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：2024-01-18 16:27:43

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
C++	SDK for C++ documentation	C++ SDK

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：2024-06-27 15:48:00

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:

API version required for connecting to Tencent Cloud International:

SMS API v2021-01-11 is required. For details, see the sample code.

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 learned about the concept of region and selected a region as needed.

You have activated SMS. For detailed directions, please see Getting Started.

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.**

The endpoint 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 from Github Repository.

Installing SDK

Installing via Maven (recommended)

See Github Repository: Installing via Maven (recommended)

Installing via source package

See Github repository: Installing via source package

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 as shown below:

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 corresponding SMS module
import com.tencentcloudapi.sms.v20210111.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20210111.models.SendSmsRequest;
import com.tencentcloudapi.sms.v20210111.models.SendSmsResponse;

/**
 * Tencent Cloud Sms Sendsms
 *
 */
public class SendSms
{
    public static void main(String[] args)
    {
        try {
            // Instantiate an authentication object, input parameters require Tence
            // To protect key security, it is suggested to set keys in environment
            // Hardcoding keys into the code might lead to exposure through leaked
            // SecretId, SecretKey inquiry: https://console.tencentcloud.com/cam/ca
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"

            // Instantiate an HTTP option, optional, can be skipped if there are no
            HttpProfile httpProfile = new HttpProfile();
            // Starting from version 3.0.96, an HTTP proxy can be set separately (i
            // httpProfile.setProxyHost("Real proxy IP");
            // httpProfile.setProxyPort(Real proxy port);
            httpProfile.setReqMethod("GET"); // GET request (the default value is P
            httpProfile.setConnTimeout(10); // Request connection timeout time, in
            httpProfile.setWriteTimeout(10);  // Sets write timeout time, in second
            httpProfile.setReadTimeout(10);  // Sets read timeout time, in seconds

            /* Specifies the access region domain name. The default nearby region d
            httpProfile.setEndpoint("sms.tencentcloudapi.com");

            /* Optional steps:
             * Instantiate a client configuration object. You can specify the timeo
            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 the client object of the requested product (with SMS as
             * The second parameter is the information on the region you select in
            SmsClient client = new SmsClient(cred, "ap-singapore",clientProfile);
            /* Instantiate a request object. You can further set the request parame
             * You can directly check the SDK source code to determine which attrib
             * An attribute may be of a basic type or import another data structure
             * We recommend you use the IDE for development where you can easily re
            SendSmsRequest req = new SendSmsRequest();

            /* Populate the request parameters. Here, the member variables of the r
             * You can view the definition of the request parameters in the API doc
             * Settings of a basic parameter:
             * Help link:
             * SMS console: https://console.tencentcloud.com/smsv2
             * sms helper: https://www.tencentcloud.com/document/product/382/3773 *

            /* SMS application ID, which is the `SdkAppId` generated after an appli
            String sdkAppId = "2400006666";
            req.setSmsSdkAppId(sdkAppId);

            /* SMS signature content, which should be encoded in UTF-8. You must en
            String signName = "Signature content";
            req.setSignName(signName);

            /* `SenderId` for Global SMS, which is not activated by default. If you
            String senderid = "";
            req.setSenderId(senderid);

            /* User session content, which can carry context information such as us
            String sessionContext = "xxx";
            req.setSessionContext(sessionContext);

            /* SMS code number extension, which is not activated by default. If you
            String extendCode = "";
            req.setExtendCode(extendCode);

            /* Template ID. You must enter the ID of an approved template, which ca
            String templateId = "400000";
            req.setTemplateId(templateId);

            /* Target mobile number in the E.164 standard (+[country/region code][m
             * Example: +60198890000, which has a + sign followed by 60 (country/re
            String[] phoneNumberSet = {"+60198890000", "+60198890001", "+6019889000
            req.setPhoneNumberSet(phoneNumberSet);

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

            /* Initialize the request by calling the `SendSms` method on the client
             * The returned `res` is an instance of the `SendSmsResponse` class whi
            SendSmsResponse res = client.SendSms(req);

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

            // You can also take a single value. You can view the definition of the
            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 corresponding SMS module
import com.tencentcloudapi.sms.v20210111.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20210111.models.PullSmsReplyStatusRequest;
import com.tencentcloudapi.sms.v20210111.models.PullSmsReplyStatusResponse;

/**
 * Tencent Cloud Sms PullSmsSendStatus
 *
 */
public class PullSmsSendStatus {
    public static void main(String[] args) {
        try {
            // Instantiate an authentication object, input parameters require Tence
            // To protect key security, it is suggested to set keys in environment
            // Hardcoding keys into the code might lead to exposure through leaked
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"

            // Instantiate an HTTP option, optional, can be skipped if there are no
            HttpProfile httpProfile = new HttpProfile();
 // Starting from version 3.0.96, an HTTP proxy can be set separately (ignore it if
            // httpProfile.setProxyHost("Real proxy IP");
            // httpProfile.setProxyPort(Real proxy port);
            httpProfile.setReqMethod("GET"); // GET request (the default value is P
            httpProfile.setConnTimeout(30); // Request connection timeout time, in
            httpProfile.setWriteTimeout(30);  // Sets write timeout time, in second
            httpProfile.setReadTimeout(30);  // Sets read timeout time, in seconds

/* Specifies the access region domain name. The default nearby region domain name i
            httpProfile.setEndpoint("sms.tencentcloudapi.com");



            /* Optional steps:
             * Instantiate a client configuration object. You can specify the timeo
            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 the client object of the requested product (with SMS as
             * The second parameter is the information on the region you select in
            SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile);

            /* Instantiate a request object. You can further set the request parame
             * You can directly check the SDK source code to determine which attrib
             * An attribute may be of a basic type or import another data structure
             * We recommend you use the IDE for development where you can easily re
            PullSmsSendStatusRequest req = new PullSmsSendStatusRequest();

            /* Populate the request parameters. Here, the member variables of the r
             * You can view the definition of the request parameters in the API doc
             * Settings of a basic parameter:
             * Help link:
             * SMS console: https://console.tencentcloud.com/smsv2
             * sms helper: https://www.tencentcloud.com/document/product/382/3773 *

            /* SMS application ID, which is the `SdkAppId` generated after an appli
            String sdkAppId = "2400006666";
            req.setSmsSdkAppId(sdkAppId);

            // 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 returned `res` is an instance of the `PullSmsSendStatusResponse`
            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 corresponding SMS module
```

```
import com.tencentcloudapi.sms.v20210111.SmsClient;

// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20210111.models.SendStatusStatisticsRequest;
import com.tencentcloudapi.sms.v20210111.models.SendStatusStatisticsResponse;

/**
 * Tencent Cloud Sms SendStatusStatistics
 *
 */
public class SendStatusStatistics {
    public static void main(String[] args) {
        try {
            // Instantiate an authentication object, input parameters require Tence
            // To protect key security, it is suggested to set keys in environment
            // Hardcoding keys into the code might lead to exposure through leaked
            // SecretId, SecretKey inquiry: https://console.tencentcloud.com/cam/ca
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"

