Short Message Service

SDK Documentation

Product Documentation

Copyright Notice

Trademark Notice

Tencent Cloud

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

SDK Documentation

SDK Download

Last updated : 2020-10-16 11:46:42

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

Relevant documentation and SDK source code can be obtained as follows:

Platform	SDK Documentation	GitHub Address
Java	SDK for Java documentation	SDK for Java
PHP	SDK for PHP documentation	SDK for PHP
Python	SDK for Python documentation	SDK for Python
JavaScript	SDK for Node.js documentation	SDK for Node.js
C#	SDK for C# documentation	SDK for C#
Go	SDK for Go documentation	SDK for Go

⚠ **Note :**

- All the SMS SDKs in this document are on the latest 3.0 version, and new SMS features will be updated here. You are strongly recommended to use SDK 3.0 and the supportive API 3.0.
- Existing APIs and SDKs on version 2.0 are still available but may be disused in the future.

SDK for Java

Last updated : 2020-10-16 11:42:04

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

> ⚠ **Note :**
>
> - SMS sending APIs
>   One message can be sent to up to 200 numbers at a time.
> - Signature and body template APIs
>   Individual users have no permission to use signature and body template APIs and can manage SMS signatures and SMS body templates only in the SMS Console. To use the APIs, change "Individual Identity" to "Organizational Identity".

Prerequisites

- You have activated SMS. For detailed directions, please see Getting Started with Mainland China SMS.
- If you need to send SMS messages in Mainland China, you need to purchase a Mainland China SMS package first.
- You have prepared the dependent environment: JDK 7 or above.
- You have obtained the `SecretID` and `SecretKey` on the **API Key Management** page in the CAM Console.
  - `SecretID` is used to identify the API caller.
  - `SecretKey` is used to encrypt the string to sign that can be verified on the server. **You should keep it private and avoid disclosure.**
- You have obtained the call address (endpoint). The call address of the SMS service is `sms.tencentcloudapi.com` .

Relevant Documents

- For more information on the APIs and their parameters, please see API Documentation.
- You can download the SDK source code here.

Installing SDK

Installing through Maven (recommended)

Maven is a dependency management tool for Java that supports the dependencies your project requires and installs them into your project.

1. Go to Maven's official website to download the corresponding Maven installation package for your system and install it.
2. Add Maven dependencies by adding the following dependencies in Maven's pom.xml:

> ⚠ **Note :**
>
> The version number here is for demonstration only. Please get the latest version number in Maven Repository for replacement.

```
<dependency>
<groupId>com.tencentcloudapi</groupId>
<artifactId>tencentcloud-sdk-java</artifactId>
<version>3.1.62</version><!-- Note: the version number here is for demonstration only. Please
get the latest version number for replacement -->
</dependency>
```

3. For importing methods, please see the sample code.

Installing through source package

1. Download the source code package.
2. Decompress the source package to an appropriate location in your project.
3. Put the jar package under the `vendor` directory in a path that can be found by Java.
4. For importing methods, please see the sample code.

Sample Code

> ⓘ **Note :**
>
> All samples are for reference only and cannot be directly compiled and executed. You need to modify them based on your actual needs. You can also use API 3.0 Explorer to automatically generate the demo code as needed.

Each API has a corresponding request structure and a response structure. This document only lists the sample code of several common features. For more samples, please see SDK for Java Samples.

Applying for SMS template

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

// Import the optional configuration classes
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// Import the client of the SMS module
import com.tencentcloudapi.sms.v20190711.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20190711.models.AddSmsTemplateRequest;
import com.tencentcloudapi.sms.v20190711.models.AddSmsTemplateResponse;

/**
* Tencent Cloud Sms Sendsms
* https://intl.cloud.tencent.com/document/product/382/34859
*
*/
public class AddSmsTemplate
{
public static void main( String[] args )
{
try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
Credential cred = new Credential(”secretId”, ”secretKey”);

// Instantiate an HTTP option (optional; skip if there are no special requirements)
HttpProfile httpProfile = new HttpProfile();
// Set the proxy
httpProfile.setProxyHost(”host”);
httpProfile.setProxyPort(port);
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
```

```
httpProfile.setReqMethod("POST");
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.setConnTimeout(60);
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
httpProfile.setEndpoint("sms.tencentcloudapi.com");

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.setSignMethod("HmacSHA256");
clientProfile.setHttpProfile(httpProfile);
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
SmsClient client = new SmsClient(cred, "",clientProfile);
/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
AddSmsTemplateRequest req = new AddSmsTemplateRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS Helper: https://cloud.tencent.com/document/product/382/3773 */

/* Template name */
String templatename = "Tencent Cloud";
req.templateName(templatename);

/* Template content */
String templatecontent = "Your login verification code is {1}. Please enter it within {2} minutes. If the login was not initiated by you, please ignore this message.";
req.templateContent (templatecontent);
```

```
/* SMS type. 0: general SMS; 1: marketing SMS */
Long smstype = 0;
req.smsType(smstype);

/* Whether it is Global SMS. 0: Mainland China SMS; 1: Global SMS */
Long international = 0;
req.international(session);

/* Template remarks, such as reason for application and use case */
String remark = "xxx";
req.remark(remark);

/* Initialize the request by calling the `AddSmsTemplate` method on the client object. Note: the
request method name corresponds to the request object
* The returned `res` is an instance of the `AddSmsTemplateResponse` class which corresponds to th
e request object */
AddSmsTemplateResponse res = client.AddSmsTemplate(req);

// A string return packet in JSON format is output
System.out.println(AddSmsTemplateResponse.toJsonString(res));

// You can take a single value. You can view the definition of the return field in the API docume
ntation at the official website or by redirecting to the definition of the response object
System.out.println(res.getRequestId());

} catch (TencentCloudSDKException e) {
e.printStackTrace();
}
}
}
```

Sending SMS message

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

// Import the optional configuration classes
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// Import the client of the SMS module
import com.tencentcloudapi.sms.v20190711.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20190711.models.SendSmsRequest;
import com.tencentcloudapi.sms.v20190711.models.SendSmsResponse;
```

```
/**
* Tencent Cloud Sms Sendsms
* https://intl.cloud.tencent.com/document/product/382/34859
*
*/
public class SendSms
{
public static void main( String[] args )
{
try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
Credential cred = new Credential(”secretId”, ”secretKey”);

// Instantiate an HTTP option (optional; skip if there are no special requirements)
HttpProfile httpProfile = new HttpProfile();
// Set the proxy
httpProfile.setProxyHost(”host”);
httpProfile.setProxyPort(port);
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.setReqMethod(”POST”);
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.setConnTimeout(60);
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
httpProfile.setEndpoint(”sms.tencentcloudapi.com”);

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.setSignMethod(”HmacSHA256”);
clientProfile.setHttpProfile(httpProfile);
```

```
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
SmsClient client = new SmsClient(cred, "",clientProfile);
/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
SendSmsRequest req = new SendSmsRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
String appid = "1400009099";
req.setSmsSdkAppid(appid);

/* The content of SMS signature should be encoded in UTF-8. You must enter an approved signature, which can be viewed in the [SMS Console] */
String sign = "Signing information";
req.setSign(sign);

/* `senderid` for global SMS, which is not activated by default. If you need to activate it, please contact [SMS Helper] for assistance. This parameter should be left empty for Mainland China SMS */
String senderid = "xxx";
req.setSenderId(senderid);

/* User session content, which can carry context information such as user-side ID and will be returned as-is by the server */
String session = "xxx";
req.setSessionContext(session);

/* SMS code number extension, which is not activated by default. If you need to activate it, please contact [SMS Helper] */
String extendcode = "xxx";
req.setExtendCode(extendcode);

/* Template ID. You must enter the ID of an approved template, which can be viewed in the [SMS Co
nsole] */
String templateID = "400000";
req.setTemplateID(templateID);

/* Target mobile number in the e.164 standard (+[country/region code][mobile number])
* Example: +8613711112222, which has a + sign followed by 86 (country/region code) and then by 13
711112222 (mobile number). Up to 200 mobile numbers are supported */
String[] phoneNumbers = {"+8621212313123", "+8612345678902", "+8612345678903"};
req.setPhoneNumberSet(phoneNumbers);

/* Template parameters. If there are no template parameters, leave it empty */
String[] templateParams = {"5678"};
req.setTemplateParamSet(templateParams);

/* Initialize the request by calling the `SendSms` method on the client object. Note: the request
method name corresponds to the request object
* The returned `res` is an instance of the `SendSmsResponse` class which corresponds to the reque
st object */
SendSmsResponse res = client.SendSms(req);

// A string return packet in JSON format is output
System.out.println(SendSmsResponse.toJsonString(res));

// You can take a single value. You can view the definition of the return field in the API docume
ntation at the official website or by redirecting to the definition of the response object
System.out.println(res.getRequestId());

} catch (TencentCloudSDKException e) {
e.printStackTrace();
}
}
}
```

Pulling receipt status

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

// Import the optional configuration classes
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// Import the client of the SMS module
import com.tencentcloudapi.sms.v20190711.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20190711.models.PullSmsSendStatusRequest;
import com.tencentcloudapi.sms.v20190711.models.PullSmsSendStatusResponse;

