.

Calling Method

For details, see **Calling APIs**.

URI

GET /

Request Parameters

Table 5-234 Request header parameters

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

Response Parameters

Status code: 200

Table 5-235 Response body parameters

Parameter	Type	Description
versions	Array of **VersionItem** objects	Specifies the list of all versions.

Table 5-236 VersionItem

Parameter	Type	Description
id	String	Specifies the version ID, for example, v2.
min_version	String	Specifies the microversion. If the API supports microversions, the system returns the supported minimum microversion. If no, the system returns an empty value.
status	String	Specifies the version status. CURRENT: The version is the primary version. SUPPORTED: The version is an earlier version, but it is still supported. DEPRECATED: The version is a deprecated version, which may be deleted later.

Parameter	Type	Description
updated	String	Specifies the version release time, which must be a UTC time. For example, the release time of v2 is 2014-06-28T12:20:21Z.
version	String	Specifies the microversion. If the API supports microversions, the system returns the supported minimum microversion. If no, the system returns an empty value.
links	Array of **LinksItem** objects	Specifies the API URL.

Table 5-237 LinksItem

Parameter	Type	Description
href	String	Specifies the corresponding shortcut link.
rel	String	Specifies the shortcut link marker name.

Status code: 400

Table 5-238 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-239 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Status code: 404

Table 5-240 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-241 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

```
GET https://{SMN_Endpoint}/

null
```

Example Responses

Status code: 200

OK

```
{
  "versions" : {
    "id" : "v2",
    "links" : [ {
      "href" : "https://127.0.0.1/v2",
      "rel" : "self"
    } ],
    "min_version" : "",
    "status" : "CURRENT",
    "updated" : "2018-09-19T00:00:00Z",
    "version" : ""
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.6.2 Querying the SMN API v2 Version

Function

This API is used to query the version of SMN API v2.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2

Request Parameters

Table 5-242 Request header parameters

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

Response Parameters

Status code: 200

Table 5-243 Response body parameters

Parameter	Type	Description
version	**VersionItem** object	Specifies the list of all v2 versions.

Table 5-244 VersionItem

Parameter	Type	Description
id	String	Specifies the version ID, for example, v2.
min_version	String	Specifies the microversion. If the API supports microversions, the system returns the supported minimum microversion. If no, the system returns an empty value.
status	String	Specifies the version status. CURRENT: The version is the primary version. SUPPORTED: The version is an earlier version, but it is still supported. DEPRECATED: The version is a deprecated version, which may be deleted later.
updated	String	Specifies the version release time, which must be a UTC time. For example, the release time of v2 is 2014-06-28T12:20:21Z.
version	String	Specifies the microversion. If the API supports microversions, the system returns the supported minimum microversion. If no, the system returns an empty value.
links	Array of **LinksItem** objects	Specifies the API URL.

Table 5-245 LinksItem

Parameter	Type	Description
href	String	Specifies the corresponding shortcut link.

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name.

Status code: 400

Table 5-246 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-247 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-248 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-249 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

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

null
```

Example Responses

Status code: 200

OK

```
{
  "version" : {
    "id" : "v2",
    "links" : [ {
      "href" : "https://127.0.0.1/v2",
      "rel" : "self"
    } ],
    "min_version" : "",
    "status" : "CURRENT",
    "updated" : "2018-09-19T00:00:00Z",
    "version" : ""
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.7 Cloud Logs

5.7.1 Binding a Cloud Log to a Topic

Function

This API is used to bind a cloud log to a specified topic to record information such as the message sending status of the topic.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/topics/{topic_urn}/logtanks

Table 5-250 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the identifier, see **Querying Topics**.

Request Parameters

Table 5-251 Request header parameters

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

Table 5-252 Request body parameters

Parameter	Mandatory	Type	Description
log_group_id	Yes	String	Specifies the LTS log group ID.
log_stream_id	Yes	String	Specifies the LTS log stream ID.

Response Parameters

Status code: 201

Table 5-253 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
id	String	Specifies the resource identifier of the cloud log, which is unique.

Status code: 400

Table 5-254 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 403

Table 5-255 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-256 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-257 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Binding a cloud log to a topic

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/logtanks

{
  "log_group_id" : "d1b6c352-ee45-4196-9230-c74b7894e356",
  "log_stream_id" : "114960b4-1c26-48dd-93c2-4caae24aec49"
}
```