            // Instantiate an HTTP option, optional, can be skipped if there are no
            HttpProfile httpProfile = new HttpProfile();
            // Starting from version 3.0.96, an HTTP proxy can be set separately (i
            // httpProfile.setProxyHost("Real proxy IP");
            // httpProfile.setProxyPort(Real proxy port);
            httpProfile.setReqMethod("GET"); // GET request (the default value is P
            httpProfile.setConnTimeout(30); // Request connection timeout time, in
            httpProfile.setWriteTimeout(30);  // Sets write timeout time, in second
            httpProfile.setReadTimeout(30);  // Sets read timeout time, in seconds

            /* Specifies the access region domain name. The default nearby region d
            httpProfile.setEndpoint("sms.tencentcloudapi.com");

            /* Optional steps:
             * Instantiate a client configuration object. You can specify the timeo
            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 the client object of the requested product (with SMS as
             * The second parameter is the information on the region you select in
            SmsClient client = new SmsClient(cred, "ap-singapore",clientProfile);

            /* Instantiate a request object. You can further set the request parame
             * You can directly check the SDK source code to determine which attrib
             * An attribute may be of a basic type or import another data structure
             * We recommend you use the IDE for development where you can easily re
            SendStatusStatisticsRequest req = new SendStatusStatisticsRequest();

            /* Populate the request parameters. Here, the member variables of the r
             * You can view the definition of the request parameters in the API doc
             * Settings of a basic parameter:
             * Help link:
             * SMS console: https://console.tencentcloud.com/smsv2
             * sms helper: https://www.tencentcloud.com/document/product/382/3773 *

            /* SMS application ID, which is the `SdkAppId` generated after an appli
            String sdkAppId = "2400006666";
            req.setSmsSdkAppId(sdkAppId);

            // Set the maximum number of pulled entries. Maximum value: 100
            Long limit = 5L;
            req.setLimit(limit);
            /* Offset. Note: this parameter is currently fixed at 0 */
            Long offset = 0L;
            req.setOffset(offset);
            /* Start time of pull in the format of `yyyymmddhh` accurate to the hou
            String beginTime = "2019071100";
            req.setBeginTime(beginTime);
            /* End time of pull in the format of `yyyymmddhh` accurate to the hour
             * Note: `EndTime` must be after `beginTime` */
            String endTime = "2019071123";
            req.setEndTime(endTime);

            /* Initialize the request by calling the `SendStatusStatistics` method
             * The returned `res` is an instance of the `SendStatusStatisticsRespon
            SendStatusStatisticsResponse res = client.SendStatusStatistics(req);

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

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

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.v20210111.SmsClient;
// Import the request response class corresponding to the request API
import com.tencentcloudapi.sms.v20210111.models.AddSmsTemplateRequest;
import com.tencentcloudapi.sms.v20210111.models.AddSmsTemplateResponse;
/**
* Tencent Cloud Sms AddSmsTemplate
*
*/
public class AddSmsTemplate
{
  public static void main( String[] args )
  {
      try {
          // Instantiate an authentication object, input parameters require Tencent
          // To protect key security, it is suggested to set keys in environment va
          // Hardcoding keys into the code might lead to exposure through leaked co
          // SecretId, SecretKey inquiry: https://console.tencentcloud.com/cam/capi
          // Credential cred = new Credential("SecretId", "SecretKey");
          Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"),

          // Instantiate an HTTP option, optional, can be skipped if there are no s
          HttpProfile httpProfile = new HttpProfile();
          // Starting from version 3.0.96, an HTTP proxy can be set separately (ign
          // httpProfile.setProxyHost("Real proxy IP");
          // httpProfile.setProxyPort(Real proxy port);
          httpProfile.setReqMethod("GET"); // GET request (the default value is POS
          httpProfile.setConnTimeout(30); // Request connection timeout time, in se
          httpProfile.setWriteTimeout(30);  // Sets write timeout time, in seconds
          httpProfile.setReadTimeout(30);  // Sets read timeout time, in seconds (t

          /* Specifies the access region domain name. The default nearby region dom
          httpProfile.setEndpoint("sms.tencentcloudapi.com");

          /* Optional steps:
           * Instantiate a client configuration object. You can specify the timeout
          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 information on the region you select in Te
          SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile);
          /* Instantiate a request object. You can further set the request paramete
           * You can directly check the SDK source code to determine which attribut
           * An attribute may be of a basic type or import another data structure
           * We recommend you use the IDE for development where you can easily redi
          AddSmsTemplateRequest req = new AddSmsTemplateRequest();
           /* Populate the request parameters. Here, the member variables of the re
           * You can view the definition of the request parameters in the API docum
           * Settings of a basic parameter:
           * Help link:
           * SMS console: https://console.tencentcloud.com/smsv2
           * sms helper：https://www.tencentcloud.com/document/product/382/3773 */
           /* Template name */
          String templatename = "Tencent Cloud";
          req.setTemplateName(templatename);
           /* Template content */
          String templatecontent = "Your login verification code is {1}. Please ent
          req.setTemplateContent(templatecontent);
           /* SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS */
          long smstype = 3;
          req.setSmsType(smstype);
           /* Whether it is Global SMS. 0: Mainland China SMS; 1: Global SMS */
          long international = 0;
          req.setInternational(international);
           /* Template remarks, such as reason for application and use case */
          String remark = "xxx";
          req.setRemark(remark);
           /* Initialize the request by calling the `AddSmsTemplate` method on the
           * The returned `res` is an instance of the `AddSmsTemplateResponse` clas
          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 retur
          System.out.println(res.getRequestId());
       } catch (TencentCloudSDKException e) {
          e.printStackTrace();
      }
  }
}
```

FAQs

java.lang.NoSuchMethodError: xxx.setSkipSign issue

During the usage, some users may encounter the error: `java.lang.NoSuchMethodError: xxx.setSkipSign`. This is due to a version mismatch between the `tencentcloud-sdk-java-common` package and the `tencentcloud-sdk-java-sms` package.

The issue could be caused by specifying an incorrect version of the `common` package in the pom, or by indirectly referencing an incompatible `common` version due to the reference to other third-party SDKs.

The solution is to explicitly specify the same version of the `common` package in pom.xml. For example, if the version of `tencentcloud-sdk-java-sms` is `3.1.1000` , you can specify the `common` package version like this:

```
<dependency>
    <groupId>com.tencentcloudapi</groupId>
    <artifactId>tencentcloud-sdk-java-common</artifactId>
    <version>3.1.1000</version>
</dependency>
```

Kotlin Issue

During the usage, some users might encounter the error: `java.lang.NoSuchMethodError: kotlin.collections.ArraysKt.copyInto` . This is due to the Kotlin runtime environment being of a lower version, and upgrading the Kotlin version might resolve the issue.

Failure to update dependencies in the repository's pom.xml file

This may be because the local server has a proxy configured, but the proxy is not configured for the tool during the update. Please update the dependencies on the command line as detailed above. If the problem persists, you need to check whether the problem is caused by network or firewall issues.

Failure to run the demo

In `[TencentCloudSDKException]message:java.net.ConnectException-Connection timed out: connect requestId:` , you need to check whether the local server has a proxy configured but the proxy is not added in the code. For more information, please see below:

```
HttpProfile httpProfile = new HttpProfile();
httpProfile.setProxyHost("actual proxy IP");
httpProfile.setProxyPort(real proxy port);
```

Version Upgrade

Please note that upgrading from v3.0.x to 3.1.x causes compatibility issues. As `integer` fields have been modified to `long` type, the project needs to be recompiled.

Certificate issue

Certificate issues are usually caused by incorrect configuration of the client environment. The SDK does not manipulate the certificate and relies on the processing by the Java runtime environment itself. After a certificate issue occurs, you can run `-Djavax.net.debug=ssl` to enable detailed logs for troubleshooting.

Some users reported that the certificate error `javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_failure` occurred while using IBM JDK 1.8. The error was resolved after Oracle JDK was used.

SDK for PHP

Last updated：2024-06-27 15:48:01

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:

API version required for connecting to Tencent Cloud International:

SMS API v2021-01-11 is required. For details, see the sample code.

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 learned about the concept of region and selected a region as needed.

You have activated SMS. For detailed directions, please see Getting Started.

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.**

The endpoint 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 from Github Repository.

Installing SDK

Installing via Composer

See Github repository:Installing via Composer

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 as shown below:

Sending SMS message

```
<?php
require_once '/path/to/vendor/autoload.php';
// Import the client of the corresponding product module
use TencentCloud\\Sms\\V20210111\\SmsClient;
// Import the `Request` class corresponding to the request API
use TencentCloud\\Sms\\V20210111\\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 {
    // Instantiate a certificate object. The input parameters require the Tencent C
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option
    $httpProfile = new HttpProfile();
    // Configure the proxy
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET");  // GET request (POST request by default)
    $httpProfile->setReqTimeout(10);    // Request timeout period in seconds (60 se
    $httpProfile->setEndpoint("sms.tencentcloudapi.com");  // Specify the access re

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

    // Instantiate the client object of the requested product (with SMS as an examp
    // The second parameter is the information on the region you select in Tencent 
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

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

    /* Populate the request parameters. Here, the member variables of the request o
     * You can view the definition of the request parameters in the API documentati
     * Settings of a basic parameter:
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper: https://www.tencentcloud.com/document/product/382/3773 */

    /* SMS application ID, which is the `SdkAppId` generated after an application i
    $req->SmsSdkAppId = "2400006666";
    /* SMS signature content, which should be encoded in UTF-8. You must enter an a
    $req->SignName = "xxx";
    /* SMS code number extension, which is not activated by default. If you need to
    $req->ExtendCode = "";
    /* Target mobile number in the E.164 standard (+[country/region code][mobile nu
     * Example: +60198890000, which has a + sign followed by 60 (country/region cod
    $req->PhoneNumberSet = array("+60198890000");
    /* `SenderId` for Global SMS, which is not activated by default. If you need to
    $req->SenderId = "";
    /* User session content, which can carry context information such as user-side 
    $req->SessionContext = "xxx";
    /* Template ID. You must enter the ID of an approved template, which can be vie
    $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.
    // The returned `resp` is an instance of the `SendSmsResponse` class which corr
    $resp = $client->SendSms($req);

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

    // You can also take a single value
    // You can view the definition of the return field in the API documentation at
    print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

Pulling receipt status

```
<?php
require_once '/path/to/vendor/autoload.php';
// Import the client of the corresponding product module
use TencentCloud\\Sms\\V20210111\\SmsClient;
// Import the `Request` class corresponding to the request API
use TencentCloud\\Sms\\V20210111\\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 {
    // Instantiate a certificate object. The input parameters require the Tencent C
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option
    $httpProfile = new HttpProfile();
    // Configure the proxy
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET");  // GET request (POST request by default)
    $httpProfile->setReqTimeout(30);    // Request timeout period in seconds (60 se
    $httpProfile->setEndpoint("sms.tencentcloudapi.com");  // Specify the access re

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

    // Instantiate the client object of the requested product (with SMS as an examp
    // The second parameter is the information on the region you select in Tencent
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

    // Instantiate an SMS message sending request object. Each API corresponds to a
    $req = new PullSmsSendStatusRequest();
    /* Populate the request parameters. Here, the member variables of the request o
     * You can view the definition of the request parameters in the API documentati
     * Settings of a basic parameter:
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper: https://www.tencentcloud.com/document/product/382/3773 */

    /* SMS application ID, which is the `SdkAppId` generated after an application i
    $req->SmsSdkAppId = "2400006666";
    /* Maximum number of pulled entries. Maximum value: 100 */
    $req->Limit = 10;


    // Initialize the request by calling the `PullSmsSendStatus` method on the clie
    // The returned `resp` is an instance of the `PullSmsSendStatusResponse` class
    $resp = $client->PullSmsSendStatus($req);

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

    // You can also take a single value
    // You can view the definition of the return field in the API documentation at
    print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

Collecting SMS message sending data

```
<?php
require_once '/path/to/vendor/autoload.php';
// Import the client of the corresponding product module
use TencentCloud\\Sms\\V20210111\\SmsClient;
// Import the `Request` class corresponding to the request API
use TencentCloud\\Sms\\V20210111\\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 {
    // Instantiate a certificate object. The input parameters require the Tencent C
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option
    $httpProfile = new HttpProfile();
    // Configure the proxy
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET");  // GET request (POST request by default)
    $httpProfile->setReqTimeout(30);    // Request timeout period in seconds (60 se
    $httpProfile->setEndpoint("sms.tencentcloudapi.com");  // Specify the access re

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

    // Instantiate the client object of the requested product (with SMS as an examp
    // The second parameter is the information on the region you select in Tencent
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

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

    /* Populate the request parameters. Here, the member variables of the request o
     * You can view the definition of the request parameters in the API documentati
     * Settings of a basic parameter:
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper: https://www.tencentcloud.com/document/product/382/3773 */

    /* SMS application ID, which is the `SdkAppId` generated after an application i
    $req->SmsSdkAppId = "2400006666";
    /* 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->BeginTime = "2019122500";
    /* End time of pull in the format of `yyyymmddhh` accurate to the hour
     * Note: `EndTime` must be after `BeginTime` */
    $req->EndTime = "2019122523";

    // Initialize the request by calling the `SendStatusStatistics` method on the c
    // The returned `resp` is an instance of the `SendStatusStatisticsResponse` cla
    $resp = $client->SendStatusStatistics($req);
    // A string return packet in JSON format is output
    print_r($resp->toJsonString());

    // You can also take a single value
    // You can view the definition of the return field in the API documentation at
    print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