/**
* Tencent Cloud Sms PullSmsSendStatus
* https://intl.cloud.tencent.com/document/product/382/34841
*
*/
public class PullSmsSendStatus {
public static void main(String[] args) {
try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi */
Credential cred = new Credential(”secretId”, ”secretKey”);

// Instantiate an HTTP option (optional; skip if there are no special requirements)
HttpProfile httpProfile = new HttpProfile();
// Set the proxy
httpProfile.setProxyHost(”host”);
httpProfile.setProxyPort(port);
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.setReqMethod(”POST”);
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.setConnTimeout(60);
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
httpProfile.setEndpoint(”sms.tencentcloudapi.com”);

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.setSignMethod(”HmacSHA256”);
clientProfile.setHttpProfile(httpProfile);
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
SmsClient client = new SmsClient(cred, "",clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
PullSmsSendStatusRequest req = new PullSmsSendStatusRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
String appid = "1400009099";
req.setSmsSdkAppid(appid);

// Set the maximum number of pulled entries. Maximum value: 100
Long limit = 5L;
req.setLimit(limit);

/* Initialize the request by calling the `PullSmsSendStatus` method on the client object. Note: the request method name corresponds to the request object
* The returned `res` is an instance of the `PullSmsSendStatusResponse` class which corresponds to the request object */
PullSmsSendStatusResponse res = client.PullSmsSendStatus(req);

// A string return packet in JSON format is output
System.out.println(PullSmsSendStatusResponse.toJsonString(res));

} catch (TencentCloudSDKException e) {
e.printStackTrace();
}
}
}
```

Collecting SMS message sending data

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

// Import the optional configuration classes
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// Import the client of the SMS module
import com.tencentcloudapi.sms.v20190711.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20190711.models.SendStatusStatisticsRequest;
import com.tencentcloudapi.sms.v20190711.models.SendStatusStatisticsResponse;

/**
* Tencent Cloud Sms SendStatusStatistics
* https://intl.cloud.tencent.com/document/product/382/34841
*
*/
public class SendStatusStatistics {
public static void main(String[] args) {
try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi */
Credential cred = new Credential(”secretId”, ”secretKey”);

// Instantiate an HTTP option (optional; skip if there are no special requirements)
HttpProfile httpProfile = new HttpProfile();
// Set the proxy
httpProfile.setProxyHost(”host”);
httpProfile.setProxyPort(port);
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.setReqMethod(”POST”);
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.setConnTimeout(60);
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
```

```
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
httpProfile.setEndpoint("sms.tencentcloudapi.com");

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.setSignMethod("HmacSHA256");
clientProfile.setHttpProfile(httpProfile);

/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
SmsClient client = new SmsClient(cred, "",clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
SendStatusStatisticsRequest req = new SendStatusStatisticsRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
String appid = "1400009099";
req.setSmsSdkAppid(appid);

// Set the maximum number of pulled entries. Maximum value: 100
Long limit = 5L;
req.setLimit(limit);
/* Offset, which is currently fixed at 0 */
Long offset = 0L;
req.setOffset(offset);
/* Start time of pull in the format of `yyyymmddhh` accurate to the hour */
String startdatetime = "2019071100";
```

```
req.setStartDateTime(startdatetime);
/* End time of pull in the format of `yyyymmddhh` accurate to the hour
* Note: `EndDataTime` must be later than `StartDateTime` */
String enddatatime = "2019071123";
req.setEndDataTime(enddatatime);

/* Initialize the request by calling the `SendStatusStatistics` method on the client object. Not
e: the request method name corresponds to the request object
* The returned `res` is an instance of the `SendStatusStatisticsResponse` class which corresponds
to the request object */
SendStatusStatisticsResponse res = client.SendStatusStatisticsStatus(req);

// A string return packet in JSON format is output
System.out.println(SendStatusStatisticsStatusResponse.toJsonString(res));

} catch (TencentCloudSDKException e) {
e.printStackTrace();
}
}
}
```

SDK for PHP

Last updated：2020-10-16 11:42:47

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

- SMS sending APIs
  One message can be sent to up to 200 numbers at a time.
- Signature and body template APIs
  Individual users have no permission to use signature and body template APIs and can manage SMS signatures and SMS body templates only in the SMS Console. To use the APIs, change "Individual Identity" to "Organizational Identity".

Prerequisites

- You have activated SMS. For detailed directions, please see Getting Started with Mainland China SMS.
- If you need to send SMS messages in Mainland China, you need to purchase a Mainland China SMS package first.
- You have prepared the dependent environment: PHP 5.6.33 or above.
- You have obtained the `SecretID` and `SecretKey` on the **API Key Management** page in the CAM Console.
  - `SecretID` is used to identify the API caller.
  - `SecretKey` is used to encrypt the string to sign that can be verified on the server. **You should keep it private and avoid disclosure.**
- You have obtained the call address (endpoint). The call address of the SMS service is `sms.tencentcloudapi.com`.

Relevant Documents

- For more information on the APIs and their parameters, please see API Documentation.
- You can download the SDK source code here.

Installing SDK

Composer is a dependency management tool for PHP that supports the dependencies your project requires and installs them into your project.

1. Install Composer.
   - For Windows, go to Composer official website to download the installation package and install it.
   - For Unix, run the following command to install Composer:

     ```
     curl -sS https://getcomposer.org/installer | php
     ```
2. Add dependencies.

   ```
   composer require tencentcloud/tencentcloud-sdk-php
   ```
3. Add the following reference code in the code.

   > This example is for reference only. Composer will generate a `vendor` directory in the project root directory, whose actual absolute path is `/path/to/`. If the operation is executed in the project root directory, you can omit the absolute path.

   ```
   require '/path/to/vendor/autoload.php';
   ```

Sample Code

> All samples are for reference only and cannot be directly compiled and executed. You need to modify them based on your actual needs. You can also use API 3.0 Explorer to automatically generate the demo code as needed.

Each API has a corresponding request structure and a response structure. This document only lists the sample code of several common features. For more samples, please see SDK for PHP Samples.

Applying for SMS template

```
<?php
require_once '../../../TCloudAutoLoader.php';
// Import the client of the SMS module
```

```
use TencentCloud¥Sms¥V20190711¥SmsClient;
// Import the `Request` class corresponding to the API to be requested
use TencentCloud¥Sms¥V20190711¥Models¥AddSmsTemplateRequest;
use TencentCloud¥Common¥Exception¥TencentCloudSDKException;
use TencentCloud¥Common¥Credential;
// Import the optional configuration classes
use TencentCloud¥Common¥Profile¥ClientProfile;
use TencentCloud¥Common¥Profile¥HttpProfile;

try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi */

$cred = new Credential(”xxx”, ”xxx”);
//$cred = new Credential(getenv(”TENCENTCLOUD_SECRET_ID”), getenv(”TENCENTCLOUD_SECRET_KEY”));

// Instantiate an HTTP option (optional; skip if there are no special requirements)
$httpProfile = new HttpProfile();
$httpProfile->setReqMethod(”GET”); // POST request (POST request by default)
$httpProfile->setReqTimeout(30); // Request timeout period in seconds (60 seconds by default)
$httpProfile->setEndpoint(”sms.tencentcloudapi.com”); // Specify the access region domain name (nearby access by default)

// Instantiate a client option (optional; skip if no special requirements are present)
$clientProfile = new ClientProfile();
$clientProfile->setSignMethod(”TC3-HMAC-SHA256”); // Specify the signature algorithm (`HmacSHA256` by default)
$clientProfile->setHttpProfile($httpProfile);

// Instantiate an SMS client object. `clientProfile` is optional
$client = new SmsClient($cred, ”ap-shanghai”, $clientProfile);

// Instantiate an `AddSmsTemplateRequest` request object. Each API corresponds to a request object
$req = new AddSmsTemplateRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
```

```
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* Template name */
$req->TemplateName = "Tencent Cloud";
/* Template content */
$req->TemplateContent = "Your login verification code is {1}. Please enter it within {2} minutes.
If the login was not initiated by you, please ignore this message.";
/* SMS type. 0: general SMS; 1: marketing SMS */
$req->SmsType = 0;
/* Whether it is Global SMS:
0: Mainland China SMS
1: Global SMS */
$req->International = 0;
/* Template remarks, such as reason for application and use case */
$req->Remark = "xxx";

// Initialize the request by calling the `AddSmsTemplate` method on the client object. Note: the
request method name corresponds to the request object
$resp = $client->AddSmsTemplate($req);

// A string return packet in JSON format is output
print_r($resp->toJsonString());

// You can take a single value. You can view the definition of the return field in the API docume
ntation at the official website or by redirecting to the definition of the response object
print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
echo $e;
}
```