Example Responses

Status code: 201

OK

```
{
  "request_id" : "42f3b0cd1c894e87aa0275ce33305b21",
  "id" : "4ea80c8a68b840e19b4efd3a8f0d62ed"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Binding a cloud log to a topic

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class CreateLogtankSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateLogtankRequest request = new CreateLogtankRequest();
        request.withTopicUrn("{topic_urn}");
        CreateLogtankRequestBody body = new CreateLogtankRequestBody();
        body.withLogStreamId("114960b4-1c26-48dd-93c2-4caae24aec49");
        body.withLogGroupId("d1b6c352-ee45-4196-9230-c74b7894e356");
        request.withBody(body);
        try {
            CreateLogtankResponse response = client.createLogtank(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

Binding a cloud log to a topic

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateLogtankRequest()
        request.topic_urn = "{topic_urn}"
        request.body = CreateLogtankRequestBody(
            log_stream_id="114960b4-1c26-48dd-93c2-4caae24aec49",
            log_group_id="d1b6c352-ee45-4196-9230-c74b7894e356"
        )
        response = client.create_logtank(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

Binding a cloud log to a topic

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateLogtankRequest{}
    request.TopicUrn = "{topic_urn}"
    request.Body = &model.CreateLogtankRequestBody{
        LogStreamId: "114960b4-1c26-48dd-93c2-4caae24aec49",
        LogGroupId: "d1b6c352-ee45-4196-9230-c74b7894e356",
    }
    response, err := client.CreateLogtank(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
201	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.7.2 Querying Cloud Logs

Function

This API is used to query cloud logs bound to a specified topic.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics/{topic_urn}/logtanks

Table 5-258 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the identifier, see **Querying Topics**.

Request Parameters

Table 5-259 Request header parameters

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

Response Parameters

Status code: 200

Table 5-260 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
count	Integer	Specifies the number of cloud logs.
logtanks	Array of **LogtankItem** objects	Specifies cloud logs.

Table 5-261 LogtankItem

Parameter	Type	Description
id	String	Specifies the resource identifier of the cloud log, which is unique.
log_group_id	String	Specifies the LTS log group ID.
log_stream_id	String	Specifies the LTS log stream ID.
create_time	String	Specifies the creation time. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
update_time	String	Specifies the update time. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.

Status code: 400

Table 5-262 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 403

Table 5-263 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-264 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.

Parameter	Type	Description
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-265 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Querying cloud logs

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/logtanks

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "f5c05cb70d264ca58d1080202ceba284",
  "count" : 1,
  "logtanks" : [ {
    "id" : "4ea80c8a68b840e19b4efd3a8f0d62ed",
    "log_group_id" : "d1b6c352-ee45-4196-9230-c74b7894e356",
    "log_stream_id" : "114960b4-1c26-48dd-93c2-4caae24aec49",
    "create_time" : "2022-10-10T09:00:00Z",
    "update_time" : "2022-10-10T09:00:00Z"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized

Status Code	Description
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.7.3 Updating a Cloud Log

Function

This API is used to update a cloud log bound to a specified topic.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/topics/{topic_urn}/logtanks/{logtank_id}

Table 5-266 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the identifier, see **Querying Topics**.
logtank_id	Yes	String	Specifies the resource identifier of the cloud log, which is unique. To obtain the identifier, see **Querying Cloud Logs**.

Request Parameters

Table 5-267 Request header parameters

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

Table 5-268 Request body parameters

Parameter	Mandatory	Type	Description
log_group_id	Yes	String	Specifies the LTS log group ID.
log_stream_id	Yes	String	Specifies the LTS log stream ID.

Response Parameters

Status code: 200

Table 5-269 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
logtank	**LogtankItem** object	Specifies the cloud log.

Table 5-270 LogtankItem

Parameter	Type	Description
id	String	Specifies the resource identifier of the cloud log, which is unique.
log_group_id	String	Specifies the LTS log group ID.
log_stream_id	String	Specifies the LTS log stream ID.
create_time	String	Specifies the creation time. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.

Parameter	Type	Description
update_time	String	Specifies the update time. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.

Status code: 400

Table 5-271 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 403

Table 5-272 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-273 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-274 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Updating a cloud log

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/logtanks/
4ea80c8a68b840e19b4efd3a8f0d62ed

{
  "log_group_id" : "d1b6c352-ee45-4196-9230-c74b7894e356",
  "log_stream_id" : "114960b4-1c26-48dd-93c2-4caae24aec49"
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "761bd2658f92411b9b01ae13e7961b11",
  "logtank" : {
    "id" : "4ea80c8a68b840e19b4efd3a8f0d62ed",
    "log_group_id" : "d1b6c352-ee45-4196-9230-c74b7894e356",
    "log_stream_id" : "114960b4-1c26-48dd-93c2-4caae24aec49",
    "create_time" : "2022-10-10T09:00:00Z",
    "update_time" : "2022-10-10T09:00:00Z"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating a cloud log

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class UpdateLogtankSolution {
    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateLogtankRequest request = new UpdateLogtankRequest();
        request.withTopicUrn("{topic_urn}");
        request.withLogtankId("{logtank_id}");
        UpdateLogtankRequestBody body = new UpdateLogtankRequestBody();
        body.withLogStreamId("114960b4-1c26-48dd-93c2-4caae24aec49");
        body.withLogGroupId("d1b6c352-ee45-4196-9230-c74b7894e356");
        request.withBody(body);
        try {
            UpdateLogtankResponse response = client.updateLogtank(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 a cloud log

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateLogtankRequest()
        request.topic_urn = "{topic_urn}"
        request.logtank_id = "{logtank_id}"
        request.body = UpdateLogtankRequestBody(
            log_stream_id="114960b4-1c26-48dd-93c2-4caae24aec49",
            log_group_id="d1b6c352-ee45-4196-9230-c74b7894e356"
        )
        response = client.update_logtank(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 a cloud log

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateLogtankRequest{}
    request.TopicUrn = "{topic_urn}"
    request.LogtankId = "{logtank_id}"
    request.Body = &model.UpdateLogtankRequestBody{
        LogStreamId: "114960b4-1c26-48dd-93c2-4caae24aec49",
        LogGroupId: "d1b6c352-ee45-4196-9230-c74b7894e356",
    }
    response, err := client.UpdateLogtank(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.7.4 Unbinding a Cloud Log

Function

This API is used to unbind a cloud log from a specified topic.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/topics/{topic_urn}/logtanks/{logtank_id}

Table 5-275 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.

Parameter	Mandatory	Type	Description
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the identifier, see **Querying Topics**.
logtank_id	Yes	String	Specifies the resource identifier of the cloud log, which is unique. To obtain the identifier, see **Querying Cloud Logs**.

Request Parameters

Table 5-276 Request header parameters

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

Response Parameters

Status code: 200

Table 5-277 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.

Status code: 400

Table 5-278 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 403

Table 5-279 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-280 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-281 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Unbinding a cloud log

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/logtanks/
4ea80c8a68b840e19b4efd3a8f0d62ed

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "fe5cec832a5249169d433b6b0b5902a3"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.8 Message Filter Policies of Subscribers

5.8.1 Creating Message Filter Policies for a Subscriber

Function

This API is used to create message filter policies for subscribers.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/subscriptions/filter_polices

Table 5-282 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Request Parameters

Table 5-283 Request header parameters

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

Table 5-284 Request body parameters

Parameter	Mandatory	Type	Description
polices	Yes	Array of **polices** objects	Specifies subscriber policies to be processed in batches.

Table 5-285 polices

Parameter	Mandatory	Type	Description
subscription_urn	Yes	String	Specifies the resource identifier of the subscriber, which is unique. To obtain the resource identifier, see **Querying Subscribers**.
filter_polices	Yes	Array of **SubscriptionsFilterPolicy** objects	Specifies the message filter policies of a subscriber. The policy name must be unique.

Table 5-286 SubscriptionsFilterPolicy

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the filter policy name. **name** can contain 1 to 32 characters, including lowercase letters, digits, and underscores (*).* ***name*** *cannot start or end with an underscore, nor contain consecutive underscores.* ***name*** *cannot start with **smn**.*
string_equals	Yes	Array of strings	Specifies the string array for exact match. An array can contain 1 to 10 strings. The array content must be unique. The string cannot be **null** or an empty string "". A string can contain 1 to 32 characters, including letters, digits, and underscores (_).

Response Parameters

Status code: 200

Table 5-287 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique request ID.
batch_result	Array of **BatchResult** objects	Specifies the batch processing result.

Table 5-288 BatchResult

Parameter	Type	Description
code	String	Specifies the returned code.
message	String	Specifies the code message.
subscription_urn	String	Specifies the subscriber URN.

Status code: 400

Table 5-289 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-290 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-291 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-292 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Example Requests

Creating message filter policies for a subscriber

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/subscriptions/filter_polices

{
  "polices" : [ {
    "subscription_urn" : 
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
    "filter_polices" : [ {
      "name" : "alarm",
      "string_equals" : [ "os", "process" ]
    }, {
      "name" : "service",
      "string_equals" : [ "api", "db" ]
    } ]
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "be368401641b406d8c28a79915ba3589",
  "batch_result" : [ {
    "code" : "SMN.00011027",
    "message" : "Parameter: subscription_urn is invalid.",
    "subscription_urn" : 
"urn:smn:regionId:98386b0630aa41d990d4729497fcd7ba:test:90b22be1efab4cd6924703c5b228e59f"
  }, {
    "code" : "SMN.00011027",
    "message" : "Parameter: subscription_urn is invalid.",
    "subscription_urn" : 
"urn:smn:regionId:98386b0630aa41d990d4729497fcd7ba:test:c872b769e60d45f682f1da44eb4dbee3"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating message filter policies for a subscriber

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class BatchCreateSubscriptionsFilterPolicesSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateSubscriptionsFilterPolicesRequest request = new
BatchCreateSubscriptionsFilterPolicesRequest();
        BatchSubscriptionsFilterPolicesRequestBody body = new BatchSubscriptionsFilterPolicesRequestBody();
        List<String> listFilterPolicesStringEquals = new ArrayList<>();
        listFilterPolicesStringEquals.add("api");
        listFilterPolicesStringEquals.add("db");
        List<String> listFilterPolicesStringEquals1 = new ArrayList<>();
        listFilterPolicesStringEquals1.add("os");
        listFilterPolicesStringEquals1.add("process");
        List<SubscriptionsFilterPolicy> listPolicesFilterPolices = new ArrayList<>();
        listPolicesFilterPolices.add(
            new SubscriptionsFilterPolicy()
                .withName("alarm")
                .withStringEquals(listFilterPolicesStringEquals1)
        );
        listPolicesFilterPolices.add(
            new SubscriptionsFilterPolicy()
                .withName("service")
                .withStringEquals(listFilterPolicesStringEquals)
        );
        List<BatchSubscriptionsFilterPolicesRequestBodyPolices> listbodyPolices = new ArrayList<>();
        listbodyPolices.add(
            new BatchSubscriptionsFilterPolicesRequestBodyPolices()
                .withSubscriptionUrn("urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e77
8e84408e44058e6cbc6d3c377837")
                .withFilterPolices(listPolicesFilterPolices)
        );
        body.withPolices(listbodyPolices);
        request.withBody(body);
        try {
            BatchCreateSubscriptionsFilterPolicesResponse response =
client.batchCreateSubscriptionsFilterPolices(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

Creating message filter policies for a subscriber

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchCreateSubscriptionsFilterPolicesRequest()
        listStringEqualsFilterPolices = [
            "api",
            "db"
        ]
        listStringEqualsFilterPolices1 = [
            "os",
            "process"
        ]
        listFilterPolicesPolices = [
            SubscriptionsFilterPolicy(
                name="alarm",
                string_equals=listStringEqualsFilterPolices1
            ),
            SubscriptionsFilterPolicy(
                name="service",
                string_equals=listStringEqualsFilterPolices
            )
        ]
        listPolicesbody = [
            BatchSubscriptionsFilterPolicesRequestBodyPolices(

subscription_urn="urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058
e6cbc6d3c377837",
                filter_polices=listFilterPolicesPolices
            )
        ]
        request.body = BatchSubscriptionsFilterPolicesRequestBody(
            polices=listPolicesbody
        )
        response = client.batch_create_subscriptions_filter_polices(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

Creating message filter policies for a subscriber

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateSubscriptionsFilterPolicesRequest{}
    var listStringEqualsFilterPolices = []string{
        "api",
        "db",
    }
    var listStringEqualsFilterPolices1 = []string{
        "os",
        "process",
    }
    var listFilterPolicesPolices = []model.SubscriptionsFilterPolicy{
        {
            Name: "alarm",
            StringEquals: listStringEqualsFilterPolices1,
        },
        {
            Name: "service",
            StringEquals: listStringEqualsFilterPolices,
        },
    }
    var listPolicesbody = []model.BatchSubscriptionsFilterPolicesRequestBodyPolices{
        {
            SubscriptionUrn: 
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
            FilterPolices: listFilterPolicesPolices,
        },
    }
    request.Body = &model.BatchSubscriptionsFilterPolicesRequestBody{
        Polices: listPolicesbody,
    }
    response, err := client.BatchCreateSubscriptionsFilterPolices(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.8.2 Updating Message Filter Policies of Subscribers

Function

This API is used to update the message filter policies of subscribers.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/subscriptions/filter_polices

Table 5-293 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Request Parameters

Table 5-294 Request header parameters

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

Table 5-295 Request body parameters

Parameter	Mandatory	Type	Description
polices	Yes	Array of **polices** objects	Specifies subscriber policies to be processed in batches.

Table 5-296 polices

Parameter	Mandatory	Type	Description
subscription_urn	Yes	String	Specifies the resource identifier of the subscriber, which is unique. To obtain the resource identifier, see **Querying Subscribers**.
filter_polices	Yes	Array of **SubscriptionsFilterPolicy** objects	Specifies the message filter policies of a subscriber. The policy name must be unique.

Table 5-297 SubscriptionsFilterPolicy

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the filter policy name. **name** can contain 1 to 32 characters, including lowercase letters, digits, and underscores (*). **name** cannot start or end with an underscore, nor contain consecutive underscores.* ***name*** *cannot start with **smn**.*
string_equals	Yes	Array of strings	Specifies the string array for exact match. An array can contain 1 to 10 strings. The array content must be unique. The string cannot be **null** or an empty string "". A string can contain 1 to 32 characters, including letters, digits, and underscores (_).

Response Parameters

Status code: 200

Table 5-298 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique request ID.
batch_result	Array of **BatchResult** objects	Specifies the batch processing result.

Table 5-299 BatchResult

Parameter	Type	Description
code	String	Specifies the returned code.
message	String	Specifies the code message.
subscription_urn	String	Specifies the subscriber URN.

Status code: 400

Table 5-300 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-301 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-302 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-303 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Example Requests

Updating message filter policies of a subscriber

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/subscriptions/filter_polices

{
  "polices" : [ {
    "subscription_urn" : 
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
    "filter_polices" : [ {
      "name" : "alarm",
      "string_equals" : [ "os", "process" ]
    }, {
      "name" : "service",
      "string_equals" : [ "api", "db" ]
    } ]
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "be368401641b406d8c28a79915ba3589",
  "batch_result" : [ {
    "code" : "SMN.00011027",
    "message" : "Parameter: subscription_urn is invalid.",
    "subscription_urn" : 
"urn:smn:regionId:98386b0630aa41d990d4729497fcd7ba:test:90b22be1efab4cd6924703c5b228e59f"
  }, {
    "code" : "SMN.00011027",
    "message" : "Parameter: subscription_urn is invalid.",
    "subscription_urn" : 
"urn:smn:regionId:98386b0630aa41d990d4729497fcd7ba:test:c872b769e60d45f682f1da44eb4dbee3"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating message filter policies of a subscriber

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class BatchUpdateSubscriptionsFilterPolicesSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchUpdateSubscriptionsFilterPolicesRequest request = new 
BatchUpdateSubscriptionsFilterPolicesRequest();
        BatchSubscriptionsFilterPolicesRequestBody body = new BatchSubscriptionsFilterPolicesRequestBody();
        List<String> listFilterPolicesStringEquals = new ArrayList<>();
        listFilterPolicesStringEquals.add("api");
        listFilterPolicesStringEquals.add("db");
        List<String> listFilterPolicesStringEquals1 = new ArrayList<>();
        listFilterPolicesStringEquals1.add("os");
        listFilterPolicesStringEquals1.add("process");
        List<SubscriptionsFilterPolicy> listPolicesFilterPolices = new ArrayList<>();
        listPolicesFilterPolices.add(
            new SubscriptionsFilterPolicy()
                .withName("alarm")
                .withStringEquals(listFilterPolicesStringEquals1)
        );
        listPolicesFilterPolices.add(
            new SubscriptionsFilterPolicy()
                .withName("service")
                .withStringEquals(listFilterPolicesStringEquals)
        );
        List<BatchSubscriptionsFilterPolicesRequestBodyPolices> listbodyPolices = new ArrayList<>();
        listbodyPolices.add(
            new BatchSubscriptionsFilterPolicesRequestBodyPolices()
                .withSubscriptionUrn("urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e77
8e84408e44058e6cbc6d3c377837")
                .withFilterPolices(listPolicesFilterPolices)
        );
        body.withPolices(listbodyPolices);
        request.withBody(body);
        try {
            BatchUpdateSubscriptionsFilterPolicesResponse response = 
client.batchUpdateSubscriptionsFilterPolices(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 message filter policies of a subscriber

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchUpdateSubscriptionsFilterPolicesRequest()
        listStringEqualsFilterPolices = [
            "api",
            "db"
        ]
        listStringEqualsFilterPolices1 = [
            "os",
            "process"
        ]
        listFilterPolicesPolices = [
            SubscriptionsFilterPolicy(
                name="alarm",
                string_equals=listStringEqualsFilterPolices1
            ),
            SubscriptionsFilterPolicy(
                name="service",
                string_equals=listStringEqualsFilterPolices
            )
        ]
        listPolicesbody = [
            BatchSubscriptionsFilterPolicesRequestBodyPolices(

subscription_urn="urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058
e6cbc6d3c377837",
                filter_polices=listFilterPolicesPolices
            )
        ]
        request.body = BatchSubscriptionsFilterPolicesRequestBody(
            polices=listPolicesbody
        )
        response = client.batch_update_subscriptions_filter_polices(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 message filter policies of a subscriber

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchUpdateSubscriptionsFilterPolicesRequest{}
    var listStringEqualsFilterPolices = []string{
        "api",
        "db",
    }
    var listStringEqualsFilterPolices1 = []string{
        "os",
        "process",
    }
    var listFilterPolicesPolices = []model.SubscriptionsFilterPolicy{
        {
            Name: "alarm",
            StringEquals: listStringEqualsFilterPolices1,
        },
        {
            Name: "service",
            StringEquals: listStringEqualsFilterPolices,
        },
    }
    var listPolicesbody = []model.BatchSubscriptionsFilterPolicesRequestBodyPolices{
        {
            SubscriptionUrn:
"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
            FilterPolices: listFilterPolicesPolices,
        },
    }
    request.Body = &model.BatchSubscriptionsFilterPolicesRequestBody{
        Polices: listPolicesbody,
    }
    response, err := client.BatchUpdateSubscriptionsFilterPolices(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.8.3 Deleting Message Filter Policies of Subscribers

Function

This API is used to delete message filter policies of subscribers.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/subscriptions/filter_polices

Table 5-304 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Request Parameters

Table 5-305 Request header parameters

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

Table 5-306 Request body parameters

Parameter	Mandatory	Type	Description
subscription_urns	Yes	Array of strings	Specifies the resource identifier of the subscriber, which is unique. To obtain the resource identifier, see **Querying Subscribers**.

Response Parameters

Status code: 200

Table 5-307 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique request ID.
batch_result	Array of **BatchResult** objects	Specifies the batch processing result.

Table 5-308 BatchResult

Parameter	Type	Description
code	String	Specifies the returned code.
message	String	Specifies the code message.
subscription_urn	String	Specifies the subscriber URN.

Status code: 400

Table 5-309 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 5-310 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 5-311 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 5-312 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a policy for filtering messages

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/subscriptions/filter_polices

{
  "subscription_urns" :
[ "urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c3778
37" ]
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Deleting a policy for filtering messages

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class BatchDeleteSubscriptionsFilterPolicesSolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchDeleteSubscriptionsFilterPolicesRequest request = new
BatchDeleteSubscriptionsFilterPolicesRequest();
        BatchDeleteSubscriptionsFilterPolicesRequestBody body = new
BatchDeleteSubscriptionsFilterPolicesRequestBody();
        List<String> listbodySubscriptionUrns = new ArrayList<>();

listbodySubscriptionUrns.add("urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e8
4408e44058e6cbc6d3c377837");
        body.withSubscriptionUrns(listbodySubscriptionUrns);
        request.withBody(body);
        try {
            BatchDeleteSubscriptionsFilterPolicesResponse response =
client.batchDeleteSubscriptionsFilterPolices(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

Deleting a policy for filtering messages

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
    client = SmnClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(SmnRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteSubscriptionsFilterPolicesRequest()
        listSubscriptionUrnsbody = [

"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
"
        ]
        request.body = BatchDeleteSubscriptionsFilterPolicesRequestBody(
            subscription_urns=listSubscriptionUrnsbody
        )
        response = client.batch_delete_subscriptions_filter_polices(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

Deleting a policy for filtering messages

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteSubscriptionsFilterPolicesRequest{}
    var listSubscriptionUrnsbody = []string{

"urn:smn:regionId:762bdb3251034f268af0e395c53ea09b:test_topic_v1:2e778e84408e44058e6cbc6d3c377837
",
    }
    request.Body = &model.BatchDeleteSubscriptionsFilterPolicesRequestBody{
        SubscriptionUrns: listSubscriptionUrnsbody,
    }
    response, err := client.BatchDeleteSubscriptionsFilterPolices(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	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.9 Notification Policies

5.9.1 Creating a Notification Policy

Function

This API is used to create a notification policy.

Calling Method

For details, see **Calling APIs**.

URI

POST /v2/{project_id}/notifications/topics/{topic_urn}/notify-policy

Table 5-313 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-314 Request header parameters

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

Table 5-315 Request body parameters

Parameter	Mandatory	Type	Description
protocol	Yes	String	Specifies the notification policy type. Only the voice notification policies are supported.
polling	Yes	Array of **PollingPolicyRequest** objects	Specifies the subscription endpoint in a polling notification policy.

Table 5-316 PollingPolicyRequest

Parameter	Mandatory	Type	Description
order	Yes	Integer	Specifies the sequence number of the subscription endpoint being polled.

Parameter	Mandatory	Type	Description
subscription_urns	Yes	Array of strings	Specifies the URN list of subscription endpoints.

Response Parameters

Status code: 201

Table 5-317 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique request ID.
id	String	Specifies the notification policy ID.

Status code: 400

Table 5-318 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 403

Table 5-319 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-320 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-321 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Creating a polling notification policy.

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/notify-policy

{
  "protocol" : "callnotify",
  "polling" : [ {
    "order" : 1,
    "subscription_urns" :
[ "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222" ]
  }, {
    "order" : 2,
    "subscription_urns" :
[ "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222" ]
  } ]
}
```