Applying for SMS template

```
<?php
require_once '/path/to/vendor/autoload.php';
// Import the client of the SMS module
use TencentCloud\\Sms\\V20210111\\SmsClient;
// Import the `Request` class corresponding to the API to be requested
use TencentCloud\\Sms\\V20210111\\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 {
    // Instantiate a certificate object. The input parameters require the Tencent C
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // Instantiate an HTTP option (optional; skip if there are no special requireme
    $httpProfile = new HttpProfile();
    // Configure the proxy
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET");  // GET request (POST request by default)
    $httpProfile->setReqTimeout(30);    // Request timeout period in seconds (60 se
    $httpProfile->setEndpoint("sms.tencentcloudapi.com");  // Specify the access re

     // Instantiate a client option (optional; skip if no special requirements are
    $clientProfile = new ClientProfile();
    $clientProfile->setSignMethod("TC3-HMAC-SHA256");  // Specify the signature alg
    $clientProfile->setHttpProfile($httpProfile);
    // Instantiate an SMS client object. `clientProfile` is optional
    // The second parameter is the information on the region you select in Tencent
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

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

    /* Populate the request parameters. Here, the member variables of the request o
     * You can view the definition of the request parameters in the API documentati
     * Settings of a basic parameter:
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper：https://www.tencentcloud.com/document/product/382/3773
     */

    /* Template name */
    $req->TemplateName = "Tencent Cloud";
    /* Template content */
    $req->TemplateContent = "Your login verification code is {1}. Please enter it w
    /* SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS */
    $req->SmsType = 3;
    /* 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
    $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
    print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

FAQs

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` may occur, which can be solved as follows:
1. Download the certificate file `cacert.pem` at https://curl.haxx.se/ca/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.

Web access exception

The command is executed normally on the command line, but when it is executed on the web server, the following error is reported:

`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 this value, and then check whether this error is fixed.

Problem with installation through source code

In order to satisfy the need for installation through source code, we previously put the dependent package files in the `vendor` directory. However, considering that incompatibility with Composer should not be caused, we had to forbid importing the `vendor` directory on GitHub, which resulted in the problem where the `git clone` command had to be used to get the `vendor` directory. This practice caused confusion for some users not familiar with GitHub. Therefore, starting from v3.0.188, we have temporarily removed the method of installation through source code, and Composer must be used to install the SDK and dependent packages.

Proxy settings

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);
```

SDK for Python

Last updated：2024-06-27 15:48:00

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:

API version required for connecting to Tencent Cloud International:

SMS API v2021-01-11 is required. For details, see the sample code.

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 learned about the concept of region and selected a region as needed.

You have activated SMS. For detailed directions, please see Getting Started.

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.**

The endpoint 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 from Github Repository.

Installing SDK

Installing via Pip (recommended)

See Github Repository: Installing via Pip (recommended)

Installing via source package

See Github Repository: Installing via source package

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 as shown below:

Sending SMS message

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# Import the client models of the corresponding product module.
from tencentcloud.sms.v20210111 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:
    # Instantiate an authentication object, the input parameters must include the T
    # To protect key security, it is suggested to set keys in environment variables
    # Hardcoding keys into the code might lead to exposure through leaked code, pos

    # Query SecretId, SecretKey: https://console.tencentcloud.com/cam/capi
    # cred = credential.Credential("secretId", "secretKey")
    cred = credential.Credential(
        os.environ.get("TENCENTCLOUD_SECRET_ID"),
        os.environ.get("TENCENTCLOUD_SECRET_KEY"))

    # (Optional) Instantiate an HTTP option
    httpProfile = HttpProfile()
    # If you need to specify the proxy for API access, you can initialize HttpProfi
    # httpProfile = HttpProfile(proxy="http://username:password@proxy IP:proxy port
    httpProfile.reqMethod = "POST"  # POST request (POST request by default)
    httpProfile.reqTimeout = 10    # Request timeout period in seconds (60 seconds
    httpProfile.endpoint = "sms.tencentcloudapi.com"  # Specify the access region d
```

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

        # Instantiate the client object of the requested product (with SMS as an exampl
        # The second parameter is the information on the region you select in Tencent C
        client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

        # Instantiate a request object. You can further set the request parameters acco
        # You can directly check the SDK source code to determine which attributes of `
        # An attribute may be of a basic type or import another data structure
        # We recommend you use the IDE for development where you can easily redirect to
        req = models.SendSmsRequest()

        # Settings of a basic parameter:
        # The SDK uses the pointer style to specify parameters, so even for basic param
        # The SDK provides encapsulation functions for importing the pointers of basic
        # Help link:
        # SMS console: https://console.tencentcloud.com/smsv2
        # sms helper: https://www.tencentcloud.com/document/product/382/3773

        # SMS application ID, which is the `SdkAppId` generated after an application is
        req.SmsSdkAppId = "2400006666"
        # SMS signature content, which should be encoded in UTF-8. You must enter an ap
        req.SignName = "xxx"
        # SMS code number extension, which is not activated by default. If you need to
        req.ExtendCode = ""
        # User session content, which can carry context information such as user-side I
        req.SessionContext = "xxx"
        # `senderid` for Global SMS, which is not activated by default. If you need to
        req.SenderId = ""
        # Target mobile number in the E.164 standard (+[country/region code][mobile num
        # Example: +60198890000, which has a + sign followed by 60 (country/region code
        req.PhoneNumberSet = ["+60198890000"]
        # Template ID. You must enter the ID of an approved template, which can be view
        req.TemplateId = "449739"
        # Template parameters. If there are no template parameters, leave it empty
        req.TemplateParamSet = ["666"]


        # Initialize the request by calling the `DescribeInstances` method on the clien
        # The returned `resp` is an instance of the `DescribeInstancesResponse` class w
        resp = client.SendSms(req)
```

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

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

Pulling SMS delivery status

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# Import the client models of the corresponding product module.
from tencentcloud.sms.v20210111 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:
    # Instantiate an authentication object, the input parameters must include the T
    # To protect key security, it is suggested to set keys in environment variables
    # Hardcoding keys into the code might lead to exposure through leaked code, pos
    # Query SecretId, SecretKey: https://console.tencentcloud.com/cam/capi
    # cred = credential.Credential("secretId", "secretKey")
    cred = credential.Credential(
        os.environ.get("TENCENTCLOUD_SECRET_ID"),
        os.environ.get("TENCENTCLOUD_SECRET_KEY"))

    # (Optional) Instantiate an HTTP option
    httpProfile = HttpProfile()
    # If you need to specify the proxy for API access, you can initialize HttpProfi
    # httpProfile = HttpProfile(proxy="http://username:password@proxy IP:proxy port
    httpProfile.reqMethod = "POST"  # POST request (POST request by default)
    httpProfile.reqTimeout = 30    # Request timeout period in seconds (60 seconds
    httpProfile.endpoint = "sms.tencentcloudapi.com"  # Specify the access region d

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

    # Instantiate the client object of the requested product (with SMS as an exampl
    # The second parameter is the information on the region you select in Tencent C
    client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)
    # Instantiate a request object. You can further set the request parameters acco
    # You can directly check the SDK source code to determine which attributes of `
    # An attribute may be of a basic type or import another data structure
    # We recommend you use the IDE for development where you can easily redirect to
    req = models.PullSmsSendStatusRequest()

    # Settings of a basic parameter:
    # The SDK uses the pointer style to specify parameters, so even for basic param
    # The SDK provides encapsulation functions for importing the pointers of basic
    # Help link:
    # SMS console: https://console.tencentcloud.com/smsv2
    # sms helper: https://www.tencentcloud.com/document/product/382/3773

    # SMS application ID, which is the `SdkAppId` generated after an application is
    req.SmsSdkAppId = "2400006666"
    # Maximum number of pulled entries. Maximum value: 100
    req.Limit = 10

    # Initialize the request by calling the `PullSmsSendStatus` method on the clien
    # The returned `resp` is an instance of the `PullSmsSendStatusResponse` class w
    resp = client.PullSmsSendStatus(req)