Sending SMS message

```
<?php
require_once '../../../TCloudAutoLoader.php';
// Import the SMS client
use TencentCloud¥Sms¥V20190711¥SmsClient;
// Import the `Request` class corresponding to the API to be requested
use TencentCloud¥Sms¥V20190711¥Models¥SendSmsRequest;
use TencentCloud¥Common¥Exception¥TencentCloudSDKException;
use TencentCloud¥Common¥Credential;
// Import the optional configuration classes
use TencentCloud¥Common¥Profile¥ClientProfile;
use TencentCloud¥Common¥Profile¥HttpProfile;

try {
```

```
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi */

$cred = new Credential("xxx", "xxx");
//$cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"), getenv("TENCENTCLOUD_SECRET_KEY"));

// Instantiate an HTTP option (optional; skip if there are no special requirements)
$httpProfile = new HttpProfile();
$httpProfile->setReqMethod("GET"); // POST request (POST request by default)
$httpProfile->setReqTimeout(30); // Request timeout period in seconds (60 seconds by default)
$httpProfile->setEndpoint("sms.tencentcloudapi.com"); // Specify the access region domain name (nearby access by default)

// Instantiate a client option (optional; skip if no special requirements are present)
$clientProfile = new ClientProfile();
$clientProfile->setSignMethod("TC3-HMAC-SHA256"); // Specify the signature algorithm (`HmacSHA256` by default)
$clientProfile->setHttpProfile($httpProfile);

// Instantiate an SMS client object. `clientProfile` is optional
$client = new SmsClient($cred, "ap-shanghai", $clientProfile);

// Instantiate an SMS message sending request object. Each API corresponds to a request object
$req = new SendSmsRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
$req->SmsSdkAppid = "1400787878";
/* The content of SMS signature should be encoded in UTF-8. You must enter an approved signature, which can be viewed in the [SMS Console] */
$req->Sign = "xxx";
/* SMS code number extension, which is not activated by default. If you need to activate it, please contact [SMS Helper] */
```

```
$req->ExtendCode = "0";
/* Target mobile number in the e.164 standard (+[country/region code][mobile number])
* Example: +8613711112222, which has a + sign followed by 86 (country/region code) and then by 13711112222 (mobile number). Up to 200 mobile numbers are supported */
$req->PhoneNumberSet = array("+8613711112222", "+8613711333222", "+8613711144422");
/* `senderid` for global SMS, which is not activated by default. If you need to activate it, please contact [SMS Helper] for assistance. This parameter should be left empty for Mainland China SMS */
$req->SenderId = "xxx";
/* User session content, which can carry context information such as user-side ID and will be returned as-is by the server */
$req->SessionContext = "xxx";
/* Template ID. You must enter the ID of an approved template, which can be viewed in the [SMS Console] */
$req->TemplateID = "449739";
/* Template parameters. If there are no template parameters, leave it empty */
$req->TemplateParamSet = array("0");


// Initialize the request by calling the `SendSms` method on the client object. Note: the request method name corresponds to the request object
$resp = $client->SendSms($req);

// A string return packet in JSON format is output
print_r($resp->toJsonString());

// You can take a single value. You can view the definition of the return field in the API documentation at the official website or by redirecting to the definition of the response object
print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
echo $e;
}
```

Pulling receipt status

```
<?php
require_once '../../../TCloudAutoLoader.php';
// Import the client of the SMS module
use TencentCloud¥Sms¥V20190711¥SmsClient;
// Import the `Request` class corresponding to the API to be requested
use TencentCloud¥Sms¥V20190711¥Models¥PullSmsSendStatusRequest;
use TencentCloud¥Common¥Exception¥TencentCloudSDKException;
use TencentCloud¥Common¥Credential;
// Import the optional configuration classes
use TencentCloud¥Common¥Profile¥ClientProfile;
use TencentCloud¥Common¥Profile¥HttpProfile;
```

```
try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi */

$cred = new Credential(”xxx”, ”xxx”);
//$cred = new Credential(getenv(”TENCENTCLOUD_SECRET_ID”), getenv(”TENCENTCLOUD_SECRET_KEY”));

// Instantiate an HTTP option (optional; skip if there are no special requirements)
$httpProfile = new HttpProfile();
$httpProfile->setReqMethod(”GET”); // POST request (POST request by default)
$httpProfile->setReqTimeout(30); // Request timeout period in seconds (60 seconds by default)
$httpProfile->setEndpoint(”sms.tencentcloudapi.com”); // Specify the access region domain name (nearby access by default)

// Instantiate a client option (optional; skip if no special requirements are present)
$clientProfile = new ClientProfile();
$clientProfile->setSignMethod(”TC3-HMAC-SHA256”); // Specify the signature algorithm (`HmacSHA256` by default)
$clientProfile->setHttpProfile($httpProfile);

// Instantiate an SMS client object. `clientProfile` is optional
$client = new SmsClient($cred, ”ap-shanghai”, $clientProfile);

// Instantiate an SMS message sending request object. Each API corresponds to a request object
$req = new PullSmsSendStatusRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
$req->SmsSdkAppid = ”1400787878”;
/* Maximum number of pulled entries. Maximum value: 100 */
$req->Limit = 10;
// Initialize the request by calling the `PullSmsSendStatus` method on the client object. Note: the request method name corresponds to the request object
$resp = $client->PullSmsSendStatus($req);

// A string return packet in JSON format is output
print_r($resp->toJsonString());

// You can take a single value. You can view the definition of the return field in the API documentation at the official website or by redirecting to the definition of the response object
print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
echo $e;
}
```

Collecting SMS message sending data

```
<?php
require_once '../../../TCloudAutoLoader.php';
// Import the client of the SMS module
use TencentCloud¥Sms¥V20190711¥SmsClient;
// Import the `Request` class corresponding to the API to be requested
use TencentCloud¥Sms¥V20190711¥Models¥SendStatusStatisticsRequest;
use TencentCloud¥Common¥Exception¥TencentCloudSDKException;
use TencentCloud¥Common¥Credential;
// Import the optional configuration classes
use TencentCloud¥Common¥Profile¥ClientProfile;
use TencentCloud¥Common¥Profile¥HttpProfile;

try {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi */

$cred = new Credential("xxx", "xxx");
//$cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"), getenv("TENCENTCLOUD_SECRET_KEY"));

// Instantiate an HTTP option (optional; skip if there are no special requirements)
$httpProfile = new HttpProfile();
$httpProfile->setReqMethod("GET"); // POST request (POST request by default)
$httpProfile->setReqTimeout(30); // Request timeout period in seconds (60 seconds by default)
$httpProfile->setEndpoint("sms.tencentcloudapi.com"); // Specify the access region domain name (nearby access by default)

// Instantiate a client option (optional; skip if no special requirements are present)
$clientProfile = new ClientProfile();
$clientProfile->setSignMethod("TC3-HMAC-SHA256"); // Specify the signature algorithm (`HmacSHA256` by default)
$clientProfile->setHttpProfile($httpProfile);

// Instantiate an SMS client object. `clientProfile` is optional
$client = new SmsClient($cred, "ap-shanghai", $clientProfile);

// Instantiate an SMS message sending request object. Each API corresponds to a request object
$req = new SendStatusStatisticsRequest();

/* Populate the request parameters. Here, the member variables of the request object are the input parameters of the corresponding API
* You can view the definition of the request parameters in the API documentation at the official website or by redirecting to the definition of the request object
* Settings of a basic parameter:
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
$req->SmsSdkAppid = "1400787878";
/* Maximum number of pulled entries. Maximum value: 100 */
$req->Limit = 10;
/* Offset. Note: this parameter is currently fixed at 0 */
$req->Offset = 0;
/* Start time of pull in the format of `yyyymmddhh` accurate to the hour */
$req->StartDateTime = "2019122500";
/* End time of pull in the format of `yyyymmddhh` accurate to the hour
* Note: `EndDataTime` must be later than `StartDateTime` */
$req->EndDataTime = "2019122523";

// Initialize the request by calling the `SendStatusStatistics` method on the client object. Note: the request method name corresponds to the request object
$resp = $client->SendStatusStatistics($req);

// A string return packet in JSON format is output
print_r($resp->toJsonString());

// You can take a single value. You can view the definition of the return field in the API documentation at the official website or by redirecting to the definition of the response object
print_r($resp->TotalCount);
}
```

```
catch(TencentCloudSDKException $e) {
echo $e;
}
```

FAQs

Proxy

If there is a proxy in your environment, you need to set the system environment variable `https_proxy` ; otherwise, it may not be called normally, and a connection timeout exception will be thrown.

You can also use `GuzzleHttp` to configure the proxy:

```
$cred = new Credential(”secretId”, ”secretKey”);

$httpProfile = new HttpProfile();
$httpProfile->setProxy(’https://ip:port’);

$clientProfile = new ClientProfile();
$clientProfile->setHttpProfile($httpProfile);

$client = new OcrClient($cred, ’ap-beijing’, $this->clientProfile);
```

Certificate issue

If there is a problem with your PHP environment certificate, errors similar to `cURL error 60: See http://curl.haxx.se/libcurl/c/libcurl-errors.html` will occur, which can be solved as follows:

1. Download the certificate file cacert.pem and save it to the PHP installation path.
2. Edit the `php.ini` file: delete the semicolon comment (;) before the `curl.cainfo` configuration item and set the value to the absolute path of the saved certificate file `cacert.pem` .
3. Restart the services that depend on PHP.

php_curl extension

`GuzzleHttp` , which this SDK depends on, needs to have the `php_curl` extension enabled. Check whether the `php.ini` environment in your environment is enabled.

For example, on Linux with PHP 7.1, for services hosted under Apache, you can open `/etc/php/7.1/apache2/php.ini` to see whether the `extension=php_curl.dll` configuration item has been commented. Please delete the comment before it and restart Apache.