Example Responses

Status code: 201

Notification policy created.

```
{
  "request_id" : "f6832d5dd17f4090b0e29c6f9a026b17",
  "id" : "7824d9c2f4e94fdfabd9fcdd4da0d564"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a polling notification policy.

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;

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

public class CreateNotifyPolicySolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateNotifyPolicyRequest request = new CreateNotifyPolicyRequest();
        request.withTopicUrn("{topic_urn}");
        NotifyPolicyRequestBody body = new NotifyPolicyRequestBody();
        List<String> listPollingSubscriptionUrns = new ArrayList<>();

listPollingSubscriptionUrns.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c4
78aa2ef6b16153b521f");

listPollingSubscriptionUrns.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c4
78aa2ef6b16153b5222");
        List<String> listPollingSubscriptionUrns1 = new ArrayList<>();

listPollingSubscriptionUrns1.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c
478aa2ef6b16153b521f");

listPollingSubscriptionUrns1.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c
478aa2ef6b16153b5222");
        List<PollingPolicyRequest> listbodyPolling = new ArrayList<>();
        listbodyPolling.add(
            new PollingPolicyRequest()
                .withOrder(1)
                .withSubscriptionUrns(listPollingSubscriptionUrns1)
        );
        listbodyPolling.add(
            new PollingPolicyRequest()
                .withOrder(2)
                .withSubscriptionUrns(listPollingSubscriptionUrns)
        );
        body.withPolling(listbodyPolling);
        body.withProtocol(NotifyPolicyRequestBody.ProtocolEnum.fromValue("callnotify"));
        request.withBody(body);
        try {
            CreateNotifyPolicyResponse response = client.createNotifyPolicy(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

Creating a polling notification policy.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateNotifyPolicyRequest()
        request.topic_urn = "{topic_urn}"
        listSubscriptionUrnsPolling = [
            "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
            "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222"
        ]
        listSubscriptionUrnsPolling1 = [
            "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
            "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222"
        ]
        listPollingbody = [
            PollingPolicyRequest(
                order=1,
                subscription_urns=listSubscriptionUrnsPolling1
            ),
            PollingPolicyRequest(
                order=2,
                subscription_urns=listSubscriptionUrnsPolling
            )
        ]
        request.body = NotifyPolicyRequestBody(
            polling=listPollingbody,
            protocol="callnotify"
        )
        response = client.create_notify_policy(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

Creating a polling notification policy.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNotifyPolicyRequest{}
    request.TopicUrn = "{topic_urn}"
    var listSubscriptionUrnsPolling = []string{
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222",
    }
    var listSubscriptionUrnsPolling1 = []string{
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222",
    }
    var listPollingbody = []model.PollingPolicyRequest{
        {
            Order: int32(1),
            SubscriptionUrns: listSubscriptionUrnsPolling1,
        },
        {
            Order: int32(2),
            SubscriptionUrns: listSubscriptionUrnsPolling,
        },
    }
    request.Body = &model.NotifyPolicyRequestBody{
        Polling: listPollingbody,
        Protocol: model.GetNotifyPolicyRequestBodyProtocolEnum().CALLNOTIFY,
    }
    response, err := client.CreateNotifyPolicy(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
201	Notification policy created.
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.9.2 Querying a Notification Policy

Function

This API is used to query a notification policy.

Calling Method

For details, see **Calling APIs**.

URI

GET /v2/{project_id}/notifications/topics/{topic_urn}/notify-policy

Table 5-322 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Request Parameters

Table 5-323 Request header parameters

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

Response Parameters

Status code: 200

Table 5-324 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
id	String	Specifies the notification policy ID.
protocol	String	Specifies the notification policy type. Only the voice notification policies are supported.
polling	Array of **PollingPolicyResponse** objects	Specifies the subscription endpoint in a polling notification policy.

Table 5-325 PollingPolicyResponse

Parameter	Type	Description
order	Integer	Specifies the sequence number of the subscription endpoint being polled.
subscriptions	Array of **PollingPolicySubscriptionDetails** objects	Specifies the URN list of subscription endpoints.

Table 5-326 PollingPolicySubscriptionDetails

Parameter	Type	Description
subscription_urn	String	Specifies the sequence number of the subscription endpoint being polled.
endpoint	String	Specifies the URN list of subscription endpoints.
remark	String	Specifies the remark.
status	Integer	Specifies the subscription status. **0** indicates the subscription has not been confirmed. **1** indicates that the subscription has been confirmed. **3** indicates that the subscription has been canceled.

Status code: 403

Table 5-327 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-328 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Querying a notification policy.

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/notify-policy
```

Example Responses

Status code: 200

Query succeeded.

```
{
  "request_id" : "2f97a871015348529bccc070417ef16f",
  "protocol" : "callnotify",
  "id" : "9d1ff67d195d44c1aed4edd92cc44cd1",
  "polling" : [ {
    "order" : 1,
    "subscriptions" : [ {
      "subscription_urn" : 
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
      "endpoint" : "13612345611",
      "remark" : "",
      "status" : 1
    }, {
      "subscription_urn" : 
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222",
      "endpoint" : "13612345622",
      "remark" : "",
      "status" : 1
    } ]
  }, {
    "order" : 2,
    "subscriptions" : [ {
      "subscription_urn" : 
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
      "endpoint" : "13612345633",
      "remark" : "",
      "status" : 1
    }, {
      "subscription_urn" : 
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222",
      "endpoint" : "13612345644",
      "remark" : "",
      "status" : 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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class ShowNotifyPolicySolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowNotifyPolicyRequest request = new ShowNotifyPolicyRequest();
        request.withTopicUrn("{topic_urn}");
        try {
            ShowNotifyPolicyResponse response = client.showNotifyPolicy(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 huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowNotifyPolicyRequest()
        request.topic_urn = "{topic_urn}"
        response = client.show_notify_policy(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"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowNotifyPolicyRequest{}
    request.TopicUrn = "{topic_urn}"
    response, err := client.ShowNotifyPolicy(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	Query succeeded.
403	Unauthorized
500	Internal Server Error

Error Codes

See **Error Codes**.

5.9.3 Modifying a Notification Policy

Function

This API is used to modify a notification policy. You need to include all parameters required in the request body.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v2/{project_id}/notifications/topics/{topic_urn}/notify-policy/{notify_policy_id}

Table 5-329 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.

Parameter	Mandatory	Type	Description
notify_policy_id	Yes	String	Specifies the notification policy ID.

Request Parameters

Table 5-330 Request header parameters

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

Table 5-331 Request body parameters

Parameter	Mandatory	Type	Description
protocol	Yes	String	Specifies the notification policy type. Only the voice notification policies are supported.
polling	Yes	Array of **PollingPolicyRequest** objects	Specifies the subscription endpoint in a polling notification policy.

Table 5-332 PollingPolicyRequest

Parameter	Mandatory	Type	Description
order	Yes	Integer	Specifies the sequence number of the subscription endpoint being polled.
subscription_urns	Yes	Array of strings	Specifies the URN list of subscription endpoints.

Response Parameters

Status code: 200

Table 5-333 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 5-334 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 403

Table 5-335 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-336 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-337 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Modifying a notification policy.

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/notify-policy/
7824d9c2f4e94fdfabd9fcdd4da0d564

{
  "protocol" : "callnotify",
  "polling" : [ {
    "order" : "1",
    "subscription_urns" :
[ "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222" ]
  }, {
    "order" : "2",
    "subscription_urns" :
[ "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222" ]
  } ]
}
```

Example Responses

Status code: 200

Notification policy modified.

```
{
  "request_id" : "f6832d5dd17f4090b0e29c6f9a026b17"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying a notification policy.

```
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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;
import java.util.List;
import java.util.ArrayList;

public class UpdateNotifyPolicySolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateNotifyPolicyRequest request = new UpdateNotifyPolicyRequest();
        request.withTopicUrn("{topic_urn}");
        request.withNotifyPolicyId("{notify_policy_id}");
        NotifyPolicyRequestBody body = new NotifyPolicyRequestBody();
        List<String> listPollingSubscriptionUrns = new ArrayList<>();

listPollingSubscriptionUrns.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c4
78aa2ef6b16153b521f");

listPollingSubscriptionUrns.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c4
78aa2ef6b16153b5222");
        List<String> listPollingSubscriptionUrns1 = new ArrayList<>();

listPollingSubscriptionUrns1.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c
478aa2ef6b16153b521f");

listPollingSubscriptionUrns1.add("urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c
478aa2ef6b16153b5222");
        List<PollingPolicyRequest> listbodyPolling = new ArrayList<>();
        listbodyPolling.add(
            new PollingPolicyRequest()
                .withOrder(1)
                .withSubscriptionUrns(listPollingSubscriptionUrns1)
        );
        listbodyPolling.add(
            new PollingPolicyRequest()
                .withOrder(2)
                .withSubscriptionUrns(listPollingSubscriptionUrns)
        );
        body.withPolling(listbodyPolling);
        body.withProtocol(NotifyPolicyRequestBody.ProtocolEnum.fromValue("callnotify"));
        request.withBody(body);
        try {
            UpdateNotifyPolicyResponse response = client.updateNotifyPolicy(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

Modifying a notification policy.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateNotifyPolicyRequest()
        request.topic_urn = "{topic_urn}"
        request.notify_policy_id = "{notify_policy_id}"
        listSubscriptionUrnsPolling = [

"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",

"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222"
        ]
        listSubscriptionUrnsPolling1 = [

"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",

"urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222"
        ]
        listPollingbody = [
            PollingPolicyRequest(
                order=1,
                subscription_urns=listSubscriptionUrnsPolling1
            ),
            PollingPolicyRequest(
                order=2,
                subscription_urns=listSubscriptionUrnsPolling
            )
        ]
        request.body = NotifyPolicyRequestBody(
            polling=listPollingbody,
            protocol="callnotify"
        )
        response = client.update_notify_policy(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

Modifying a notification policy.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNotifyPolicyRequest{}
    request.TopicUrn = "{topic_urn}"
    request.NotifyPolicyId = "{notify_policy_id}"
    var listSubscriptionUrnsPolling = []string{
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222",
    }
    var listSubscriptionUrnsPolling1 = []string{
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b521f",
        "urn:smn:regionId:240489644ad14ace8e022869d34411a4:eee:45c05e14762c478aa2ef6b16153b5222",
    }
    var listPollingbody = []model.PollingPolicyRequest{
        {
            Order: int32(1),
            SubscriptionUrns: listSubscriptionUrnsPolling1,
        },
        {
            Order: int32(2),
            SubscriptionUrns: listSubscriptionUrnsPolling,
        },
    }
    request.Body = &model.NotifyPolicyRequestBody{
        Polling: listPollingbody,
        Protocol: model.GetNotifyPolicyRequestBodyProtocolEnum().CALLNOTIFY,
    }
    response, err := client.UpdateNotifyPolicy(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	Notification policy modified.
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

5.9.4 Deleting a Notification Policy

Function

This API is used to delete a notification policy.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v2/{project_id}/notifications/topics/{topic_urn}/notify-policy/{notify_policy_id}

Table 5-338 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining the Project ID**.