    # A string return packet in JSON format is outputted
    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 TencentCloudS
# Import the client models of the corresponding product module.
from tencentcloud.sms.v20210111 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:
    # Instantiate an authentication object, the input parameters must include the T
    # To protect key security, it is suggested to set keys in environment variables
    # Hardcoding keys into the code might lead to exposure through leaked code, pos
    # Query SecretId, SecretKey: https://console.tencentcloud.com/cam/capi
    # cred = credential.Credential("secretId", "secretKey")
    cred = credential.Credential(
        os.environ.get("TENCENTCLOUD_SECRET_ID"),
        os.environ.get("TENCENTCLOUD_SECRET_KEY"))

    # (Optional) Instantiate an HTTP option
    httpProfile = HttpProfile()
    # If you need to specify the proxy for API access, you can initialize HttpProfi
    # httpProfile = HttpProfile(proxy="http://username:password@proxy IP:proxy port
    httpProfile.reqMethod = "POST"  # POST request (POST request by default)
    httpProfile.reqTimeout = 30    # Request timeout period in seconds (60 seconds
    httpProfile.endpoint = "sms.tencentcloudapi.com"  # Specify the access region d

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

    # Instantiate the client object of the requested product (with SMS as an exampl
    # The second parameter is the information on the region you select in Tencent C
    client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

    # Instantiate a request object. You can further set the request parameters acco
    # You can directly check the SDK source code to determine which attributes of `
    # An attribute may be of a basic type or import another data structure
    # We recommend you use the IDE for development where you can easily redirect to
    req = models.SendStatusStatisticsRequest()

    # Settings of a basic parameter:
    # The SDK uses the pointer style to specify parameters, so even for basic param
    # The SDK provides encapsulation functions for importing the pointers of basic
    # Help link:
    # SMS console: https://console.tencentcloud.com/smsv2
    # sms helper: https://www.tencentcloud.com/document/product/382/3773

    # SMS application ID, which is the `SmsSdkAppId` generated after an application
    req.SmsSdkAppId = "2400006666"
    # 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.BeginTime = "2019122400"
    # End time of pull in the format of `yyyymmddhh` accurate to the hour
    # Note: `EndTime` must be after `BeginTime`
    req.EndTime = "2019122523"
    # Initialize the request by calling the `SendStatusStatistics` method on the cl
    # The returned `resp` is an instance of the `SendStatusStatisticsResponse` clas
    resp = client.SendStatusStatistics(req)

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

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

Applying for SMS template

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# Import the client models of the corresponding product module.
from tencentcloud.sms.v20210111 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:
    # Instantiate an authentication object, the input parameters must include the T
    # To protect key security, it is suggested to set keys in environment variables
    # Hardcoding keys into the code might lead to exposure through leaked code, pos
    # Query SecretId, SecretKey: https://console.tencentcloud.com/cam/capi
    # cred = credential.Credential("secretId", "secretKey")
    cred = credential.Credential(
        os.environ.get("TENCENTCLOUD_SECRET_ID"),
        os.environ.get("TENCENTCLOUD_SECRET_KEY"))

    # Instantiate an HTTP option (optional; skip if there are no special requiremen
    httpProfile = HttpProfile()
    # If you need to specify the proxy for API access, you can initialize HttpProfi
    # httpProfile = HttpProfile(proxy="http://username:password@proxy IP:proxy port
    httpProfile.reqMethod = "POST"  # POST request (POST request by default)
    httpProfile.reqTimeout = 30    # Request timeout period in seconds (60 seconds 
    httpProfile.endpoint = "sms.tencentcloudapi.com"  # Specify the access region d

    # Optional steps:
    # Instantiate a client configuration object. You can specify the timeout period
    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 information on the region you select in Tencent C
    client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

    # Instantiate a request object. You can further set the request parameters acco
    # You can directly check the SDK source code to determine which attributes of `
    # An attribute may be of a basic type or import another data structure
    # We recommend you use the IDE for development where you can easily redirect to
    req = models.AddSmsTemplateRequest()

    # Settings of a basic parameter:
    # The SDK uses the pointer style to specify parameters, so even for basic param
    # The SDK provides encapsulation functions for importing the pointers of basic 
    # Help link:
    # SMS console: https://console.tencentcloud.com/smsv2
    # sms helper: https://www.tencentcloud.com/document/product/382/3773

    # Template name
    req.TemplateName = "Tencent Cloud"
    # Template content
    req.TemplateContent = "Your login verification code is {1}. Please enter it wit
    # SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS
    req.SmsType = 3
    # 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 o
    # The returned `resp` is an instance of the `AddSmsTemplateResponse` class whic
    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)
```

FAQs

Certificate issue

When you install Python 3.6 or above on macOS, you may encounter a certificate error: `Error: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1056).`

This is because that on macOS, Python no longer uses the system's default certificate and does not provide a certificate itself. When an HTTPS request is made, the certificate provided by the `certifi` library needs to be used, but the SDK does not support specifying it; therefore, you can only solve this problem by installing the certificate with the `sudo "/Applications/Python 3.6/Install Certificates.command"` command.

Although this problem should not occur in Python 2, there may be similar situations in specific user environments, which can also be solved with `sudo /Applications/Python 2.7/Install Certificates.command` .

Proxy settings

If there is a proxy in your environment, you can set the proxy in the following two ways:

Specify the proxy when initializing HttpProfile. For more information, please see the example.

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.

SDK for Node.js

Last updated：2024-06-27 15:48:00

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:

API version required for connecting to Tencent Cloud International:

SMS API v2021-01-11 is required. For details, see the sample code.

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 learned about the concept of region and selected a region as needed.

You have activated SMS. For detailed directions, please see Getting Started.

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.**

The endpoint 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 from Github Repository.

Installing SDK

Installing via npm

See Github Repository: Installing via npm

Installing via source package

See Github Repository: Installing via source package

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 as shown below:

Sending SMS message

```
const tencentcloud = require("tencentcloud-sdk-nodejs")

// Import the client models of the corresponding product module
const smsClient = tencentcloud.sms.v20210111.Client

/* Instantiate the client object of the requested product (with SMS as an example)
const client = new smsClient({
  credential: {
    /* To protect key security, it is suggested to set keys in environment variable
    * Hardcoding keys into the code might lead to exposure through leaked code, pos
    * Query the CAM Key: https://console.tencentcloud.com/cam/capi */
    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  /* Required: region information. The parameter is the information on the region y
  region: "ap-singapore",
  /* Optional:
   * Client configuration object. You can specify the timeout period and other conf
  profile: {
    /* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field
    signMethod: "HmacSHA256",
    httpProfile: {
      reqMethod: "POST", // Request Method
      reqTimeout: 10, // Request timeout time, in seconds (the default value is 60
      endpoint: "sms.tencentcloudapi.com"
    },
  },
})
```