Exceptional web access

Access is normal on the command line, but the following error is reported when access is performed on a web server: cURL error 0: The cURL request was retried 3 times and did not succeed. The most likely reason for the failure is that cURL was unable to rewind the body of the request and subsequent retries resulted in the same error. Turn on the debug option to see what went wrong. See https://bugs.php.net/bug.php?id=47204 for more information. (see http://curl.haxx.se/libcurl/c/libcurl-errors.html).
This error may occur in different cases. You can run `php -r ”echo sys_get_temp_dir();”` to print the absolute path of the default system temporary directory and set `sys_temp_dir` in `php.ini` to the printed path, and then check whether this error is fixed.

SDK for Python

Last updated：2020-10-16 11:31:53

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

⚠ **Note：**

- SMS sending APIs
  One message can be sent to up to 200 numbers at a time.
- Signature and body template APIs
  Individual users have no permission to use signature and body template APIs and can manage SMS signatures and SMS body templates only in the SMS Console. To use the APIs, change "Individual Identity" to "Organizational Identity".

Prerequisites

- You have activated SMS. For detailed directions, please see Getting Started with Mainland China SMS.
- If you need to send SMS messages in Mainland China, you need to purchase a Mainland China SMS package first.
- You have prepared the dependent environment: Python v2.7–3.6.
- You have obtained the `SecretID` and `SecretKey` on the **API Key Management** page in the CAM Console.
  - `SecretID` is used to identify the API caller.
  - `SecretKey` is used to encrypt the string to sign that can be verified on the server. **You should keep it private and avoid disclosure.**
- You have obtained the call address (endpoint). The call address of the SMS service is `sms.tencentcloudapi.com` .

Relevant Documents

- For more information on the APIs and their parameters, please see API Documentation.
- You can download the SDK source code here.

Installing SDK

Installation through Pip (recommended)

1. Download and install pip.
2. Run the following command to install the SDK:

```
pip install tencentcloud-sdk-python
```

Installing through source package

1. Go to the GitHub code hosting page or quick download address to download the latest code.
2. After decompressing, run the following commands in sequence to install the SDK.

```
$ cd tencentcloud-sdk-python
$ python setup.py install
```

Sample Code

Note :

All samples are for reference only and cannot be directly compiled and executed. You need to modify them based on your actual needs. You can also use API 3.0 Explorer to automatically generate the demo code as needed.

Each API has a corresponding request structure and a response structure. This document only lists the sample code of several common features. For more samples, please see SDK for Python Samples.

Applying for SMS template

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
# Import the client models of the SMS module
from tencentcloud.sms.v20190711 import sms_client, models

# Import the optional configuration classes
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
```

```
# Required steps:
# Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
# This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
# You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
# Query the CAM key: https://console.cloud.tencent.com/cam/capi

cred = credential.Credential("secretId", "secretKey")
# cred = credential.Credential(
# os.environ.get(""),
# os.environ.get("")
# )

# Instantiate an HTTP option (optional; skip if there are no special requirements)
httpProfile = HttpProfile()
httpProfile.reqMethod = "POST" # POST request (POST request by default)
httpProfile.reqTimeout = 30 # Request timeout period in seconds (60 seconds by default)
httpProfile.endpoint = "sms.tencentcloudapi.com" # Specify the access region domain name (nearby access by default)

# Optional steps:
# Instantiate a client configuration object. You can specify the timeout period and other configuration items
clientProfile = ClientProfile()
clientProfile.signMethod = "TC3-HMAC-SHA256" # Specify the signature algorithm
clientProfile.language = "en-US"
clientProfile.httpProfile = httpProfile

# Instantiate an SMS client object
# The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant
client = sms_client.SmsClient(cred, "ap-guangzhou", clientProfile)

# Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
# You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
# An attribute may be of a basic type or import another data structure
# You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure
req = models.AddSmsTemplateRequest()

# Settings of a basic parameter:
# The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
# The SDK provides encapsulation functions for importing the pointers of basic parameters
# Help link:
# SMS Console: https://console.cloud.tencent.com/smsv2
# SMS Helper: https://intl.cloud.tencent.com/document/product/382/3773

# Template name
$req.TemplateName = "Tencent Cloud"
# Template content
$req.TemplateContent = "Your login verification code is {1}. Please enter it within {2} minutes.
If the login was not initiated by you, please ignore this message."
# SMS type. 0: general SMS; 1: marketing SMS
$req.SmsType = 0;
# Whether it is Global SMS:
# 0: Mainland China SMS
# 1: global SMS
$req.International = 0
# Template remarks, such as reason for application and use case
$req.Remark = "xxx"

# Initialize the request by calling the `AddSmsTemplate` method on the client object. Note: the r
equest method name corresponds to the request object
resp = client.AddSmsTemplate(req)

# A string return packet in JSON format is output
print(resp.to_json_string(indent=2))

except TencentCloudSDKException as err:
print(err)
```

Sending SMS message

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
# Import the client models of the SMS module
from tencentcloud.sms.v20190711 import sms_client, models

# Import the optional configuration classes
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
# Required steps:
# Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
Key` need to be passed in as the input parameters
# This example uses the way to read from the environment variable, so you need to set these two v
alues in the environment variable in advance
# You can also write the key pair directly into the code, but be careful not to copy, upload, or
share the code to others
```

```
# Query the CAM key: https://console.cloud.tencent.com/cam/capi

cred = credential.Credential("secretId", "secretKey")
# cred = credential.Credential(
# os.environ.get(""),
# os.environ.get("")
# )

# Instantiate an HTTP option (optional; skip if there are no special requirements)
httpProfile = HttpProfile()
httpProfile.reqMethod = "POST" # POST request (POST request by default)
httpProfile.reqTimeout = 30 # Request timeout period in seconds (60 seconds by default)
httpProfile.endpoint = "sms.tencentcloudapi.com" # Specify the access region domain name (nearby
access by default)

# Optional steps:
# Instantiate a client configuration object. You can specify the timeout period and other configu
ration items
clientProfile = ClientProfile()
clientProfile.signMethod = "TC3-HMAC-SHA256" # Specify the signature algorithm
clientProfile.language = "en-US"
clientProfile.httpProfile = httpProfile

# Instantiate an SMS client object
# The second parameter is the region information. You can directly enter the string `ap-guangzhou
` or import the preset constant
client = sms_client.SmsClient(cred, "ap-guangzhou", clientProfile)

# Instantiate a request object. You can further set the request parameters according to the API c
alled and actual conditions
# You can directly check the SDK source code to determine which attributes of `SendSmsRequest` ca
n be set
# An attribute may be of a basic type or import another data structure
# You are recommended to use the IDE for development where you can easily redirect to and view th
e documentation of each API and data structure
req = models.SendSmsRequest()

# Settings of a basic parameter:
# The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to
use pointers to assign values to them
# The SDK provides encapsulation functions for importing the pointers of basic parameters
# Help link:
# SMS Console: https://console.cloud.tencent.com/smsv2
# SMS Helper: https://intl.cloud.tencent.com/document/product/382/3773

# SMS application ID, which is the actual `SDKAppID` generated after an application is added in t
he [SMS Console], such as 1400006666
req.SmsSdkAppid = "1400787878"
```

```
# The content of SMS signature should be encoded in UTF-8. You must enter an approved signature, which can be viewed in the [SMS Console]
req.Sign = ”xxx”
# SMS code number extension, which is not activated by default. If you need to activate it, please contact [SMS Helper]
req.ExtendCode = ””
# User session content, which can carry context information such as user-side ID and will be returned as-is by the server
req.SessionContext = ”xxx”
# `senderid` for global SMS, which is not activated by default. If you need to activate it, please contact [SMS Helper] for assistance. This parameter should be left empty for Mainland China SMS
req.SenderId = ””
# Target mobile number in the e.164 standard (+[country/region code][mobile number])
# Example: +8613711112222, which has a + sign followed by 86 (country/region code) and then by 13711112222 (mobile number). Up to 200 mobile numbers are supported
req.PhoneNumberSet = [”+8613711112222”, ”+8613711333222”, ”+8613711144422”]
# Template ID. You must enter the ID of an approved template, which can be viewed in the [SMS Console]
req.TemplateID = ”449739”
# Template parameters. If there are no template parameters, leave it empty
req.TemplateParamSet = [”666”]


# Initialize the request by calling the `SendSms` method on the client object. Note: the request method name corresponds to the request object
resp = client.SendSms(req)

# A string return packet in JSON format is output
print(resp.to_json_string(indent=2))

except TencentCloudSDKException as err:
print(err)
```

Pulling receipt status

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
# Import the client models of the SMS module
from tencentcloud.sms.v20190711 import sms_client, models

# Import the optional configuration classes
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
# Required steps:
# Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
```

```
Key` need to be passed in as the input parameters
# This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
# You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
# Query the CAM key: https://console.cloud.tencent.com/cam/capi

cred = credential.Credential(”secretId”, ”secretKey”)
# cred = credential.Credential(
# os.environ.get(””),
# os.environ.get(””)
# )

# Instantiate an HTTP option (optional; skip if there are no special requirements)
httpProfile = HttpProfile()
httpProfile.reqMethod = ”POST” # POST request (POST request by default)
httpProfile.reqTimeout = 30 # Request timeout period in seconds (60 seconds by default)
httpProfile.endpoint = ”sms.tencentcloudapi.com” # Specify the access region domain name (nearby access by default)