Parameter	Mandatory	Type	Description
topic_urn	Yes	String	Specifies the resource identifier of the topic, which is unique. To obtain the resource identifier, see **Querying Topics**.
notify_policy_id	Yes	String	Specifies the notification policy ID.

Request Parameters

Table 5-339 Request header parameters

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

Response Parameters

Status code: 200

Table 5-340 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 403

Table 5-341 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 404

Table 5-342 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Status code: 500

Table 5-343 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the unique ID of a request.
error_code	String	Specifies the error code.
error_msg	String	Specifies the error message.

Example Requests

Deleting a notification policy.

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/
urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2/notify-policy/
7824d9c2f4e94fdfabd9fcdd4da0d564
```

Example Responses

Status code: 200

Notification policy deleted.

```
{
  "request_id" : "f6832d5dd17f4090b0e29c6f9a026b17"
}
```

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.smn.v2.region.SmnRegion;
import com.huaweicloud.sdk.smn.v2.*;
import com.huaweicloud.sdk.smn.v2.model.*;


public class DeleteNotifyPolicySolution {

    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");
        String projectId = "{project_id}";

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

        SmnClient client = SmnClient.newBuilder()
                .withCredential(auth)
                .withRegion(SmnRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteNotifyPolicyRequest request = new DeleteNotifyPolicyRequest();
        request.withTopicUrn("{topic_urn}");
        request.withNotifyPolicyId("{notify_policy_id}");
        try {
            DeleteNotifyPolicyResponse response = client.deleteNotifyPolicy(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 huaweicloudsdksmn.v2.region.smn_region import SmnRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksmn.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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteNotifyPolicyRequest()
        request.topic_urn = "{topic_urn}"
        request.notify_policy_id = "{notify_policy_id}"
        response = client.delete_notify_policy(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"
    smn "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/smn/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")
    projectId := "{project_id}"

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

    client := smn.NewSmnClient(
        smn.SmnClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteNotifyPolicyRequest{}
    request.TopicUrn = "{topic_urn}"
    request.NotifyPolicyId = "{notify_policy_id}"
    response, err := client.DeleteNotifyPolicy(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	Notification policy deleted.
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

6 Examples

6.1 Example 1: Publishing Messages to a Topic

Scenarios

This section describes how to publish messages by calling APIs. For details, see **Calling APIs**.

Involved APIs

- **Obtaining a User Token Through Password Authentication**
- **Creating a Topic**
- **Adding a Subscription**
- **Publishing a Message**

Procedure

1. Obtain the token by following instructions in **3.2 Authentication**.
2. Create a topic.
   a. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics**.
   b. Add **X-Auth-Token** to the request header.
   c. Specify the following parameters in the request body:
   For details about the parameters, see **Creating a Topic**.

```
{
    "name": "test_topic_v2", //(Mandatory) Topic name
    "display_name": "test", //(Optional) Display name
    "enterprise_project_id": "0" //Enterprise project ID (mandatory in the enterprise project
scenario)
}
```

   Example response

```
{
    "request_id": "6a63a18b8bab40ffb71ebd9cb80d0085",
   "topic_urn": "urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2"
}
```

3. Add a subscription.

   a. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/subscriptions**.

      Set **topic_urn** based on the response in 2.c.

   b. Add **X-Auth-Token** to the request header.

   c. Specify the following parameters in the request body:

```
{
   "protocol": "email", //(Mandatory) Subscription protocol
   "endpoint": "xxx@xx.com", //(Mandatory) Email address
   "remark": "Subscription" //(Optional) Description
}
```

      Example response

```
{
   "subscriptions_result": [{
      "request_id": "126c17f805934e5eadf647b0860c7a0a",
      "subscription_urn": "urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:test_topic_v2",
      "http_code": 201}]
}
```

      NOTE

      You can add multiple subscriptions to a topic.

   d. The subscription endpoint receives a confirmation message, and you must confirm the subscription in the email.

4. Publish a message.

   a. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/publish**.

   b. Add **X-Auth-Token** to the request header.

   c. Specify the following parameters in the request body:

```
{
  "subject":"test message v2", //(Optional) Message subject, which is displayed in email
messages
  "message": "Message test message v2." //(Mandatory) Message to be published
}
```

      Example response

```
{
   "request_id": "6a63a18b8bab40ffb71ebd9cb80d0085",
   "message_id": "992bbd32819448af95f83814cb3c7454"
}
```

      NOTE

      Only confirmed subscription endpoints can receive the message you publish.

6.2 Example 2: Publishing a Message Using a Message Template

Scenarios

This section describes how to publish a message using a message template. For details, see **Calling APIs**.

Involved APIs

- **Obtaining a User Token Through Password Authentication**
- **Creating a Topic**
- **Adding a Subscription**
- **Creating a Message Template**
- **Publishing a Message**

Procedure

1. Obtain the token by following instructions in **3.2 Authentication**.
2. Create a topic.
   a. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics**.
   b. Add **X-Auth-Token** to the request header.
   c. Specify the following parameters in the request body:

      For details about the parameters, see **Creating a Topic**.

      ```
      {
          "name": "publish_by_template", //(Mandatory) Topic name
          "display_name": "test", //(Optional) Display name
          "enterprise_project_id": "0" //Enterprise project ID (mandatory in the enterprise project
      scenario)
      }
      ```

      Example response

      ```
      {
          "request_id": "a8f8161449564f569c19e2a9e3be25c9",
          "topic_urn": "urn:smn:br-iaas-
      odin1:0605767e0a00d5762fe7c00d0e33345c:publish_by_template"
      }
      ```
3. Add a subscription.
   a. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/subscriptions**.

      Set **topic_urn** based on the response in **2.c**.
   b. Add **X-Auth-Token** to the request header.
   c. Specify the following parameters in the request body:

      ```
      {
          "protocol": "sms", //(Mandatory) Subscription protocol
          "endpoint": "86137****3629", //(Mandatory) Email address
          "remark": "Subscription" //(Optional) Description
      }
      ```

      Example response

      ```
      {
          "subscriptions_result":[{
              "request_id":"126c17f805934e5eadf647b0860c7a0a",
              "subscription_urn":"urn:smn:br-iaas-
      odin1:0605767e0a00d5762fe7c00d0e33345c:publish_by_template:3e5ac4544a1947b7b0d9f743fe
      52bdbd",
              "http_code":201}]
      }
      ```

      NOTE

      You can add multiple subscriptions to a topic.
   d. The subscription endpoint receives a confirmation message, and you must confirm the subscription in the email.

4. Create a message template.

   Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/message_template**.

5. Add **X-Auth-Token** to the request header.

   a. Specify the following parameters in the request body:

      ```
      {
          "message_template_name": "sms_template_test",
          "locale": "zh-cn",
          "content": "{content}",
          "protocol": "sms"
      }
      ```

      Example response

      ```
      {
          "request_id": "e67228ac92324a58980e922ca600f8ef",
          "message_template_id": "cdb113996c5d49149fb97af36d9801ce"
      }
      ```

   b. The subscription endpoint receives a confirmation message and you must confirm the subscription with your phone.

6. Use the message template to publish a message.

   a. Send request **POST https://{SMN_Endpoint}/v2/{project_id}/notifications/topics/{topic_urn}/publish**.

   b. Add **X-Auth-Token** to the request header.

   c. Specify the following parameters in the request body:

      ```
      {
          "message_template_name": "sms_template_test",
          "tags": {
          "content": "hello test"
          },
          "subject": "template message test"
      }
      ```

      Example response

      ```
      {
          "message_id": "5da965fd9e2c4036a0542317d944af83",
          "request_id": "c6be4517f985470e8448e42f9d2ec5a2"
      }
      ```

      NOTE

      Only confirmed subscription endpoints can receive the message you publish.

7 Permissions Policies and Supported Actions

7.1 Permissions Policies and Supported Actions

This section describes fine-grained permissions management for your SMN resources. If your Huawei Cloud account does not need individual IAM users, you can skip over this section.

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

You can grant users permissions by using **roles** and **policies**. Roles: A type of coarse-grained authorization mechanism that defines service-level permissions based on user responsibilities. There are only a limited number of roles for granting permissions to users. Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization and secure access control.

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 query ECSs using an API, the user must have been granted permissions that allow the **ecs:servers:list** action.

Supported Actions

IAM provides system-defined policies that can be directly used. You can also create custom policies to supplement system-defined policies for more refined access control. Actions supported by policies are specific to APIs. Common concepts related to policies include:

- Permissions: statements in a policy that allow or deny certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.
- Actions: specific operations that are allowed or denied.
- Related actions: actions on which a specific action depends to take effect. When assigning permissions for the action to a user, you also need to assign permissions for the dependent actions.
- IAM or enterprise projects: type of projects for which an action will take effect. Policies that contain actions for both IAM and enterprise projects can be used and applied for both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM. Administrators can check whether an action supports IAM projects or enterprise projects in the action list. "√" indicates that the action supports the project and "×" indicates that the action does not support the project. For details about the differences between IAM and enterprise management, see **What Are the Differences Between IAM and Enterprise Management?**

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

- **7.2 Topic Operations**, including actions supported by topic management APIs, such as APIs for creating, querying, updating, and deleting topics
- **7.3 Template Operations**, including actions supported by message template management APIs, such as the APIs for creating, querying, updating, and deleting message templates
- **7.6 Application Operations (Deprecated)**, including actions supported by application management APIs, such as the APIs for creating, querying, updating, and deleting applications
- **7.4 Tag Operations**, including actions supported by TMS APIs, such as the APIs for querying resources, adding a resource tag, adding or deleting resource tags in batches, and deleting a resource a tag
- **7.5 Message Publishing**, including actions supported by message publishing management APIs, such as the message publishing API
- **7.7 Sending an Application Message (Deprecated)**, including actions supported by direct application messaging management API, such as the API for sending an application message

7.2 Topic Operations

Permission	API	Action	IAM Project	Enterprise Project
Creating a topic	POST /v2/{project_id}/notifications/topics	smn:topic:create	√	√
Querying a topic	GET /v2/{project_id}/notifications/topics	smn:topic:list	√	√
Updating a topic	PUT /v2/{project_id}/notifications/topics/{topic_urn}	smn:topic:update	√	√
Deleting a topic	DELETE /v2/{project_id}/notifications/topics/{topic_urn}	smn:topic:delete	√	√

7.3 Template Operations

Permission	API	Action	IAM Project	Enterprise Project
Creating a message template	POST /v2/{project_id}/notifications/message_template	smn:template:create	√	×
Querying a message template	GET /v2/{project_id}/notifications/message_template	smn:template:list	√	×

Permission	API	Action	IAM Project	Enterprise Project
Modifying a message template	PUT /v2/{project_id}/notifications/message_template/{message_template_id}	smn:template:update	√	×
Deleting a message template	DELETE /v2/{project_id}/notifications/message_template/{message_template_id}	smn:template:delete	√	×

7.4 Tag Operations

Permission	API	Action	IAM Project	Enterprise Project
Adding a resource tag	POST /v2/{project_id}/{resource_type}/{resource_id}/tags	smn:tag:create	√	√
Updating a tag	-	smn:tag:update	√	√
Deleting a resource tag	DELETE /v2/{project_id}/{resource_type}/{resource_id}/tags/{key}	smn:tag:delete	√	√
Querying resource tags	GET /v2/{project_id}/{resource_type}/{resource_id}/tags	smn:tag:list	√	√

7.5 Message Publishing

Permission	API	Action	IAM Project	Enterprise Project
Publishing a message	POST /v2/{project_id}/notifications/topics/{topic_urn}/publish	smn:topic:publish	√	×

7.6 Application Operations (Deprecated)

Permission	API	Action	IAM Project	Enterprise Project
Creating a platform application	POST /v2/{project_id}/notifications/applications	smn:application:create	√	√
Querying platform applications	GET /v2/{project_id}/notifications/applications?offset={offset}&limit={limit}&name={name}&platform={platform}	smn:application:list	√	√
Updating a platform application	PUT /v2/{project_id}/notifications/applications/{application_urn}	smn:application:update	√	√
Deleting a platform application	DELETE /v2/{project_id}/notifications/applications/{application_urn}	smn:application:delete	√	√

7.7 Sending an Application Message (Deprecated)

Permission	API	Action	IAM Project	Enterprise Project
Sending an application message	POST /v2/{project_id}/notifications/endpoints/{endpoint_urn}/publish	smn:application:publish	√	√

8 Public Parameters

8.1 Obtaining a Project ID

Scenarios

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

- **Obtain the Project ID by Calling an API**
- **Obtain the Project ID from the Console**

Obtain the Project ID by Calling an API

You can obtain a project ID by calling the API used to **query projects based on specified criteria**.

The API used to obtain a project ID is GET https://{Endpoint}/v3/projects. {Endpoint} is the IAM endpoint and can be obtained from **Regions and Endpoints**. For details about API authentication, see **3.2 Authentication**.

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