```
/* Request parameter. You can further set the request parameters according to the A
 * An attribute may be of a basic type or import another data structure
 * We recommend you use the IDE for development where you can easily redirect to an
const params = {
  /* SMS application ID, which is the `SdkAppId` generated after an application is
  SmsSdkAppId: "2400006666",
  /* SMS signature content, which should be encoded in UTF-8. You must enter an app
  SignName: "xxx",
  /* SMS code number extension, which is not activated by default. If you need to a
  ExtendCode: "",
  /* `senderid` for Global SMS, which is not activated by default. If you need to a
  SenderId: "",
  /* User session content, which can carry context information such as user-side ID
  SessionContext: "",
  /* Target mobile number in the e.164 standard (+[country/region code][mobile numb
   * Example: +60198890000, which has a + sign followed by 60 (country/region code)
  PhoneNumberSet: ["+60198890000"],
  /* Template ID. You must enter the ID of an approved template, which can be viewe
  TemplateId: "449739",
  /* Template parameters. If there are no template parameters, leave it empty */
  TemplateParamSet: ["666"],
}
// Call the API you want to access through the client object; you need to pass in t
client.SendSms(params, 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)
})
```

Pulling receipt status

```
const tencentcloud = require("tencentcloud-sdk-nodejs")

// Import the client models of the corresponding product module
const smsClient = tencentcloud.sms.v20210111.Client

/* Instantiate the client object of the requested product (with SMS as an example)
const client = new smsClient({
  credential: {
    /* To protect key security, it is suggested to set keys in environment variable
     * Hardcoding keys into the code might lead to exposure through leaked code, pos
    * Query the CAM Key: https://console.tencentcloud.com/cam/capi */
    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  /* Required: region information. The parameter is the information on the region y
  region: "ap-singapore",
  /* Optional:
   * Client configuration object. You can specify the timeout period and other conf
  profile: {
    /* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field
    signMethod: "HmacSHA256",
    httpProfile: {
      reqMethod: "POST", // Request Method
      reqTimeout: 30, // Request timeout time, in seconds (the default value is 60
      endpoint: "sms.tencentcloudapi.com"
    },
  },
})

/* Request parameter. You can further set the request parameters according to the A
 * An attribute may be of a basic type or import another data structure
 * We recommend you use the IDE for development where you can easily redirect to an
const params = {
  // SMS application ID, which is the `SdkAppId` generated after an application is
  SmsSdkAppId: "2400006666",
  // Maximum number of pulled entries. Maximum value: 100
  Limit: 10,
}
// Call the API you want to access through the client object; you need to pass in t
client.PullSmsSendStatus(params, 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)
})
```

Collecting SMS message sending data

```
const tencentcloud = require("tencentcloud-sdk-nodejs")

// Import the client models of the corresponding product module
const smsClient = tencentcloud.sms.v20210111.Client
```

```
/* Instantiate the client object of the requested product (with SMS as an example)
const client = new smsClient({
  credential: {
    /* To protect key security, it is suggested to set keys in environment variable
    * Hardcoding keys into the code might lead to exposure through leaked code, pos
    * Query the CAM Key: https://console.tencentcloud.com/cam/capi */
    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  /* Required: region information. The parameter is the information on the region y
  region: "ap-singapore",
  /* Optional:
   * Client configuration object. You can specify the timeout period and other conf
  profile: {
    /* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field
    signMethod: "HmacSHA256",
    httpProfile: {
      reqMethod: "POST", // Request Method
      reqTimeout: 30, // Request timeout time, in seconds (the default value is 60
      endpoint: "sms.tencentcloudapi.com"
    },
  },
})

/* Request parameter. You can further set the request parameters according to the A
 * An attribute may be of a basic type or import another data structure
 * We recommend you use the IDE for development where you can easily redirect to an
const params = {
    // SMS application ID, which is the `SdkAppId` generated after an application i
    SmsSdkAppId: "2400006666",
    // Maximum number of pulled entries. Maximum value: 100
    Limit: 10,
    // Offset. Note: this parameter is currently fixed at 0
    Offset: 0,
    // Start time of pull in the format of `yyyymmddhh` accurate to the hour
    BeginTime: "2019122400",
    // End time of pull in the format of `yyyymmddhh` accurate to the hour
    // Note: `EndTime` must be after `BeginTime`
    EndTime: "2019122523",
}
// Call the API you want to access through the client object; you need to pass in t
client.SendStatusStatistics(params, 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)
})
```

Applying for SMS template

```
const tencentcloud = require("tencentcloud-sdk-nodejs")

// Import the client models of the corresponding product module
const smsClient = tencentcloud.sms.v20210111.Client

/* Instantiate the client object of the requested product (with SMS as an example)
const client = new smsClient({
  credential: {
    /* To protect key security, it is suggested to set keys in environment variable
    * Hardcoding keys into the code might lead to exposure through leaked code, pos
    * Query the CAM Key: https://console.tencentcloud.com/cam/capi */
    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  /* Required: region information. The parameter is the information on the region y
  region: "ap-singapore",
  /* Optional:
   * Client configuration object. You can specify the timeout period and other conf
  profile: {
    /* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field
    signMethod: "HmacSHA256",
    httpProfile: {
      reqMethod: "POST", // Request Method
      reqTimeout: 30, // Request timeout, default is 60s
      endpoint: "sms.tencentcloudapi.com"
    },
  },
})