# Optional steps:
# Instantiate a client configuration object. You can specify the timeout period and other configuration items
clientProfile = ClientProfile()
clientProfile.signMethod = ”TC3-HMAC-SHA256” # Specify the signature algorithm
clientProfile.language = ”en-US”
clientProfile.httpProfile = httpProfile

# Instantiate an SMS client object
# The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant
client = sms_client.SmsClient(cred, ”ap-guangzhou”, clientProfile)

# Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
# You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
# An attribute may be of a basic type or import another data structure
# You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure
req = models.PullSmsSendStatusRequest()

# Settings of a basic parameter:
# The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
# The SDK provides encapsulation functions for importing the pointers of basic parameters
# Help link:
# SMS Console: https://console.cloud.tencent.com/smsv2
# SMS Helper: https://intl.cloud.tencent.com/document/product/382/3773

# SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666
req.SmsSdkAppid = ”1400787878”
# Maximum number of pulled entries. Maximum value: 100
req.Limit = 10

# Initialize the request by calling the `PullSmsSendStatus` method on the client object. Note: the request method name corresponds to the request object
resp = client.PullSmsSendStatus(req)

# A string return packet in JSON format is output
print(resp.to_json_string(indent=2))


except TencentCloudSDKException as err:
print(err)
```

Collecting SMS message sending data

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
# Import the client models of the SMS module
from tencentcloud.sms.v20190711 import sms_client, models

# Import the optional configuration classes
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
# Required steps:
# Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
# This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
# You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
# Query the CAM key: https://console.cloud.tencent.com/cam/capi

cred = credential.Credential(”secretId”, ”secretKey”)
# cred = credential.Credential(
# os.environ.get(””),
# os.environ.get(””)
# )

# Instantiate an HTTP option (optional; skip if there are no special requirements)
```

```
httpProfile = HttpProfile()
httpProfile.reqMethod = "POST" # POST request (POST request by default)
httpProfile.reqTimeout = 30 # Request timeout period in seconds (60 seconds by default)
httpProfile.endpoint = "sms.tencentcloudapi.com" # Specify the access region domain name (nearby access by default)

# Optional steps:
# Instantiate a client configuration object. You can specify the timeout period and other configuration items
clientProfile = ClientProfile()
clientProfile.signMethod = "TC3-HMAC-SHA256" # Specify the signature algorithm
clientProfile.language = "en-US"
clientProfile.httpProfile = httpProfile

# Instantiate an SMS client object
# The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant
client = sms_client.SmsClient(cred, "ap-guangzhou", clientProfile)

# Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
# You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
# An attribute may be of a basic type or import another data structure
# You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure
req = models.SendStatusStatisticsRequest()

# Settings of a basic parameter:
# The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
# The SDK provides encapsulation functions for importing the pointers of basic parameters
# Help link:
# SMS Console: https://console.cloud.tencent.com/smsv2
# SMS Helper: https://intl.cloud.tencent.com/document/product/382/3773

# SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666
req.SmsSdkAppid = "1400787878"
# Maximum number of pulled entries. Maximum value: 100
req.Limit = 10
# Offset. Note: this parameter is currently fixed at 0
req.Offset = 0
# Start time of pull in the format of `yyyymmddhh` accurate to the hour
req.StartDateTime = 2019122400
# End time of pull in the format of `yyyymmddhh` accurate to the hour
# Note: `EndDataTime` must be later than `StartDateTime`
req.EndDataTime = 2019122523
```

```
# Initialize the request by calling the `SendStatusStatistics` method on the client object. Note:
the request method name corresponds to the request object
resp = client.SendStatusStatistics(req)

# A string return packet in JSON format is output
print(resp.to_json_string(indent=2))

except TencentCloudSDKException as err:
print(err)
```

SDK for Node.js

Last updated : 2020-10-16 11:43:07

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

- SMS sending APIs
  One message can be sent to up to 200 numbers at a time.
- Signature and body template APIs
  Individual users have no permission to use signature and body template APIs and can manage SMS signatures and SMS body templates only in the SMS Console. To use the APIs, change "Individual Identity" to "Organizational Identity".

Prerequisites

- You have activated SMS. For detailed directions, please see Getting Started with Mainland China SMS.
- If you need to send SMS messages in Mainland China, you need to purchase a Mainland China SMS package first.
- You have prepared the dependent environment: Node.js 7.10.1 or above.
- You have obtained the `SecretID` and `SecretKey` on the **API Key Management** page in the CAM Console.
  - `SecretID` is used to identify the API caller.
  - `SecretKey` is used to encrypt the string to sign that can be verified on the server. **You should keep it private and avoid disclosure.**
- You have obtained the call address (endpoint). The call address of the SMS service is `sms.tencentcloudapi.com` .

Relevant Documents

- For more information on the APIs and their parameters, please see API Documentation.
- You can download the SDK source code here.

Installing SDK

Installing through npm (recommended)

npm is a package management tool for Node.js.

1. Run the following installation command:

```
npm install tencentcloud-sdk-nodejs --save
```

2. Refer to the corresponding module code in your code. For more information, please see the sample code.

Installing through source package

1. Go to the GitHub code hosting page or quick download address to download the source code package.
2. Decompress the source package to an appropriate location in your project.
3. Refer to the corresponding module code in your code. For more information, please see the sample code.

Sample Code

> All samples are for reference only and cannot be directly compiled and executed. You need to modify them based on your actual needs. You can also use API 3.0 Explorer to automatically generate the demo code as needed.

Each API has a corresponding request structure and a response structure. This document only lists the sample code of several common features. For more samples, please see SDK for Node.js Samples.

Applying for SMS template

```
const tencentcloud = require(”../../../../tencentcloud-sdk-nodejs”);

// Import the client models of the SMS module
const smsClient = tencentcloud.sms.v20190711.Client;
const models = tencentcloud.sms.v20190711.Models;

const Credential = tencentcloud.common.Credential;
const ClientProfile = tencentcloud.common.ClientProfile;
const HttpProfile = tencentcloud.common.HttpProfile;


/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
//let cred = new Credential(process.env.TENCENTCLOUD_SECRET_ID, process.env.TENCENTCLOUD_SECRET_KEY);
let cred = new Credential(”xxx”, ”xxx”);
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
let httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.reqMethod = ”POST”;
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.reqTimeout = 30;
httpProfile.endpoint = ”sms.tencentcloudapi.com”;

// Instantiate a client option (optional; skip if no special requirements are present)
let clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field unless absolutely necessary */
clientProfile.signMethod = ”HmacSHA256”;
clientProfile.httpProfile = httpProfile;

/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` *
* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
let client = new smsClient(cred, ”ap-guangzhou”, clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
let req = new models.AddSmsTemplateRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* Template name */
req.TemplateName = ”Tencent Cloud”;
/* Template content */
req.TemplateContent = ”Your login verification code is {1}. Please enter it within {2} minutes. If the login was not initiated by you, please ignore this message.”;
/* SMS type. 0: general SMS; 1: marketing SMS */
req.SmsType = 0;
/* Whether it is Global SMS:
0: Mainland China SMS.
1: Global SMS */
req.International = 0;
/* Template remarks, such as reason for application and use case */
req.Remark = ”xxx”;

// Call the API you want to access through the client object; you need to pass in the request object and the response callback function
client.AddSmsTemplate(req, function (err, response) {
// The request returns an exception and the exception information is printed
if (err) {
console.log(err);
return;
}
// The request is returned normally, and the `response` object is printed
console.log(response.to_json_string());
});
```

Sending SMS message

```
const tencentcloud = require(”../../../../tencentcloud-sdk-nodejs”);

// Import the client models of the SMS module
const smsClient = tencentcloud.sms.v20190711.Client;
const models = tencentcloud.sms.v20190711.Models;
const Credential = tencentcloud.common.Credential;
const ClientProfile = tencentcloud.common.ClientProfile;
const HttpProfile = tencentcloud.common.HttpProfile;


/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
//let cred = new Credential(process.env.TENCENTCLOUD_SECRET_ID, process.env.TENCENTCLOUD_SECRET_KEY);
let cred = new Credential("xxx", "xxx");
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
let httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.reqMethod = "POST";
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.reqTimeout = 30;
httpProfile.endpoint = "sms.tencentcloudapi.com";

// Instantiate a client option (optional; skip if no special requirements are present)
let clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field unless absolutely necessary */
clientProfile.signMethod = "HmacSHA256";
clientProfile.httpProfile = httpProfile;

/* The SDK automatically specifies the domain name. Generally, you don't need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` *
* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
let client = new smsClient(cred, "ap-guangzhou", clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
```

```
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
let req = new models.SendSmsRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
req.SmsSdkAppid = ”1400787878”;
/* The content of SMS signature should be encoded in UTF-8. You must enter an approved signature, which can be viewed in the [SMS Console] */
req.Sign = ”xxx”;
/* SMS code number extension, which is not activated by default. If you need to activate it, please contact [SMS Helper] */
req.ExtendCode = ””;
/* `senderid` for global SMS, which is not activated by default. If you need to activate it, please contact [SMS Helper] for assistance. This parameter should be left empty for Mainland China SMS */
req.SenderId = ””;
/* User session content, which can carry context information such as user-side ID and will be returned as-is by the server */
req.SessionContext = ””;
/* Target mobile number in the e.164 standard (+[country/region code][mobile number])
* Example: +8613711112222, which has a + sign followed by 86 (country/region code) and then by 13711112222 (mobile number). Up to 200 mobile numbers are supported */
req.PhoneNumberSet = [”+8613711112222”, ”+8613711333222”, ”+8613711144422”];
/* Template ID. You must enter the ID of an approved template, which can be viewed in the [SMS Console] */
req.TemplateID = ”449739”;
/* Template parameters. If there are no template parameters, leave it empty */
req.TemplateParamSet = [”666”];

// Call the API you want to access through the client object; you need to pass in the request object and the response callback function
client.SendSms(req, function (err, response) {
// The request returns an exception and the exception information is printed
if (err) {
console.log(err);
return;
}
// The request is returned normally, and the `response` object is printed
console.log(response.to_json_string());
});
```