```
{
    "projects": [
        {
            "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
            "is_domain": false,
            "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
            },
            "id": "a4adasfjljaaaakla12334jklga9sasfg",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

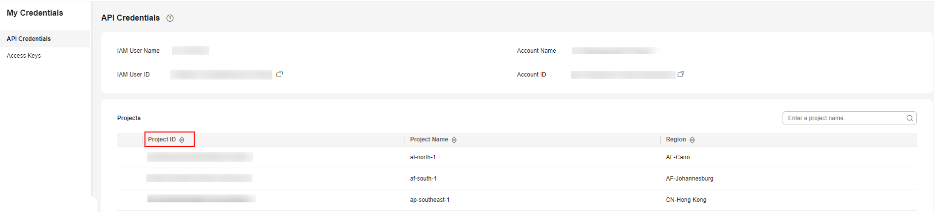
Obtain a Project ID from the Console

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

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

   On the **API Credentials** page, view the project ID in the project list.

Figure 8-1 Viewing the project ID



My Credentials

API Credentials

Access Keys

API Credentials

IAM User Name [redacted]

Account Name [redacted]

IAM User ID [redacted]

Account ID [redacted]

Projects

Enter a project name.

Project ID	Project Name	Region
[redacted]	af-north-1	AF-Cairo
[redacted]	af-south-1	AF-Johannesburg
[redacted]	ap-southeast-1	CN-Hong Kong

8.2 Application Message Format (Deprecated)

Table 8-1 Application message format

Type	Example Value	Remarks

HMS transparent push message	"hps": {
	"msg" : {
	"type" : 1,
	"body" : {"key":"value"}
	}
	}
		You can choose to specify the message in text or JSON format.
		NOTE
		• The JSON format is recommended, as shown in the example value.
		• The message in text format is encoded using UTF-8.
		• The message body cannot exceed 2 KB. For details, visit **Huawei Mobile Services**.

Type	Example Value	Remarks

Huawei notification push message	"hps": {
	"msg": {
	"type": 3,
	"body": {
	"content": "Push message content",
	"title": "Push message content"
	},
	"action": {
	"type": 1,
	"param": {
	"intent":"#Intent;compo=com.rvr/.Activity;S.W=U;end"
	}
	}
	},
	"ext": {
	"biTag": "Trump",
	"icon": "http://upload.w.org/00/150pxsvg.png"
	}
	}	

Apple platform message	{
	"aps": {
	"alert": "hello world"
	}
	}
		The message body cannot exceed 4 KB.
		For details, visit **Apple Push Notification Service**.

9 Out-of-Date APIs

9.1 Applications

9.1.1 Querying Platform Applications (Deprecated)

Function

This API is used to query platform applications.

URI

GET /v2/{project_id}/notifications/applications

Table 9-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Table 9-2 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. If the offset is an integer greater than 0 but less than the number of resources, all resources in excess of this offset will be queried. The default offset is 0. Default: **0**
limit	No	Integer	Specifies the maximum number of records in each query. The number ranges from 1 to 100 and is generally 10, 20, or 50. Specifies the number of resources returned on each page. The default number is 100. Default: **100**
name	No	String	Specifies the application name entered for a search. Backward fuzzy match is supported. The name contains a maximum of 64 characters, including only letters, underscores (_), and digits.
platform	No	String	Specifies the application platform. Only HMS, APNS, and APNS_SANDBOX are supported. HMS is a platform for developers to push Android messages. APNS and APNS_SANDBOX are platforms for pushing iOS messages.

Request Parameters

Table 9-3 Request header parameters

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

Response Parameters

Status code: 200

Table 9-4 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
application_count	Integer	Specifies the number of returned applications. No matter what offset and limit values you have set in the request, this parameter always returns the total number of applications in your account.
applications	Array of **ApplicationItem** objects	Specifies the application list.

Table 9-5 ApplicationItem

Parameter	Type	Description
name	String	Specifies the application name.
platform	String	Specifies the application platform.
create_time	String	Specifies when an application was created. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
application_urn	String	Specifies the unique resource ID of an application.
application_id	String	Specifies the application ID, which is unique.
enabled	String	Specifies whether the platform application is enabled.

Parameter	Type	Description
apple_certificate_expiration_date	String	Specifies the time when the Apple certificate expires. This parameter is available only for the APNS and APNS_SANDBOX platforms. The value is a UTC time in YYYY-MM-DDTHH:MM:SSZ format.

Status code: 400

Table 9-6 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-7 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-8 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-9 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/applications?
offset=0&limit=100&name=application_name&platform=APNS

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "application_count" : 2,
  "applications" : [ {
    "create_time" : "2018-02-08T08:30:16Z",
    "apple_certificate_expiration_date" : "2018-03-09T12:21:40Z",
    "name" : "application_name001",
    "platform" : "APNS",
    "enabled" : "true",
    "application_urn" : "urn:smn:regionId:429ffced18074da0938112f2c362b935:app-APNS-
application_name001",
    "application_id" : "b1b8643dc12b4g77ad6e35a16003119b"
  }, {
    "create_time" : "2018-01-11T12:58:58Z",
    "apple_certificate_expiration_date" : "2018-03-09T12:21:40Z",
    "name" : "application_name002",
    "platform" : "APNS",
    "enabled" : "true",
    "application_urn" : "urn:smn:regionId:429ffced18074da0938112f2c362b935:app-APNS-
application_name002",
    "application_id" : "a3a4643dc12b4g77ad6e35a16002558c"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found

Status Code	Description
500	Internal Server Error

Error Codes

See **Error Codes**.

9.1.2 Creating a Platform Application (Deprecated)

Function

This API is used to create a platform application.

URI

POST /v2/{project_id}/notifications/applications

Table 9-10 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Request Parameters

Table 9-11 Request header parameters

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

Table 9-12 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the application name. The name can contain a maximum of 64 characters, including only letters, underscores (_), and digits.
platform	Yes	String	Specifies the application platform. Only HMS, APNS, and APNS_SANDBOX are supported. HMS is a platform for developers to push Android messages. APNS and APNS_SANDBOX are platforms for pushing iOS messages.
platform_principal	Yes	String	For HMS, set it to the app ID, which contains a maximum of 20 characters, including only letters and digits. For APNS and APNS_SANDBOX, set it to the platform certificate, which is encoded using Base64 and cannot exceed 8 KB.
platform_credential	Yes	String	For HMS, set it to the app secret, which contains 32 to 64 characters and includes letters and digits only. For APNS and APNS_SANDBOX, set it to the certificate private key, which is encoded using Base64 and cannot exceed 8 KB.

Response Parameters

Status code: 201

Table 9-13 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
application_urn	String	Specifies the unique resource ID of an application.
application_id	String	Specifies the application resource ID.

Status code: 400

Table 9-14 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-15 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-16 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-17 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Creating a platform application

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/applications

{
  "name" : "application_name",
  "platform" : "HMS",
  "platform_principal" : "appId",
  "platform_credential" : "appSecret"
}
```

Example Responses

Status code: 201

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "application_urn" : "urn:smn:regionId:429ffced18074da0938112f2c362b935:app-APNS-application_name002",
  "application_id" : "a3a4643dc12b4g77ad6e35a16002558c"
}
```

Status Codes

Status Code	Description
201	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.1.3 Querying Application Attributes (Deprecated)

Function

This API is used to query attributes of a platform application.

URI

GET /v2/{project_id}/notifications/applications/{application_urn}

Table 9-18 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
application_urn	Yes	String	Specifies the unique resource identifier of the application. To obtain it, see Querying an Application.

Request Parameters

Table 9-19 Request header parameters

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

Response Parameters

Status code: 200

Table 9-20 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
application_id	String	Specifies the application ID, which is unique.

Parameter	Type	Description
attributes	**attributes** object	Specifies the application attributes.

Table 9-21 attributes

Parameter	Type	Description
enabled	String	Specifies whether the platform application is enabled.
apple_certificate_expiration_date	String	Specifies the time when the Apple certificate expires. This parameter is available only for the APNS and APNS_SANDBOX platforms. The value is a UTC time in YYYY-MM-DDTHH:MM:SSZ format.

Status code: 400

Table 9-22 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-23 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-24 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-25 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/applications/{application_urn}

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "application_id" : "b1b8643dc12b4g77ad6e35a16003119b",
  "attributes" : {
    "enabled" : "true",
    "apple_certificate_expiration_date" : "2018-03-09T12:21:40Z"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found

Status Code	Description
500	Internal Server Error

Error Codes

See **Error Codes**.

9.1.4 Updating a Platform Application (Deprecated)

Function

This API is used to update a platform application.

URI

PUT /v2/{project_id}/notifications/applications/{application_urn}

Table 9-26 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
application_urn	Yes	String	Specifies the unique resource identifier of the application. To obtain it, see Querying an Application.

Request Parameters

Table 9-27 Request header parameters

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

Table 9-28 Request body parameters

Parameter	Mandatory	Type	Description
platform_principal	Yes	String	For HMS, set it to the app ID, which contains a maximum of 20 characters, including only letters and digits. For APNS and APNS_SANDBOX, set it to the platform certificate, which is encoded using Base64 and cannot exceed 8 KB.
platform_credential	Yes	String	For HMS, set it to the app secret, which contains 32 to 64 characters and includes letters and digits only. For APNS and APNS_SANDBOX, set it to the certificate private key, which is encoded using Base64 and cannot exceed 8 KB.

Response Parameters

Status code: 200

Table 9-29 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 9-30 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-31 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-32 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-33 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Updating a platform application

```
PUT https://{SMN_Endpoint}/v2/{project_id}/notifications/applications/{application_urn}

{
  "platform_principal" : "appId",
  "platform_credential" : "appSecret"
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.1.5 Deleting a Platform Application (Deprecated)

Function

This API is used to delete a platform application.

URI

DELETE /v2/{project_id}/notifications/applications/{application_urn}

Table 9-34 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
application_urn	Yes	String	Specifies the unique resource identifier of the application. To obtain it, see Querying an Application.

Request Parameters

Table 9-35 Request header parameters

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

Response Parameters

Status code: 200

Table 9-36 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 9-37 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-38 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-39 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-40 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting a platform application

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/applications/{application_urn}

{
  "platform_principal" : "appId",
  "platform_credential" : "appSecret"
}
```

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.2 Application Endpoints

9.2.1 Querying the Endpoints of an Application (Deprecated)

Function

This API is used to query the endpoints of an application.

URI

GET /v2/{project_id}/notifications/applications/{application_urn}/endpoints

Table 9-41 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
application_urn	Yes	String	Specifies the unique resource identifier of the application. To obtain it, see Querying an Application.

Table 9-42 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. If the offset is an integer greater than 0 but less than the number of resources, all resources in excess of this offset will be queried. The default offset is 0. Default: **0**

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the maximum number of records in each query. The number ranges from 1 to 100 and is generally 10, 20, or 50. Specifies the number of resources returned on each page. The default number is 100. Default: **100**
enabled	No	String	Specifies whether a device is enabled. The value is a string and can be true or false.
token	No	String	Specifies the device token, which contains a maximum of 512 bytes.
user_data	No	String	Specifies the description or additional information about the endpoint. The value cannot exceed 2,048 bytes.