/* Request parameter. You can further set the request parameters according to the A
 * An attribute may be of a basic type or import another data structure
 * We recommend you use the IDE for development where you can easily redirect to an
const params = {
  /* Template name */
  TemplateName: "Tencent Cloud",
  /* Template content */
  TemplateContent: "Your login verification code is {1}. Please enter it within {2}
  /* SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS */
  SmsType: 3,
  /* Whether it is Global SMS. 0: Mainland China SMS; 1: Global SMS */
  International: 0,
  /* Template remarks, such as reason for application and use case */
  Remark: "xxx",
}
// Call the API you want to access through the client object; you need to pass in t
client.AddSmsTemplate(params, 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)
})
```

FAQs

Proxy settings

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.

SDK for C#

Last updated：2024-06-27 15:48:01

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:

API version required for connecting to Tencent Cloud International:

SMS API v2021-01-11 is required. For details, see the sample code.

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 learned about the concept of region and selected a region as needed.

You have activated SMS. For detailed directions, please see Getting Started.

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.**

The endpoint 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 from Github Repository.

Installing via NuGet (recommended)

See Github Repository: Installing via NuGet (recommended)

Installing via source code

See Github Repository: Installing via source 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 as shown below:

Sending SMS message

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20210111;
using TencentCloud.Sms.V20210111.Models;

namespace TencentCloudExamples
{
    class SendSms
    {
        static void Main(string[] args)
        {
            try
            {
                // Required steps:
                // Instantiate an authentication object. The input parameters need
                // To protect key security, it is suggested to set keys in environm
                // Hardcoding keys into the code might lead to exposure through lea
                // The example here uses the way to read from the environment varia
                // For SecretId, SecretKey inquiry: https://console.tencentcloud.co
                Credential cred = new Credential {
                    SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SEC
                    SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SE
                };

                /* Optional steps:
                 * Instantiate a client configuration object. You can specify the t
                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 t
                HttpProfile httpProfile = new HttpProfile();
                /* The SDK uses the POST method by default
                 * If you have to use the GET method, you can set it here, but the
                httpProfile.ReqMethod = "GET";
                httpProfile.Timeout = 10; // Request connection timeout period in s
                /* The SDK automatically specifies the domain name. Generally, you
                 * you need to manually specify the domain name. For example, the S
                httpProfile.Endpoint = "sms.tencentcloudapi.com";
                // Proxy server. Set it when there is a proxy server in your enviro
                // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS

                clientProfile.HttpProfile = httpProfile;
                /* Instantiate the client object of the requested product (with SMS
                 * The second parameter is the information on the region you select
                SmsClient client = new SmsClient(cred, "ap-singapore", clientProfil

                /* Instantiate a request object. You can further set the request pa
                 * You can directly check the SDK source code to determine which at
                 * An attribute may be of a basic type or import another data struc
                 * We recommend you use the IDE for development where you can easil
                SendSmsRequest req = new SendSmsRequest();

                /* Settings of a basic parameter:
                 * The SDK uses the pointer style to specify parameters, so even fo
                 * The SDK provides encapsulation functions for importing the point
                 * Help link:
                 * SMS console: https://console.tencentcloud.com/smsv2
                 * sms helper: https://www.tencentcloud.com/document/product/382/37

                req.SmsSdkAppId = "2400006666";
                /* SMS signature content, which should be encoded in UTF-8. You mus
                req.SignName = "xxx";
                /* SMS code number extension, which is not activated by default. If
                req.ExtendCode = "";
                /* `senderid` for Global SMS, which is not activated by default. If
                req.SenderId = "";
                /* User session content, which can carry context information such a
                req.SessionContext = "";
                /* Target mobile number in the E.164 standard (+[country/region cod
                 * Example: +60198890000, which has a + sign followed by 60 (countr
                req.PhoneNumberSet = new String[] {"+60198890000"};
                /* Template ID. You must enter the ID of an approved template, whic
                req.TemplateId = "449739";
                /* Template parameters. If there are no template parameters, leave
                req.TemplateParamSet = new String[] {"666"};
                // Initialize the request by calling the `DescribeInstances` method
                // The returned `resp` is an instance of the `DescribeInstancesResp
                SendSmsResponse resp = client.SendSmsSync(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.V20210111;
using TencentCloud.Sms.V20210111.Models;
namespace TencentCloudExamples
{
  class PullSmsSendStatus
  {
      static void Main(string[] args)
      {
          try
          {
              // Required steps:
              // Instantiate an authentication object. The input parameters need Te
              // To protect key security, it is suggested to set keys in environmen
              // Hardcoding keys into the code might lead to exposure through leake
              // The example here uses the way to read from the environment variabl
              // For SecretId, SecretKey inquiry: https://console.tencentcloud.com/
              Credential cred = new Credential {
                  SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRE
                  SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECR
              };
              /* Optional steps:
               * Instantiate a client configuration object. You can specify the tim
```

```
                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 tim
                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 GE
                httpProfile.ReqMethod = "POST";
                httpProfile.Timeout = 30; // Request connection timeout period in sec
                /* The SDK automatically specifies the domain name. Generally, you do
                 * For example, the SMS domain name of the Shanghai Finance Zone is `
                httpProfile.Endpoint = "sms.tencentcloudapi.com";
                // Proxy server. Set it when there is a proxy server in your environm
                // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_P
                clientProfile.HttpProfile = httpProfile;
                /* Instantiate an SMS client object
                 * The second parameter is the information on the region you select i
                SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile)
                /* Instantiate a request object. You can further set the request para
                 * You can directly check the SDK source code to determine which attr
                 * An attribute may be of a basic type or import another data structu
                 * We recommend you use the IDE for development where you can easily 
                PullSmsSendStatusRequest req = new PullSmsSendStatusRequest();

                /* Settings of a basic parameter:
                 * The SDK uses the pointer style to specify parameters, so even for 
                 * The SDK provides encapsulation functions for importing the pointer
                 * Help link:
                 * SMS console: https://console.tencentcloud.com/smsv2
                 * sms helper: https://www.tencentcloud.com/document/product/382/3773

                // Set the maximum number of pulled entries. Maximum value: 100
                req.Limit = 100;
                /* SMS application ID, which is the `SdkAppId` generated after an app
                req.SmsSdkAppId = "2400006666";
                // Initialize the request by calling the `PullSmsSendStatus` method o
                // The returned `resp` is an instance of the `PullSmsSendStatusRespon
                PullSmsSendStatusResponse resp = client.PullSmsSendStatusSync(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.V20210111;
using TencentCloud.Sms.V20210111.Models;
namespace TencentCloudExamples
{
   class SendStatusStatistics
   {
       static void Main(string[] args)
       {
           try
           {
               // Required steps:
               // Instantiate an authentication object. The input parameters need T
               // To protect key security, it is suggested to set keys in environme
               // Hardcoding keys into the code might lead to exposure through leak
               // The example here uses the way to read from the environment variab
               // For SecretId, SecretKey inquiry: https://console.tencentcloud.com
               Credential cred = new Credential {
                   SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECR
                   SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SEC
               };
               /* Optional steps:
                * Instantiate a client configuration object. You can specify the ti
               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 ti
               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 G
               httpProfile.ReqMethod = "POST";
               httpProfile.Timeout = 30; // Request connection timeout period in se
               /* The SDK automatically specifies the domain name. Generally, you d
                * For example, the SMS domain name of the Shanghai Finance Zone is
               httpProfile.Endpoint = "sms.tencentcloudapi.com";
               // Proxy server. Set it when there is a proxy server in your environ
               // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_
               clientProfile.HttpProfile = httpProfile;
               /* Instantiate an SMS client object
                * The second parameter is the information on the region you select
               SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile
               /* Instantiate a request object. You can further set the request par
                * You can directly check the SDK source code to determine which att
                * An attribute may be of a basic type or import another data struct
                * We recommend you use the IDE for development where you can easily
               SendStatusStatisticsRequest req = new SendStatusStatisticsRequest();

               /* Settings of a basic parameter:
                * The SDK uses the pointer style to specify parameters, so even for
                * The SDK provides encapsulation functions for importing the pointe
                * Help link:
                * SMS console: https://console.tencentcloud.com/smsv2
                * sms helper: https://www.tencentcloud.com/document/product/382/377
                */

               /* SMS application ID, which is the `SdkAppId` generated after an ap
               req.SmsSdkAppId = "2400006666";
               // 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
               req.BeginTime = "2019071100";
               /* End time of pull in the format of `yyyymmddhh` accurate to the ho
                * Note: `EndTime` must be after `BeginTime` */
               req.EndTime = "2019071123";
               // Initialize the request by calling the `SendStatusStatistics` meth
               // The returned `resp` is an instance of the `SendStatusStatisticsRe
               SendStatusStatisticsResponse resp = client.SendStatusStatisticsSync(
               // A string return packet in JSON format is output
               Console.WriteLine(AbstractModel.ToJsonString(resp));
           }
           catch (Exception e)
           {
               Console.WriteLine(e.ToString());
           }
           Console.Read();
       }
   }
}
```