Pulling receipt status

```
const tencentcloud = require("../../../../tencentcloud-sdk-nodejs");

// Import the client models of the SMS module
const smsClient = tencentcloud.sms.v20190711.Client;
const models = tencentcloud.sms.v20190711.Models;

const Credential = tencentcloud.common.Credential;
const ClientProfile = tencentcloud.common.ClientProfile;
const HttpProfile = tencentcloud.common.HttpProfile;



/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
Key` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two v
alues in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or
share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
//let cred = new Credential(process.env.TENCENTCLOUD_SECRET_ID, process.env.TENCENTCLOUD_SECRET_K
EY);
let cred = new Credential("xxx", "xxx");
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configu
ration items */
let httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some l
arge requests */
httpProfile.reqMethod = "POST";
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.reqTimeout = 30;
httpProfile.endpoint = "sms.tencentcloudapi.com";

// Instantiate a client option (optional; skip if no special requirements are present)
let clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field unless absolutely
necessary */
clientProfile.signMethod = "HmacSHA256";
clientProfile.httpProfile = httpProfile;

/* The SDK automatically specifies the domain name. Generally, you don't need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` *
* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
let client = new smsClient(cred, "ap-guangzhou", clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
let req = new models.PullSmsSendStatusRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
req.SmsSdkAppid = "1400787878";
/* Maximum number of pulled entries. Maximum value: 100 */
req.Limit = 10;

// Call the API you want to access through the client object; you need to pass in the request object and the response callback function
client.PullSmsSendStatus(req, function (err, response) {
// The request returns an exception and the exception information is printed
if (err) {
console.log(err);
return;
}
// The request is returned normally, and the `response` object is printed
console.log(response.to_json_string());
});
```

Collecting SMS message sending data

```
const tencentcloud = require("../../../../tencentcloud-sdk-nodejs");

// Import the client models of the SMS module
const smsClient = tencentcloud.sms.v20190711.Client;
const models = tencentcloud.sms.v20190711.Models;

const Credential = tencentcloud.common.Credential;
const ClientProfile = tencentcloud.common.ClientProfile;
const HttpProfile = tencentcloud.common.HttpProfile;


/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
Key` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two v
alues in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or
share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
//let cred = new Credential(process.env.TENCENTCLOUD_SECRET_ID, process.env.TENCENTCLOUD_SECRET_K
EY);
let cred = new Credential("xxx", "xxx");
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configu
ration items */
let httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some l
arge requests */
httpProfile.reqMethod = "POST";
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.reqTimeout = 30;
httpProfile.endpoint = "sms.tencentcloudapi.com";

// Instantiate a client option (optional; skip if no special requirements are present)
let clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field unless absolutely
necessary */
clientProfile.signMethod = "HmacSHA256";
clientProfile.httpProfile = httpProfile;

/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain
name, but if you are accessing a service in a finance AZ, you must manually specify the domain na
me
```

```
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` *
* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
let client = new smsClient(cred, "ap-guangzhou", clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
let req = new models.SendStatusStatisticsRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

// SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666
req.SmsSdkAppid = "1400787878";
// Maximum number of pulled entries. Maximum value: 100
req.Limit = 10;
// Offset, which is currently fixed at 0
req.Offset = 0;
// Start time of pull in the format of `yyyymmddhh` accurate to the hour
req.StartDateTime = 2019122400;
// End time of pull in the format of `yyyymmddhh` accurate to the hour
// Note: `EndDataTime` must be later than `StartDateTime`
req.EndDataTime = 2019122523;

// Call the API you want to access through the client object; you need to pass in the request object and the response callback function
client.SendStatusStatistics(req, function (err, response) {
// The request returns an exception and the exception information is printed
if (err) {
console.log(err);
return;
}
// The request is returned normally, and the `response` object is printed
console.log(response.to_json_string());
});
```

SDK for C#

Last updated：2020-10-16 11:31:52

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

> ⚠ **Note：**
>
> - SMS sending APIs
>   One message can be sent to up to 200 numbers at a time.
> - Signature and body template APIs
>   Individual users have no permission to use signature and body template APIs and can manage SMS signatures and SMS body templates only in the SMS Console. To use the APIs, change "Individual Identity" to "Organizational Identity".

Prerequisites

- You have activated SMS. For detailed directions, please see Getting Started with Mainland China SMS.
- If you need to send SMS messages in Mainland China, you need to purchase a Mainland China SMS package first.
- You have prepared the dependent environments: .NET Framework 4.5+ and .NET Core 2.1.
- You have obtained the `SecretID` and `SecretKey` on the **API Key Management** page in the CAM Console.
  - `SecretID` is used to identify the API caller.
  - `SecretKey` is used to encrypt the string to sign that can be verified on the server. **You should keep it private and avoid disclosure.**
- You have obtained the call address (endpoint). The call address of the SMS service is `sms.tencentcloudapi.com`.

Relevant Documents

- For more information on the APIs and their parameters, please see API Documentation.
- You can download the SDK source code here.

Installing SDK

Installing through NuGet (recommended)

1. Run the following installation command:

   ```
   dotnet add package TencentCloudSDK --version 3.0.0
   ```

   Other information can be obtained through NuGet.
2. Add the package through Visual Studio.

Installing through source package

1. Go to the GitHub code hosting page or quick download address to download the latest code.
2. After decompressing, install it in the working directory.
3. Use Visual Studio 2017 to open the compilation.

Sample Code

> **Note :**
>
> All samples are for reference only and cannot be directly compiled and executed. You need to modify them based on your actual needs. You can also use API 3.0 Explorer to automatically generate the demo code as needed.

Each API has a corresponding request structure and a response structure. This document only lists the sample code of several common features. For more samples, please see SDK for C# Samples.

Applying for SMS template

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20190711;
using TencentCloud.Sms.V20190711.Models;

namespace TencentCloudExamples
{
class AddSmsTemplate
{
static void Main(string[] args)
{
```

```
try
{
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
Credential cred = new Credential {
SecretId = ”xxx”,
SecretKey = ”xxx”
};
/*
Credential cred = new Credential {
SecretId = Environment.GetEnvironmentVariable(”TENCENTCLOUD_SECRET_ID”),
SecretKey = Environment.GetEnvironmentVariable(”TENCENTCLOUD_SECRET_KEY”)
};*/

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
/* Optional steps
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
HttpProfile httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.ReqMethod = ”GET”;
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.Timeout = 10; // Request connection timeout period in seconds (60 seconds by default)
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
httpProfile.Endpoint = ”sms.tencentcloudapi.com”;
// Proxy server. Set it when there is a proxy server in your environment
httpProfile.WebProxy = Environment.GetEnvironmentVariable(”HTTPS_PROXY”);

clientProfile.HttpProfile = httpProfile;
```

```
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
SmsClient client = new SmsClient(cred, "ap-guangzhou", clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
AddSmsTemplateRequest req = new AddSmsTemplateRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/sms/smslist
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* Template name */
req.TemplateName = "Tencent Cloud";
/* Template content */
req.TemplateContent = "Your login verification code is {1}. Please enter it within {2} minutes. If the login was not initiated by you, please ignore this message.";
/* SMS type. 0: general SMS; 1: marketing SMS */
req.SmsType = 0;
/* Whether it is Global SMS:
0: Mainland China SMS
1: Global SMS */
req.International = 0;
/* Template remarks, such as reason for application and use case */
req.Remark = "xxx";

// Initialize the request by calling the `AddSmsTemplate` method on the client object. Note: the request method name corresponds to the request object
// The returned `resp` is an instance of the `AddSmsTemplateResponse` class which corresponds to the request object
AddSmsTemplateResponse resp = client.AddSmsTemplate(req);

// A string return packet in JSON format is output
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
Console.WriteLine(e.ToString());
```