Request Parameters

Table 9-43 Request header parameters

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

Response Parameters

Status code: 200

Table 9-44 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
next_page_flag	Boolean	Specifies whether a link to the next page is provided.
endpoints	Array of **ApplicationEndpoint** objects	Specifies the Application_endpoint structure array.

Table 9-45 ApplicationEndpoint

Parameter	Type	Description
create_time	String	Specifies the time when the application was created. The UTC time is in YYYY-MM-DDTHH:MM:SSZ format.
endpoint_urn	String	Specifies the unique resource ID of an application endpoint.
user_data	String	Specifies the description or additional information about the endpoint. The value is a UTF-8 encoded string not exceeding 2,048 bytes.
enabled	String	Specifies whether the endpoint is enabled. The value is a string and can be true or false.
token	String	Specifies the device token, which contains a maximum of 512 bytes.

Status code: 400

Table 9-46 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-47 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-48 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-49 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/applications/{application_urn}/endpoints?
offset=0&limit=10

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "c90d871f77f7425cae0a8b772a563d17",
  "next_page_flag" : false,
  "endpoints" : [ {
    "create_time" : "2018-01-12T01:25:14Z",
    "endpoint_urn" : "urn:smn:regionId:429ffced18074da0938112f2c362b935:endpoint-APNS-
application_name-62f0bafec61c45fbbb5d0fc6bd696a2d",
    "user_data" : "no user data",
    "enabled" : "true",
    "token" : "1f2fda4c1013c47ac4c59bd22379d88e14b07b0ebf47f2f79d1a6b06ed03390d"
  } ]
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.2.2 Adding an Application Endpoint (Deprecated)

Function

This API is used to add an endpoint to a platform application.

URI

POST /v2/{project_id}/notifications/applications/{application_urn}/endpoints

Table 9-50 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
application_urn	Yes	String	Specifies the unique resource identifier of the application. To obtain it, see Querying an Application.

Request Parameters

Table 9-51 Request header parameters

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

Table 9-52 Request body parameters

Parameter	Mandatory	Type	Description
token	Yes	String	Specifies the mobile device token, which cannot exceed 512 bytes.
user_data	Yes	String	Specifies the description or additional information about the endpoint. The value is a UTF-8 encoded string not exceeding 2,048 bytes.

Response Parameters

Status code: 201

Table 9-53 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
endpoint_urn	String	Specifies the unique resource ID of an endpoint.

Status code: 400

Table 9-54 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.

Parameter	Type	Description
message	String	Describes the error message.

Status code: 403

Table 9-55 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-56 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-57 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Adding an application endpoint

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/applications/{application_urn}/endpoints

{
  "token" : "3708232124742383445",
  "user_data" : "This is token for user001"
}
```

Example Responses

Status code: 201

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "endpoint_urn" : "urn:smn:regionId:f96188c7ccaf4ffba0c9aa149ab2bd57:endpoint-APNS-
example_application_name-37a2cb947fa43a20bc66e91281194e7a"
}
```

Status Codes

Status Code	Description
201	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.2.3 Querying Application Endpoint Attributes (Deprecated)

Function

This API is used to query attributes of an endpoint.

URI

GET /v2/{project_id}/notifications/endpoints/{endpoint_urn}

Table 9-58 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.

Parameter	Mandatory	Type	Description
endpoint_urn	Yes	String	Specifies the unique resource identifier of the endpoint. To obtain it, see Querying the Endpoints of an Application.

Request Parameters

Table 9-59 Request header parameters

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

Response Parameters

Status code: 200

Table 9-60 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
attributes	**attributes** object	Specifies the attribute list.

Table 9-61 attributes

Parameter	Type	Description
enabled	String	Specifies whether an endpoint is enabled for notifications.
token	String	Specifies the device token.
user_data	String	Specifies the user data.

Status code: 400

Table 9-62 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-63 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-64 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-65 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

```
GET https://{SMN_Endpoint}/v2/{project_id}/notifications/endpoints/{endpoint_urn}

null
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "c90d871f77f7425cae0a8b772a563d17",
  "attributes" : {
    "enabled" : "true",
    "token" : "3708232124742383445",
    "user_data" : "abc"
  }
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.2.4 Updating Application Endpoint Attributes (Deprecated)

Function

This API is used to update application endpoint attributes.

URI

PUT /v2/{project_id}/notifications/endpoints/{endpoint_urn}

Table 9-66 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
endpoint_urn	Yes	String	Specifies the unique resource identifier of the endpoint. To obtain it, see Querying the Endpoints of an Application.

Request Parameters

Table 9-67 Request header parameters

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

Table 9-68 Request body parameters

Parameter	Mandatory	Type	Description
enabled	No	String	Specifies whether a device is enabled. The value is a string and can be true or false.
user_data	No	String	Specifies the description or additional information about the endpoint. The value is a UTF-8 encoded string not exceeding 2,048 bytes.

Response Parameters

Status code: 200

Table 9-69 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 9-70 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-71 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-72 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-73 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Updating application endpoint attributes

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/applications/{application_urn}/endpoints

{
  "enabled" : "false",
  "user_data" : "This is an APNS token for user001"
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.2.5 Deleting an Application Endpoint (Deprecated)

Function

This API is used to delete an application endpoint.

URI

DELETE /v2/{project_id}/notifications/endpoints/{endpoint_urn}

Table 9-74 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
endpoint_urn	Yes	String	Specifies the unique resource identifier of the endpoint. To obtain it, see Querying the Endpoints of an Application.

Request Parameters

Table 9-75 Request header parameters

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

Response Parameters

Status code: 200

Table 9-76 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 9-77 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.

Parameter	Type	Description
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-78 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-79 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-80 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

Deleting an application endpoint

```
DELETE https://{SMN_Endpoint}/v2/{project_id}/notifications/endpoints/{endpoint_urn}

null
```

Example Responses

None

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.3 Sending an Application Message

9.3.1 Publishing an Application Message (Deprecated)

Function

Sending an Application Message

URI

POST /v2/{project_id}/notifications/endpoints/{endpoint_urn}/publish

Table 9-81 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.For details about how to obtain the project ID, see **Obtaining the Project ID**.
endpoint_urn	Yes	String	Specifies the unique resource identifier of the endpoint. To obtain it, see Querying the Endpoints of an Application.

Request Parameters

Table 9-82 Request header parameters

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

Table 9-83 Request body parameters

Parameter	Mandatory	Type	Description
message	No	String	Select either message or message_structure. message: Publish an application message. message_structure: Publish an application message using a message structure. Specifies the message content. HMS, APNS, and APNS_SANDBOX are supported. HMS is a platform for developers to push Android messages. APNS and APNS_SANDBOX are platforms for pushing iOS messages. Message content for HMS cannot exceed 2 KB. Message content for APNS and APNS_SANDBOX cannot exceed 4 KB. For details about the message content format requirements of the push platform, see the format of the application message body. HMS transparent push message { "hps": { "msg": { "type": 1, "body": { "key": "value" } } } } Huawei notification push message { "hps": { "msg": { "type": 3, "body": { "content": "Push message content", "title": "Push message content" }, "action": { "type": 1, "param": { "intent": "#Intent;compo=com.rvr/.Activity;S.W=U;end" } } }, "ext": { "biTag": "Trump", "icon": "http://upload.w.org/00/150pxsvg.png" } } }

Parameter	Mandatory	Type	Description
			Apple platform message { "aps": { "alert": "hello world" } }
message_structure	No	String	Specifies the message content. HMS, APNS, and APNS_SANDBOX are supported. HMS is a platform for developers to push Android messages. APNS and APNS_SANDBOX are platforms for pushing iOS messages. Message content for HMS cannot exceed 2 KB. Message content for APNS and APNS_SANDBOX cannot exceed 4 KB. For details about the message content format requirements of the push platform, see the format of the application message body. HMS transparent push message { "HMS": { "hps": { "msg": { "type": 1, "body": { "key": "value" } } } } } Huawei notification push message { "HMS": { "hps": { "msg": { "type": 3, "body": { "content": "Push message content", "title": "Push message content" }, "action": { "type": 1, "param": { "intent": "#Intent;compo=com.rvr/.Activity;S.W=U;end" } } }, "ext": { "biTag": "Trump", "icon": "http://upload.w.org/00/150pxsvg.png" } } } } Apple platform message { "APNS": { "aps": { "alert": "hello world" } } }

Parameter	Mandatory	Type	Description
time_to_live	No	String	Specifies message TTL (counting from the time when you make the sending request).SMN will count the time a message already consumed in the SMN system before forwarding it to the mobile push platform. The message will be forwarded only when the time it consumed in SMN is less than the TTL, and the consumed time is deducted from the TTL. If the consumed time has exceeded the TTL, the message will be discarded.The time period is measured in seconds, and the default TTL is 3600s (one hour). The retention period must be a positive integer less than or equal to 86,400 (3600 x 24). Default: **3600**

Response Parameters

Status code: 200

Table 9-84 Response body parameters

Parameter	Type	Description
message_id	String	Specifies the message ID, which is unique.
request_id	String	Specifies the request ID, which is unique.

Status code: 400

Table 9-85 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 403

Table 9-86 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 404

Table 9-87 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Status code: 500

Table 9-88 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID, which is unique.
code	String	Specifies the error code.
message	String	Describes the error message.

Example Requests

- Publishing an application message

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/endpoints/{endpoint_urn}/publish

{
  "message" : {
    "hps" : {
      "msg" : {
        "action" : {
          "param" : {
            "intent" : "#Intent;compo=com.HC.SmnAppPush.defaultAction/.Activity;S.W=U;end"
          },
          "type" : 1
        },
        "body" : {
          "title" : "hello world",
          "content" : "hello world"
        },
        "type" : 3
      }
    }
  },
  "time_to_live" : 3600
}
```

- Publishing an application message using a message structure

```
POST https://{SMN_Endpoint}/v2/{project_id}/notifications/endpoints/{endpoint_urn}/publish

{
  "message" : {
    "HMS" : {
      "hps" : {
        "msg" : {
          "action" : {
            "param" : {
              "intent" : "#Intent;compo=com.HC.SmnAppPush.defaultAction/.Activity;S.W=U;end"
            },
            "type" : 1
          },
          "body" : {
            "title" : "hello world",
            "content" : "hello world"
          },
          "type" : 3
        }
      }
    }
  },
  "time_to_live" : 3600
}
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "6a63a18b8bab40ffb71ebd9cb80d0085",
  "message_id" : "bf94b63a5dfb475994d3ac34664e24f2"
}
```

Status Codes

Status Code	Description
200	OK
400	Bad Request
403	Unauthorized
404	Not Found
500	Internal Server Error

Error Codes

See **Error Codes**.

9.4 Confirming a Subscription (V2 Deprecated)

Function

This API is used to confirm the subscription.

APIs

PUT	/v2/{tenant_id}/notifications/topics/{topic_urn}/subscriptions/{subscription_urn}/confirmation

PUT	/v2/{tenant_id}/notifications/subscriptions/confirmation

API for managing the lifecycle of an email template (Deprecated)

POST	/v2/{tenant_id}/notifications/integrated_system_message_template

API for managing the lifecycle of a contact person (Deprecated)

POST	/v2/notifications/contact/callback/{event_type}

API for converting a **P12** certificate to a **Cert** certificate (Deprecated)

POST	/rest/v2/notifications/credentials_file_upload

A Appendix

A.1 Topic Attribute List

Attribute Name	Function	Default Value
access_policy	Configures topic access policy, which is used to limit the access permission of other users.	Empty string
introduction	Configures the topic introduction, which is displayed in the email or SMS confirmation messages.	Empty string

A.2 General Request Return Values

A.2.1 Exception Response

- Parameter description

Parameter	Type	Description
request_id	String	Request ID
code	String	See **Error Codes**.
message	String	See **Error Codes**.

- Example

```
{
    "request_id": "aad0860d089c482b943971f802a6718e",
    "code": "SMN.0006",
    "message": "Topic not found."
}
```

A.2.2 Status Codes

Table A-1 Normal values

Returned Value	Description
200	Task submitted.
201	Resource created.

Table A-2 Abnormal values

Returned Value	Description
400 Bad Request	Incorrect request parameters.
401 Unauthorized	Authentication failed.
403 Forbidden	No permission to access the requested resource.
404 Not Found	The requested resource does not exist.
500 Internal Server Error	Failed to complete the request because the service is abnormal.
501 Not Implemented	Failed to complete the request. The server does not support the requested function.
502 Bad Gateway	Failed to complete the request. The server receives an invalid response from the upstream server.
503 Service Unavailable	Failed to complete the request because the service is unavailable.
504 Gateway Timeout	Gateway times out.