```
}
Console.Read();
}
}
}
```

Sending SMS message

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20190711;
using TencentCloud.Sms.V20190711.Models;

namespace TencentCloudExamples
{
class SendSms
{
static void Main(string[] args)
{
try
{
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
Key` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two v
alues in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or
share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
Credential cred = new Credential {
SecretId = "xxx",
SecretKey = "xxx"
};
/*
Credential cred = new Credential {
SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID"),
SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY")
};*/

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configu
ration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
```

```
clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
/* Optional steps
* Instantiate a client configuration object. You can specify the timeout period and other configu
ration items */
HttpProfile httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some l
arge requests */
httpProfile.ReqMethod = ”GET”;
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.Timeout = 10; // Request connection timeout period in seconds (60 seconds by default)
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain
name, but if you are accessing a service in a finance AZ, you must manually specify the domain na
me
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcl
oudapi.com` */
httpProfile.Endpoint = ”sms.tencentcloudapi.com”;
// Proxy server. Set it when there is a proxy server in your environment
httpProfile.WebProxy = Environment.GetEnvironmentVariable(”HTTPS_PROXY”);

clientProfile.HttpProfile = httpProfile;
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou
` or import the preset constant */
SmsClient client = new SmsClient(cred, ”ap-guangzhou”, clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API
called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` ca
n be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view th
e documentation of each API and data structure */
SendSmsRequest req = new SendSmsRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to
use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/sms/smslist
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

req.SmsSdkAppid = ”1400787878”;
/* The content of SMS signature should be encoded in UTF-8. You must enter an approved signature,
which can be viewed in the [SMS Console] */
req.Sign = ”xxx”;
```

```
/* SMS code number extension, which is not activated by default. If you need to activate it, please contact [SMS Helper] */
req.ExtendCode = "x";
/* `senderid` for global SMS, which is not activated by default. If you need to activate it, please contact [SMS Helper] for assistance. This parameter should be left empty for Mainland China SMS */
req.SenderId = "";
/* User session content, which can carry context information such as user-side ID and will be returned as-is by the server */
req.SessionContext = "";
/* Target mobile number in the e.164 standard (+[country/region code][mobile number])
* Example: +8613711112222, which has a + sign followed by 86 (country/region code) and then by 13711112222 (mobile number). Up to 200 mobile numbers are supported */
req.PhoneNumberSet = new String[] {"+8613711112222", "+8613711333222", "+8613711144422"};
/* Template ID. You must enter the ID of an approved template, which can be viewed in the [SMS Console] */
req.TemplateID = "449739";
/* Template parameters. If there are no template parameters, leave it empty */
req.TemplateParamSet = new String[] {"666"};


// Initialize the request by calling the `SendSms` method on the client object. Note: the request method name corresponds to the request object
// The returned `resp` is an instance of the `SendSmsResponse` class which corresponds to the request object
SendSmsResponse resp = client.SendSms(req);

// A string return packet in JSON format is output
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
Console.WriteLine(e.ToString());
}
Console.Read();
}
}
}
```

Pulling receipt status

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20190711;
using TencentCloud.Sms.V20190711.Models;
```

```
namespace TencentCloudExamples
{
class PullSmsSendStatus
{
static void Main(string[] args)
{
try
{
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
Credential cred = new Credential {
SecretId = ”xxx”,
SecretKey = ”xxx”
};
/*
Credential cred = new Credential {
SecretId = Environment.GetEnvironmentVariable(”TENCENTCLOUD_SECRET_ID”),
SecretKey = Environment.GetEnvironmentVariable(”TENCENTCLOUD_SECRET_KEY”)
};*/

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
/* Optional steps
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
HttpProfile httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.ReqMethod = ”POST”;
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.Timeout = 10; // Request connection timeout period in seconds (60 seconds by default)
/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcl
oudapi.com` */
httpProfile.Endpoint = "sms.tencentcloudapi.com";
// Proxy server. Set it when there is a proxy server in your environment
httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_PROXY");

clientProfile.HttpProfile = httpProfile;
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou
` or import the preset constant */
SmsClient client = new SmsClient(cred, "ap-guangzhou", clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API
called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` ca
n be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view th
e documentation of each API and data structure */
PullSmsSendStatusRequest req = new PullSmsSendStatusRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to
use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/sms/smslist
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

// Set the maximum number of pulled entries. Maximum value: 100
req.Limit = 100;
/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in
the [SMS Console], such as 1400006666 */
req.SmsSdkAppid = "1400009099";

// Initialize the request by calling the `PullSmsSendStatus` method on the client object. Note: t
he request method name corresponds to the request object
// The returned `resp` is an instance of the `PullSmsSendStatusResponse` class which corresponds
to the request object
PullSmsSendStatusResponse resp = client.PullSmsSendStatus(req);

// A string return packet in JSON format is output
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
Console.WriteLine(e.ToString());
}
```

```
Console.Read();
}
}
}
```

Collecting SMS message sending data

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20190711;
using TencentCloud.Sms.V20190711.Models;

namespace TencentCloudExamples
{
class SendStatusStatistics
{
static void Main(string[] args)
{
try
{
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
Key` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two v
alues in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or
share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
Credential cred = new Credential {
SecretId = "xxx",
SecretKey = "xxx"
};
/*
Credential cred = new Credential {
SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID"),
SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY")
};*/

/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configu
ration items */
ClientProfile clientProfile = new ClientProfile();
/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
```

```
/* Optional steps
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
HttpProfile httpProfile = new HttpProfile();
/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
httpProfile.ReqMethod = "POST";
/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
httpProfile.Timeout = 10; // Request connection timeout period in seconds (60 seconds by default)
/* The SDK automatically specifies the domain name. Generally, you don't need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
httpProfile.Endpoint = "sms.tencentcloudapi.com";
// Proxy server. Set it when there is a proxy server in your environment
httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_PROXY");

clientProfile.HttpProfile = httpProfile;
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
SmsClient client = new SmsClient(cred, "ap-guangzhou", clientProfile);

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of `SendSmsRequest` can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
SendStatusStatisticsRequest req = new SendStatusStatisticsRequest();

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/sms/smslist
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
req.SmsSdkAppid = "1400009099";
// Set the maximum number of pulled entries. Maximum value: 100
req.Limit = 5L;
```

```
/* Offset, which is currently fixed at 0 */
req.Offset = 0L;
/* Start time of pull in the format of `yyyymmddhh` accurate to the hour */
req.StartDateTime = "2019071100";
/* End time of pull in the format of `yyyymmddhh` accurate to the hour
* Note: `EndDataTime` must be later than `StartDateTime` */
req.EndDataTime = "2019071123";

// Initialize the request by calling the `SendStatusStatistics` method on the client object. Note: the request method name corresponds to the request object
// The returned `resp` is an instance of the `SendStatusStatisticsResponse` class which corresponds to the request object
SendStatusStatisticsResponse resp = client.SendStatusStatistics(req);

// A string return packet in JSON format is output
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
Console.WriteLine(e.ToString());
}
Console.Read();
}
}
}
```

SDK for Go

Last updated : 2020-10-16 11:31:52

SDK 3.0 is a companion tool for the TencentCloud API 3.0 platform. You can use all SMS APIs through the SDK. The new SDK version is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.

> ⚠ **Note :**
>
> - SMS sending APIs
>   One message can be sent to up to 200 numbers at a time.
> - Signature and body template APIs
>   Individual users have no permission to use signature and body template APIs and can manage SMS signatures and SMS body templates only in the SMS Console. To use the APIs, change "Individual Identity" to "Organizational Identity".

Prerequisites

- You have activated SMS. For detailed directions, please see Getting Started with Mainland China SMS.
- If you need to send SMS messages in Mainland China, you need to purchase a Mainland China SMS package first.
- You have prepared the dependent environment: Node.js 7.10.1 or above.
- You have obtained the `SecretID` and `SecretKey` on the **API Key Management** page in the CAM Console.
  - `SecretID` is used to identify the API caller.
  - `SecretKey` is used to encrypt the string to sign that can be verified on the server. **You should keep it private and avoid disclosure.**
- You have obtained the call address (endpoint). The call address of the SMS service is `sms.tencentcloudapi.com` .

Relevant Documents

- For more information on the APIs and their parameters, please see API Documentation.
- You can download the SDK source code here.

Installing SDK

Installing through go get (recommended)

You are recommended to install the SDK by using the tool that comes with the language:

```
go get -u github.com/tencentcloud/tencentcloud-sdk-go
```

Installing through source code

1. Go to the GitHub code hosting page or quick download address to download the latest code.
2. Decompress and install in the `$GOPATH/src/github.com/tencentcloud` directory.

Sample Code

Note :

All samples are for reference only and cannot be directly compiled and executed. You need to modify them based on your actual needs. You can also use API 3.0 Explorer to automatically generate the demo code as needed.

Each API has a corresponding request structure and a response structure. This document only lists the sample code of several common features. For more samples, please see SDK for Go Samples.