A.3 Error Codes

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

Status Code	Error Codes	Error Message	Description	Solution
200	SMN.0190	This endpoint already exists.	The endpoint already exists.	Enter another endpoint.
400	SMN.0002	Parameter: Name is invalid.	Invalid topic name.	Enter a valid topic name.
400	SMN.0003	Parameter: DisplayName is invalid.	Invalid display name.	Enter a valid display name for the topic.
400	SMN.0005	Parameter: TopicUrn is invalid.	Invalid topic URN.	Enter a valid topic_urn value.
400	SMN.0011	Parameter: Protocol is invalid.	Invalid protocol.	Specify a correct protocol.
400	SMN.0012	Parameter: Endpoint is invalid.	Invalid endpoint.	Specify a valid endpoint.
400	SMN.0014	Parameter: SubscriptionUrn is invalid.	Invalid subscription URN.	Enter a valid subscription-urn value.
400	SMN.0015	Parameter: Offset or limit is invalid.	Invalid offset or limit value.	Enter valid offset and limit values.
400	SMN.0017	Parameter: Remark is invalid.	Invalid remark value.	Enter a valid remark value.
400	SMN.0019	Producer Exceptions.	Generator exception.	Contact technical support.
400	SMN.0021	MessageStructure is invalid.	Incorrect message structure.	Enter a valid message_structure value.
400	SMN.0024	Parameter: content is invalid.	Invalid content.	Enter a valid content value.
400	SMN.0025	Template already exists.	The template already exists.	You cannot create the same template twice.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.00290	SubscriberUrn is unavailable.	SubscriberUrn is unavailable.	Check whether the value of SubscriberUrn is correct.
400	SMN.0031	Parameter: template_id is invalid.	Invalid template ID.	Enter a valid template_id value.
400	SMN.0032	Parameter: message_template_name is invalid.	Invalid template name.	Enter a valid message_template_name value.
400	SMN.0033	Parameter: push_policy is invalid.	Invalid push_policy value.	Enter a valid push_policy value.
400	SMN.0034	Parameter: subscription_auth_policy is invalid.	Invalid subscription_auth_policy value.	Enter a valid subscription_auth_policy value.
400	SMN.0035	Parameter: validation_policy is invalid.	Invalid validation_policy value.	Enter a valid validation_policy value.
400	SMN.0036	Parameter: xdomain_type is invalid.	Invalid xdomain_type value.	Enter a valid xdomain_type value.
400	SMN.0038	Parameter: tag is invalid.	Invalid tag value.	Enter a valid tag value.
400	SMN.0042	Parameter: Message_template_name is invalid.	Invalid template name.	Enter a valid message_template_name value.
400	SMN.0043	Parameter: subscription is invalid.	Invalid subscription.	Enter a valid subscription.
400	SMN.0044	Exceeded template limit.	Templates exceeded the maximum number allowed.	Each user can create no more than 100 message templates.
400	SMN.0046	Parameter: Attribute name is invalid.	Invalid attribute name.	Enter a valid attribute_name value.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0048	Parameter: Access policy is invalid.	Invalid access policy.	Enter a valid access_policy value.
400	SMN.0049	Parameter: Access policy version is invalid.	Invalid access policy version.	Enter a valid version of the access policy.
400	SMN.0050	Parameter: Access policy ID is invalid.	Invalid access policy ID.	Enter a valid access_policy id value.
400	SMN.0051	Parameter: Access policy statement is invalid.	Invalid value for access_policy statement.	Enter a valid access_policy statement value.
400	SMN.0052	Parameter: Access policy statement ID is invalid.	Invalid value for access_policy statement id.	Enter a valid access_policy statement id value.
400	SMN.0053	Parameter: Access policy statement resource is invalid.	Invalid value for access_policy statement resource.	Enter a valid access_policy statement resource value.
400	SMN.0054	Parameter: Access policy statement effect is invalid.	Invalid value for access_policy statement effect.	Enter a valid access_policy statement effect value.
400	SMN.0055	Parameter: Access policy statement action is invalid.	Invalid value for access_policy statement action.	Enter a valid access_policy statement action value.
400	SMN.0056	Parameter: Access policy statement principle is invalid.	Invalid value for access_policy statement principle.	Enter a valid access_policy statement principle value.
400	SMN.0058	Parameter: Access policy statement condition is invalid.	Invalid value for access_policy statement condition.	Enter a valid access_policy statement condition value.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0059	Parameter: job_id is invalid.	Invalid job_id value.	Contact technical support.
400	SMN.0061	Parameter: Service name is invalid.	Invalid Service name value.	Contact technical support.
400	SMN.0062	Parameter: Show name is invalid.	Invalid Show name value.	Contact technical support.
400	SMN.0066	Parameter: System parameter name is invalid.	Invalid System parameter name value.	Contact technical support.
400	SMN.0067	Parameter: System parameter value is invalid.	Invalid value for System parameter value.	Contact technical support.
400	SMN.0072	Parameter: Service is invalid.	Invalid service value.	Enter a valid service value.
400	SMN.0073	Parameter: Invalid action.	Invalid action value.	Enter a valid action value.
400	SMN.0074	Failed to add permission.	Failed to add permissions.	Contact technical support.
400	SMN.0077	Parameter: Status is invalid.	Invalid status value.	Enter a valid status value.
400	SMN.0080	The signId parameter is invalid.	Invalid sign-id value.	Enter a valid sign_id value.
400	SMN.0081	The signName parameter is invalid.	Invalid sign-name value.	Enter a valid sign_name value. It is a string of 2 to 8 characters containing only letters and digits.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0082	The length of the remark parameter value has reached the maximum allowed.	Remark too long	Enter a valid remark value that contains no more than 1500 bytes.
400	SMN.0087	The SMS message signature has not been approved.	SMS message signature not approved.	Enter an approved SMS message signature.
400	SMN.0090	Invalid domain ID.	Invalid domain ID.	Enter a valid domain_id value.
400	SMN.0092	The kafka_id parameter is invalid.	Invalid Kafka ID.	Contact technical support.
400	SMN.0093	The address_list parameter is invalid.	Invalid address list.	Contact technical support.
400	SMN.0094	The type of queue parameter is invalid.	Invalid queue parameter type.	Contact technical support.
400	SMN.0095	The count of queue parameter is invalid.	Invalid queue parameter statistics.	Contact technical support.
400	SMN.0097	The names parameter is invalid.	Invalid names value.	Contact technical support.
400	SMN.0100	The eventType parameter is invalid.	Invalid event type.	Enter a valid event_type value.
400	SMN.0102	Parameter: message_id is invalid.	Invalid message ID.	Enter a valid message_id value.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0103	The start_time parameter value is invalid.	Invalid start time.	Enter a valid start_time value.
400	SMN.0104	The end_time parameter value is invalid.	Invalid end time.	Enter a valid end_time value.
400	SMN.0108	Parameter: Type is invalid.	Invalid type.	Enter a valid type.
400	SMN.0110	Parameter: Start date is invalid.	Invalid start date.	Enter a valid start_date value.
400	SMN.0112	Parameter: End date is invalid.	Invalid end date.	Enter a valid end_date value.
400	SMN.0113	The start time and end time must be within the past 3 months.	The start time and end time must be within the past 3 months.	The start time and end time must be within the past 3 months.
400	SMN.0114	The end time must be later than the start time.	The end time must be later than the start time.	The end time must be later than the start time.
400	SMN.0115	Parameter: Application name is invalid.	Invalid application name.	Enter a valid application name.
400	SMN.0118	Wrong Certificate.	Certificate error.	Specify a valid certificate.
400	SMN.0121	An application with the same name but different properties already exists.	An application with the same name but different attributes already exists.	Enter a different application name.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0123	Parameter: ApplicationUrn is invalid.	Invalid application URN.	Enter a valid application_urn value.
400	SMN.0124	Parameter: application attribute is invalid.	Invalid application_attributes value.	Specify a valid application_attributes value.
400	SMN.0126	ContentType is wrong.	Incorrect content type.	Contact technical support.
400	SMN.0129	Certificate type is wrong.	Incorrect certificate type.	Contact technical support.
400	SMN.0132	The value of the User_data parameter is invalid.	Invalid user_data value.	Enter a valid user_data value.
400	SMN.0133	The device token already exists.	Device token already exists.	You cannot create the same device token twice.
400	SMN.0134	The endpoint URN is invalid.	Invalid endpoint URN.	Enter a valid endpoint_urn value.
400	SMN.0135	The endpoint does not exist.	The endpoint does not exist.	Specify a valid endpoint.
400	SMN.0136	The value of the enabled parameter is invalid.	Invalid enabled value.	Enter a valid enabled value.
400	SMN.0137	The application URN is invalid.	Invalid application URN.	Enter a valid application_urn value.
400	SMN.0140	The topic URN configured for the message sending failure event is invalid.	The topic URN configured for the message sending failure event is invalid.	Enter a valid topic_urn value.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0141	The topic configured for the message sending failure event does not exist.	The topic URN configured for the message sending failure event does not exist.	Enter a valid topic_urn value.
400	SMN.0142	The topic URN configured for the message sending success event is invalid.	Invalid topic URN.	Enter a valid topic_urn value.
400	SMN.0143	The topic configured for the message sending success event does not exist.	The topic URN configured for the message sending success event does not exist.	Enter a valid topic_urn value.
400	SMN.0144	The topic URN configured for the response message event is invalid.	The topic URN configured for the response message event is invalid.	Enter a valid topic_urn value.
400	SMN.0145	The topic configured for the response message event does not exist.	The topic URN configured for the response message event does not exist.	Enter a valid topic_urn value.
400	SMN.0148	Parameter: user_id is invalid.	Invalid user ID.	Enter a valid user_id value.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0150	Complete real-name registration using your company's certificate.	Use your certificate to complete real-name registration.	Contact technical support.
400	SMN.0151	The message signature type cannot be promotional.	The signature cannot be a promotional message signature.	Specify a transactional SMS message signature.
400	SMN.0152	The number allocated to the SMS template has reached the maximum allowed.	SMS templates reached the maximum allowed number.	You can create no more than 100 SMS templates.
400	SMN.0155	Parameter: SmsTemplate Type is invalid.	Invalid SMS template type.	Enter a valid sms_template_type value.
400	SMN.0156	Parameter: SmsTemplate Name is invalid.	Invalid SMS template name.	Enter a valid sms_template_name value.
400	SMN.0157	Parameter: SmsTemplate Content is invalid.	Invalid SMS template content.	Enter a valid sms_template_content value.
400	SMN.0159	The number of endpoints has reached the maximum allowed.	Endpoints reached the quota.	Enter no more than 1,000 phone numbers.
400	SMN.0160	Invalid endpoint.	Invalid endpoint.	Specify a valid endpoint.
400	SMN.0161	Invalid SMS template ID.	Invalid SMS template ID.	Enter a valid sms_template_id value.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0188	Invalid SMS template type.	Invalid SMS template type.	The template must be a promotional template.
400	SMN.0189	Failed to obtain the certificate required for reviewing the SMS message template function.	Failed to obtain the certificate required for viewing the SMS template.	Contact technical support.
400	SMN.0191	Parameter: Invalid enterprise name.	Invalid enterprise name.	Contact technical support.
400	SMN.0192	Parameter: Invalid enterprise certificate ID.	Invalid enterprise certificate.	Contact technical support.
400	SMN.0204	Promotional SMS messages must contain the way to unsubscribe in the end, for example, by replying "TD", or "T".	Promotional SMS messages must explain how recipients can unsubscribe, for example, by replying "unsubscribe".	Promotional SMS messages must explain how recipients can unsubscribe, for example, by replying "unsubscribe".
400	SMN.0215	The SMS message signature ID does not match the signature name in the SMS message template content.	The SMS message signature ID does not match the signature name in the SMS template.	Specify a signature ID that matches the signature name in the template content.

Status Code	Error Codes	Error Message	Description	Solution
400	SMN.0216	The template content does not include any valid SMS message signature.	No valid SMS message signature in the template content.	Enter a valid signature in the template content.
400	SMN.0217	The template content does not include any SMS message signature.	No SMS message signature in the template content.	The SMS template must contain an approved and available signature, for example, [xx mall].
400	SMN.0224	The SMS message template is invalid because the associated signature has been deleted.	The SMS template has expired due to the deletion of the associated signature.	Specify a valid sms_template_id value.
400	SMN.0227	The message contains sensitive words.	Sensitive words contained in the message content.	There were sensitive words in the message content. Modify the message content.
400	SMN.0230	Parameter: Invalid title.	Invalid title.	Enter a valid title.
400	SMN.0261	Check whether your account balance is sufficient and top up your account if necessary.	Check the account balance and top up your account if necessary.	Check the account balance and top up your account if necessary.
403	SMN.0001	No permission to request resources.	No permissions for the requested resource.	Request the required permissions.