Applying for SMS template

```
package main

import (
"encoding/json"
"fmt"

"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
sms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20190711"// Import SMS
)

func main() {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secret
Key` need to be passed in as the input parameters
```

```
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
credential := common.NewCredential(
// os.Getenv(”TENCENTCLOUD_SECRET_ID”),
// os.Getenv(”TENCENTCLOUD_SECRET_KEY”),
”xxx”,
”xxx”,
)
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
cpf := profile.NewClientProfile()

/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
cpf.HttpProfile.ReqMethod = ”POST”

/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
//cpf.HttpProfile.ReqTimeout = 5

/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
cpf.HttpProfile.Endpoint = ”sms.tencentcloudapi.com”

/* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field unless absolutely necessary */
cpf.SignMethod = ”HmacSHA1”

/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
client, _ := sms.NewClient(credential, ”ap-guangzhou”, cpf)

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
request := sms.NewAddSmsTemplateRequest()
```

```
/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to
use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* Template name */
request.TemplateName = common.StringPtr("Tencent Cloud")
/* Template content */
request.TemplateContent = common.StringPtr("Your login verification code is {1}. Please enter it
within {2} minutes. If the login was not initiated by you, please ignore this message.")
/* SMS type. 0: general SMS; 1: marketing SMS */
request.SmsType = common.Uint64Ptr(0)
/* Whether it is Global SMS:
0: Mainland China SMS
1: Global SMS */
request.International = common.Uint64Ptr(0)
/* Template remarks, such as reason for application and use case */
request.Remark = common.StringPtr("xxx")

// Call the API you want to access through the client object. You need to pass in the request obj
ect
response, err := client.AddSmsTemplate(request)
// Handle the exception
if _, ok := err.(*errors.TencentCloudSDKError); ok {
fmt.Printf("An API error has returned: %s", err)
return
}
// This is a direct failure instead of SDK exception. You can add other troubleshooting measures
in the real code
if err != nil {
panic(err)
}
b, _ := json.Marshal(response.Response)
// Print the returned JSON string
fmt.Printf("%s", b)
}
```

Sending SMS message

```
package main

import (
"encoding/json"
```

```
"fmt"

"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
sms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20190711"// Import SMS
)

func main() {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
credential := common.NewCredential(
// os.Getenv("TENCENTCLOUD_SECRET_ID"),
// os.Getenv("TENCENTCLOUD_SECRET_KEY"),
"xxx",
"xxx",
)
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
cpf := profile.NewClientProfile()

/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
cpf.HttpProfile.ReqMethod = "POST"

/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
//cpf.HttpProfile.ReqTimeout = 5

/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

/* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field unless absolutely necessary */
cpf.SignMethod = "HmacSHA1"
```

```
/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
client, _ := sms.NewClient(credential, "ap-guangzhou", cpf)

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
request := sms.NewSendSmsRequest()

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in the [SMS Console], such as 1400006666 */
request.SmsSdkAppid = common.StringPtr("1400787878")
/* The content of SMS signature should be encoded in UTF-8. You must enter an approved signature, which can be viewed in the [SMS Console] */
request.Sign = common.StringPtr("xxx")
/* `senderid` for global SMS, which is not activated by default. If you need to activate it, please contact [SMS Helper] for assistance. This parameter should be left empty for Mainland China SMS */
request.SenderId = common.StringPtr("xxx")
/* User session content, which can carry context information such as user-side ID and will be returned as-is by the server */
request.SessionContext = common.StringPtr("xxx")
/* SMS code number extension, which is not activated by default. If you need to activate it, please contact [SMS Helper] */
request.ExtendCode = common.StringPtr("0")
/* Template parameters. If there are no template parameters, leave it empty */
request.TemplateParamSet = common.StringPtrs([]string{"0"})
/* Template ID. You must enter the ID of an approved template, which can be viewed in the [SMS Console] */
request.TemplateID = common.StringPtr("449739")
/* Target mobile number in the e.164 standard (+[country/region code][mobile number])
* Example: +8613711112222, which has a + sign followed by 86 (country/region code) and then by 13711112222 (mobile number). Up to 200 mobile numbers are supported */
request.PhoneNumberSet = common.StringPtrs([]string{"+8613711112222", "+8613711333222", "+8613711144422"})
```

```
// Call the API you want to access through the client object. You need to pass in the request object
response, err := client.SendSms(request)
// Handle the exception
if _, ok := err.(*errors.TencentCloudSDKError); ok {
fmt.Printf(”An API error has returned: %s”, err)
return
}
// This is a direct failure instead of SDK exception. You can add other troubleshooting measures in the real code
if err != nil {
panic(err)
}
b, _ := json.Marshal(response.Response)
// Print the returned JSON string
fmt.Printf(”%s”, b)
}
```

Pulling receipt status

```
package main

import (
”encoding/json”
”fmt”

”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common”
”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors”
”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile”
sms ”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20190711”// Import SMS
)

func main() {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
credential := common.NewCredential(
// os.Getenv(”TENCENTCLOUD_SECRET_ID”),
// os.Getenv(”TENCENTCLOUD_SECRET_KEY”),
”xxx”,
”xxx”,
)
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configu
ration items */
cpf := profile.NewClientProfile()

/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some l
arge requests */
cpf.HttpProfile.ReqMethod = "POST"

/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
//cpf.HttpProfile.ReqTimeout = 5

/* The SDK automatically specifies the domain name. Generally, you don't need to specify a domain
name, but if you are accessing a service in a finance AZ, you must manually specify the domain na
me
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcl
oudapi.com` */
cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
cpf.SignMethod = "HmacSHA1"

/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou
` or import the preset constant */
client, _ := sms.NewClient(credential, "ap-guangzhou", cpf)

/* Instantiate a request object. You can further set the request parameters according to the API
called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view th
e documentation of each API and data structure */
request := sms.NewPullSmsSendStatusRequest()

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to
use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in
```

```
the [SMS Console], such as 1400006666 */
request.SmsSdkAppid = common.StringPtr(”1400787878”)
/* Maximum number of pulled entries. Maximum value: 100 */
request.Limit = common.Uint64Ptr(10)

// Call the API you want to access through the client object. You need to pass in the request object
response, err := client.PullSmsSendStatus(request)
// Handle the exception
if _, ok := err.(*errors.TencentCloudSDKError); ok {
fmt.Printf(”An API error has returned: %s”, err)
return
}
// This is a direct failure instead of SDK exception. You can add other troubleshooting measures in the real code
if err != nil {
panic(err)
}
b, _ := json.Marshal(response.Response)
// Print the returned JSON string
fmt.Printf(”%s”, b)
}
```

Collecting SMS message sending data

```
package main

import (
”encoding/json”
”fmt”
”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common”
”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors”
”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile”
sms ”github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20190711” // Import SMS
)

func main() {
/* Required steps:
* Instantiate an authentication object. The Tencent Cloud account key pair `secretId` and `secretKey` need to be passed in as the input parameters
* This example uses the way to read from the environment variable, so you need to set these two values in the environment variable in advance
* You can also write the key pair directly into the code, but be careful not to copy, upload, or share the code to others
* Query the CAM key: https://console.cloud.tencent.com/cam/capi*/
credential := common.NewCredential(
// os.Getenv(”TENCENTCLOUD_SECRET_ID”),
// os.Getenv(”TENCENTCLOUD_SECRET_KEY”),
”xxx”,
”xxx”,
)
/* Optional steps:
* Instantiate a client configuration object. You can specify the timeout period and other configuration items */
cpf := profile.NewClientProfile()

/* The SDK uses the POST method by default
* If you need to use the GET method, you can set it here, but the GET method cannot handle some large requests */
cpf.HttpProfile.ReqMethod = ”POST”

/* The SDK has a default timeout period. Do not adjust it unless absolutely necessary
* If needed, check in the code to get the latest default value */
//cpf.HttpProfile.ReqTimeout = 5

/* The SDK automatically specifies the domain name. Generally, you don’t need to specify a domain name, but if you are accessing a service in a finance AZ, you must manually specify the domain name
* For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-shanghai-fsi.tencentcloudapi.com` */
cpf.HttpProfile.Endpoint = ”sms.tencentcloudapi.com”

/* The SDK uses `TC3-HMAC-SHA256` to sign by default
* Do not modify this field unless absolutely necessary */
cpf.SignMethod = ”HmacSHA1”

/* Instantiate an SMS client object
* The second parameter is the region information. You can directly enter the string `ap-guangzhou` or import the preset constant */
client, _ := sms.NewClient(credential, ”ap-guangzhou”, cpf)

/* Instantiate a request object. You can further set the request parameters according to the API called and actual conditions
* You can directly check the SDK source code to determine which attributes of the API can be set
* An attribute may be of a basic type or import another data structure
* You are recommended to use the IDE for development where you can easily redirect to and view the documentation of each API and data structure */
request := sms.NewSendStatusStatisticsRequest()

/* Settings of a basic parameter:
* The SDK uses the pointer style to specify parameters, so even for basic parameters, you need to use pointers to assign values to them
* The SDK provides encapsulation functions for importing the pointers of basic parameters
* Help link:
* SMS Console: https://console.cloud.tencent.com/smsv2
* SMS helper: https://intl.cloud.tencent.com/document/product/382/3773 */

/* SMS application ID, which is the actual `SDKAppID` generated after an application is added in
the [SMS Console], such as 1400006666 */
request.SmsSdkAppid = common.StringPtr("1400787878")
/* Maximum number of pulled entries. Maximum value: 100 */
request.Limit = common.Uint64Ptr(0)
/* Offset, which is currently fixed at 0 */
request.Offset = common.Uint64Ptr(0)
/* Start time of pull in the format of `yyyymmddhh` accurate to the hour */
request.StartDateTime = common.Uint64Ptr(2019122400)
/* End time of pull in the format of `yyyymmddhh` accurate to the hour
* Note: `EndDataTime` must be later than `StartDateTime` */
request.EndDataTime = common.Uint64Ptr(2019122523)

// Call the API you want to access through the client object. You need to pass in the request obj
ect
response, err := client.SendStatusStatistics(request)
// Handle the exception
if _, ok := err.(*errors.TencentCloudSDKError); ok {
fmt.Printf("An API error has returned: %s", err)
return
}
// This is a direct failure instead of SDK exception. You can add other troubleshooting measures
in the real code
if err != nil {
panic(err)
}
b, _ := json.Marshal(response.Response)
// Print the returned JSON string
fmt.Printf("%s", b)
}
```