Status Code	Error Codes	Error Message	Description	Solution
403	SMN.0004	Exceeded topic limit.	Topics exceeded the maximum number allowed.	You can create a maximum of 3,000 topics.
403	SMN.0007	Exceeded subscription limit.	Subscriptions exceeded the maximum allowed number.	You can add a maximum of 10,000 subscriptions to a topic.
403	SMN.0008	Parameter: Subject is invalid.	Invalid subject value.	Enter a valid subject value.
403	SMN.0009	Parameter: Message is invalid.	Invalid message.	Enter a valid message value.
403	SMN.0022	Parameter: token is invalid.	Invalid token value.	Specify a valid token.
403	SMN.00308	Policy doesn't allow {} to be performed.	Policy doesn't allow {} to be performed.	Check the SMN permission.
403	SMN.0047	Parameter: Value exceeds the maximum length.	value exceeds the maximum length.	Enter a valid value.
403	SMN.0057	The number of SMS messages exceeds the limit.	SMS messages exceeded the maximum number allowed.	You can send no more than 60 SMS messages to a phone number within one hour and no more than 200 within 12 hours.
403	SMN.0065	The number of email messages exceeds the limit.	Email messages exceeded the maximum number allowed.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
403	SMN.0069	Not authorized to subscribe internal endpoints.	Subscription endpoint not authorized.	Enter a correct HTTP/HTTPS endpoint.
403	SMN.0070	No permission to request resources. The role is op_restricted. Contact customer service.	Insufficient permissions to operate the current resource.	Check the account balance and top up your account if necessary. If the fault persists, contact customer service.
403	SMN.0071	No permission to request resources.Role is op_suspended.	Insufficient permissions to request the current resource. Your role is suspended.	Check whether your account has been frozen.
403	SMN.0075	Parameter: tags are too many.	Tags exceeded the maximum allowed number.	Enter no more than 90 non-repeated tags in a message template.
403	SMN.0078	The maximum number of subscriptions has been reached.	Subscriptions reached the maximum allowed number.	Create a topic.
403	SMN.0083	Finish real-name authentication before continuing.	Complete real-name authentication.	Contact technical support.
403	SMN.0086	The number allocated to the SMS message signature has reached the maximum allowed.	SMS message signatures have reached quota.	You can create no more than 10 SMS message signatures.

Status Code	Error Codes	Error Message	Description	Solution
403	SMN.0099	The SMN service has not been authorized to send messages to this queue.	SMN is not authorized to send messages to this queue.	Grant permissions to SMN in the queue policy.
403	SMN.0116	PlatForm do not support yet.	Unsupported platform.	Enter a supported platform.
403	SMN.0117	Certificate type is not supported.	Unsupported certificate type.	Enter a supported certificate type.
403	SMN.0119	PlatformPrinciple must be at most 8192 bytes long in UTF-8 encoding.	UTF-8 encoded platform_principle must not exceed 8192 bytes.	Enter a valid, UTF-8 encoded value for platform_principle that does not exceed 8192 bytes.
403	SMN.0120	PlatformCredential must be at most 8192 bytes long in UTF-8 encoding.	UTF-8 encoded platform_credential must not exceed 8192 bytes.	Enter a valid, UTF-8 encoded value for platform_credential that does not exceed 8192 bytes.
403	SMN.0127	File is too large.	Excessive file size.	Contact technical support.
403	SMN.0128	Password is empty.	Password not specified.	Contact technical support.
403	SMN.0131	The value of the Token parameter is invalid.	Invalid token.	Specify a valid token.
403	SMN.0147	Only enterprise users who use the certificate images for authentication can create message signatures.	Only enterprise users who use certificates for identity authentication can create message signatures.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
403	SMN.0153	Only enterprise users can create SMS message Template.	Only enterprise users can create SMS templates.	Complete enterprise authentication.
403	SMN.0154	Only enterprise users who use the certificate images for authentication can create sms message templates.	Only enterprise users who use certificates for identity authentication can create SMS templates.	Contact technical support.
403	SMN.0165	Parameter: tenant_token is invalid.	Invalid tenant token.	Specify a valid tenant_token value.
403	SMN.0195	Complete enterprise real-name authentication first.	Complete real-name authentication.	Complete real-name authentication.
403	SMN.0228	Do not have permission to use this API. Contact customer service to apply for permission.	Insufficient permissions to use this API.	Contact technical support to apply for the permissions required to use this API.
404	SMN.0006	Topic not found.	Topic not found.	The topic does not exist. Create a topic.
404	SMN.0013	Subscription resource not found.	No subscription endpoint found.	Specify a valid subscription endpoint.
404	SMN.0027	Template not found.	Specified template not found.	Using an existing template.
404	SMN.0045	Attribute not found.	Attribute not found.	Specify a valid attribute.

Status Code	Error Codes	Error Message	Description	Solution
404	SMN.0060	Job not found.	Failed to query the job.	Contact technical support.
404	SMN.0063	Cloud service not found.	Cloud service not found.	Contact technical support.
404	SMN.0068	System parameter not found.	System parameter not found.	Contact technical support.
404	SMN.0076	Default message template not found.	The message template using the default protocol does not exist.	Create a message template using the default protocol.
404	SMN.0084	The SMS message signature could not be found.	SMS message signature not found.	Enter an approved SMS message signature.
404	SMN.0091	Default system message template not found.	The message template using the default protocol does not exist.	Create a system template using the default protocol.
404	SMN.0096	Kafka not found.	Failed to find Kafka.	Contact technical support.
404	SMN.0101	The graph does not exist in the functiongraph service.	The function does not exist in FunctionGrap h.	Enter a valid FunctionGraph endpoint.
404	SMN.0109	Start date is not found.	start_date not found.	start_date must be specified.
404	SMN.0111	End date is not found.	end_date not found.	end_date must be specified.
404	SMN.0122	Application not found.	Application not found.	Enter a different application name.
404	SMN.0125	Endpoint does not exist.	The endpoint does not exist.	Specify a valid application endpoint.

Status Code	Error Codes	Error Message	Description	Solution
404	SMN.0130	The application does not exist.	The cloud platform does not exist.	Enter a valid application_urn value.
404	SMN.0162	SMS template not found.	The SMS template does not exist.	Specify a valid SMS template.
404	SMN.0167	Phone number does not exist.	The mobile number does not exist.	Enter a valid phone number.
404	SMN.0168	Email Address does not exist.	The e-mail address does not exist.	Enter a valid email address.
429	SMN.00359	Topic publishing exceeds the flow control limit, please try again later.	The number of messages published by a topic exceeds the traffic control limit.	Try again later.
429	SMN.00362	Topic publishing exceeds the global flow control limit, please try again later.	The number of messages published by a topic exceeds the global traffic control limit.	Try again later.
500	SMN.0016	Database Exceptions.	Database exception.	Contact technical support.
500	SMN.0018	Service internal error.	Internal service error.	Contact technical support.
500	SMN.0089	An error occurred in the BSS system. Try again later.	BSS error. Try again later.	Contact technical support.
500	SMN.0158	Failed to upload the file.	Failed to upload the file.	Contact technical support.

A.4 SMS Message Delivery Status (CMPP)

Status	Description
DELIVRD	The message is successfully delivered.
EXPIRED	Message delivery times out.
DELETED	The message has been deleted.
UNDELIV	The message delivery has failed.
ACCEPTD	The message is accepted by the receiver.
UNKNOWN	The message status is unknown.
REJECTD	The message is rejected.

A.5 Country/Region Code

Country/Region	Country/Region Code
Afghanistan	93
Albania	355
Algeria	213
American Samoa	1684
Andorra	376
Angola	244
Anguilla	1264
Antigua and Barbuda	1268
Argentina	54
Armenia	374
Aruba	297
Australia	61
Austria	43
Azerbaijan	994
Bahamas	1242
Bahrain	973
Bangladesh	880

Country/Region	Country/Region Code
Barbados	1246
Belarus	375
Belgium	32
Belize	501
Benin	229
Bermuda	1441
Bhutan	975
Bolivia	591
Bosnia and Herzegovina	387
Botswana	267
Brazil	55
British Indian Ocean Territory	246
Brunei	673
Bulgaria	359
Burkina Faso	226
Burundi	257
Cambodia	855
Cameroon	237
Canada	1
Cape Verde	238
Cayman Islands	1345
Central African Republic	236
Chad	235
Chile	56
Colombia	57
Comoros	269
Republic Of The Congo	242
Democratic Republic of the Congo	243
Cook Islands	682

Country/Region	Country/Region Code
Costa Rica	506
Croatia	385
Curacao	599
Cyprus	357
Czechia	420
Denmark	45
Djibouti	253
Dominica	1767
Dominican Republic	1809
Ecuador	593
Egypt	20
El Salvador	503
Equatorial Guinea	240
Eritrea	291
Estonia	372
Eswatini	268
Ethiopia	251
Falkland Islands	500
Faroe Islands	298
Fiji	679
Finland	358
France	33
French Guiana	594
French Polynesia	689
Gabon	241
Gambia	220
Georgia	995
Germany	49
Ghana	233
Gibraltar	350

Country/Region	Country/Region Code
Greece	30
Greenland	299
Grenada	1473
Guadeloupe	590
Guatemala	502
Guernsey	44
Guinea	224
Guinea-Bissau	245
Guyana	592
Haiti	509
Honduras	504
Hong Kong (China)	852
Hungary	36
Iceland	354
India	91
Indonesia	62
Iraq	964
Ireland	353
Isle of Man	44
Israel	972
Italy	39
Ivory Coast	225
Jamaica	1876
Japan	81
Jersey	44
Jordan	962
Kazakhstan	7
Kenya	254
Kuwait	965
Kyrgyzstan	996

Country/Region	Country/Region Code
Laos	856
Latvia	371
Lebanon	961
Lesotho	266
Liberia	231
Libya	218
Liechtenstein	423
Lithuania	370
Luxembourg	352
Macao (China)	853
Madagascar	261
Malawi	265
Malaysia	60
Maldives	960
Mali	223
Malta	356
Marshall Islands	692
Martinique	596
Mauritania	222
Mauritius	230
Mexico	52
Moldova	373
Monaco	377
Mongolia	976
Montenegro	382
Montserrat	1664
Morocco	212
Mozambique	258
Myanmar	95
Namibia	264

Country/Region	Country/Region Code
Nauru	674
Nepal	977
Netherlands	31
New Caledonia	687
New Zealand	64
Nicaragua	505
Niger	227
Nigeria	234
Niue	683
Norfolk Island	672
North Macedonia	389
Norway	47
Oman	968
Pakistan	92
Palau	680
Palestinian Territory	970
Panama	507
Papua New Guinea	675
Paraguay	595
Peru	51
Philippines	63
Poland	48
Portugal	351
Qatar	974
Réunion Island	262
Romania	40
Rwanda	250
Saint Kitts and Nevis	1869
Saint Lucia	1758
Saint Pierre and Miquelon	508

Country/Region	Country/Region Code
Saint Vincent and The Grenadines	1784
Samoa	685
San Marino	378
Sao Tome and Principe	239
Saudi Arabia	966
Senegal	221
Serbia	381
Seychelles	248
Sierra Leone	232
Singapore	65
Sint Maarten (Dutch Part)	1721
Slovakia	421
Slovenia	386
Solomon Islands	677
Somalia	252
South Africa	27
South Korea	82
Spain	34
Sri Lanka	94
Suriname	597
Sweden	46
Switzerland	41
Taiwan (China)	886
Tajikistan	992
Tanzania	255
Thailand	66
Timor L'este	670
Togo	228
Tonga	676

Country/Region	Country/Region Code
Trinidad and Tobago	1868
Tunisia	216
Türkiye	90
Turkmenistan	993
Turks and Caicos Islands	1649
Tuvalu	688
Uganda	256
Ukraine	380
United Arab Emirates	971
United Kingdom	44
United States	1
Uruguay	598
Uzbekistan	998
Vanuatu	678
Venezuela	58
Vietnam	84
Virgin Islands, British	1284
Wallis and Futuna	681
Yemen	967
Zambia	260
Zimbabwe	263