Applying for SMS template

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20210111;
using TencentCloud.Sms.V20210111.Models;
namespace TencentCloudExamples
{
  class AddSmsTemplate
  {
      static void Main(string[] args)
      {
          try
          {
              // Required steps:
              // Instantiate an authentication object. The input parameters need Te
              // To protect key security, it is suggested to set keys in environmen
              // Hardcoding keys into the code might lead to exposure through leake
              // The example here uses the way to read from the environment variabl
              // For SecretId, SecretKey inquiry: https://console.tencentcloud.com/
              Credential cred = new Credential {
                  SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRE
                  SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECR
              };
              /* Optional steps:
               * Instantiate a client configuration object. You can specify the tim
              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 tim
              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 GE
              httpProfile.ReqMethod = "GET";
              httpProfile.Timeout = 30; // Request connection timeout period in sec
              /* The SDK automatically specifies the domain name. Generally, you do
               * For example, the SMS domain name of the Shanghai Finance Zone is `
              httpProfile.Endpoint = "sms.tencentcloudapi.com";
              // Proxy server. Set it when there is a proxy server in your environm
              // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_P
              clientProfile.HttpProfile = httpProfile;
              /* Instantiate an SMS client object
               * The second parameter is the information on the region you select i
              SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile)
              /* Instantiate a request object. You can further set the request para
               * You can directly check the SDK source code to determine which attr
               * An attribute may be of a basic type or import another data structu
               * We recommend you use the IDE for development where you can easily
              AddSmsTemplateRequest req = new AddSmsTemplateRequest();

              /* Settings of a basic parameter:
               * The SDK uses the pointer style to specify parameters, so even for
               * The SDK provides encapsulation functions for importing the pointer
               * Help link:
               * SMS console: https://console.tencentcloud.com/smsv2
               * sms helper: https://www.tencentcloud.com/document/product/382/3773
               */

              /* Template name */
              req.TemplateName = "Tencent Cloud";
              /* Template content */
              req.TemplateContent = "Your login verification code is {1}. Please en
              /* SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS */
              req.SmsType = 3;
              /* 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 t
              // The returned `resp` is an instance of the `AddSmsTemplateResponse`
              AddSmsTemplateResponse resp = client.AddSmsTemplateSync(req);
              // A string return packet in JSON format is output
              Console.WriteLine(AbstractModel.ToJsonString(resp));
          }
          catch (Exception e)
          {
              Console.WriteLine(e.ToString());
          }
          Console.Read();
      }
  }
}
```

FAQs

Proxy settings

If you use the SDK to call an API in a proxy environment, you need to set the system environment variable `https_proxy` (as shown in the sample code); otherwise, it may not be called normally, and a connection timeout exception will be thrown.

Sync and async calls

The new version of the SDK provides both async and sync APIs. The sync APIs uniformly have the `Sync` suffix after the async APIs as demonstrated in the code above.

Note:

In the sample, as it is a console application, you can call the async APIs synchronously, i.e., `ConfigureAwait(false).GetAwaiter().GetResult()`. When developing ASP applications or Windows Forms applications, you cannot call the async APIs synchronously in the response method of UI controls; otherwise, the UI will stop responding.

The solution is to change the response method of the UI controls to async and pay attention to the sync context. In addition, as async call immediately returns control to the user, it is prone to cause the user to click multiple times or perform unexpected operations. Such problems should be avoided in the program. For the source code, please refer to the `WindowsFormsDemo` project.

Dependent version

Note：

The `FluentClient` on which the SDK depends is on v3.2, but this package is currently available on v4.0, which is incompatible with lower versions. Upgrading this package to v4.0 in NuGet will cause problems such as inability to call or call failure.

SDK for Go

Last updated：2024-06-27 15:48:00

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:

API version required for connecting to Tencent Cloud International:

SMS API v2021-01-11 is required. For details, see the sample code.

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 learned about the concept of region and selected a region as needed.

You have activated SMS. For detailed directions, please see Getting Started.

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.**

The endpoint 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 from Github Repository.

Installing SDK

Installing via go get (recommended)

See Github repository: Installing via go get (recommended)

Installing via source code

See Github Repository: Installing via source 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 as shown below:

Sending SMS message

```
package main

import (
    "encoding/json"
    "fmt"
    "os"

    "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/v20210111" //
)

func main() {
    // Required steps:
    // Instantiate an authentication object. The input parameters need Tencent Clou
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* Optional steps:
     * Instantiate a client configuration object. You can specify the timeout perio
    cpf := profile.NewClientProfile()
```

```
        /* The SDK uses the POST method by default
         * If you have to use the GET method, you can set it here, but the GET method c
        cpf.HttpProfile.ReqMethod = "POST"
        cpf.HttpProfile.ReqTimeout = 10 // Request timeout time, in seconds (the defaul
        /* Specifies the access region domain name. The default nearby region domain na
        cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

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

        /* Instantiate the client object of the requested product (with SMS as an examp
         * The second parameter is the information on the region you select in Tencent
        client, _ := sms.NewClient(credential, "ap-singapore", cpf)

        /* Instantiate a request object. You can further set the request parameters acc
         * You can directly check the SDK source code to determine which attributes of
         * An attribute may be of a basic type or import another data structure
         * We recommend you use the IDE for development where you can easily redirect t
        request := sms.NewSendSmsRequest()

        /* Settings of a basic parameter:
         * The SDK uses the pointer style to specify parameters, so even for basic para
         * The SDK provides encapsulation functions for importing the pointers of basic
         * Help link:
         * SMS console: https://console.tencentcloud.com/smsv2
         * sms helper: https://cloud.tencent.com/document/product/382/3773 */

        /* SMS application ID, which is the `SdkAppId` generated after an application i
        request.SmsSdkAppId = common.StringPtr("2400006666")
        /* SMS signature content, which should be encoded in UTF-8. You must enter an a
        request.SignName = common.StringPtr("xxx")
        /* `SenderId` for Global SMS, which is not activated by default. If you need to
        request.SenderId = common.StringPtr("")
        /* User session content, which can carry context information such as user-side
        request.SessionContext = common.StringPtr("xxx")
        /* SMS code number extension, which is not activated by default. If you need to
        request.ExtendCode = common.StringPtr("")
        /* 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 vie
        request.TemplateId = common.StringPtr("449739")
        /* Target mobile number in the E.164 standard (+[country/region code][mobile nu
         * Example: +60198890000, which has a + sign followed by 60 (country/region cod
        request.PhoneNumberSet = common.StringPtrs([]string{"+60198890000"})

        // Call the API you want to access through the client object. You need to pass
        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 trouble
    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"
    "os"

    "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/v20210111" //
)

func main() {
    // Required steps:
    // Instantiate an authentication object. The input parameters need Tencent Clou
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* Optional steps:
     * Instantiate a client configuration object. You can specify the timeout perio
    cpf := profile.NewClientProfile()

    /* The SDK uses the POST method by default
     * If you have to use the GET method, you can set it here, but the GET method c
    cpf.HttpProfile.ReqMethod = "POST"

    /* The SDK has a default timeout period. Do not adjust it unless absolutely nec
     * 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 t
     * you need to manually specify the domain name. For example, the SMS domain na
    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 the client object of the requested product (with SMS as an examp
     * The second parameter is the information on the region you select in Tencent
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* Instantiate a request object. You can further set the request parameters acc
     * You can directly check the SDK source code to determine which attributes of
     * An attribute may be of a basic type or import another data structure
     * We recommend you use the IDE for development where you can easily redirect t
    request := sms.NewPullSmsSendStatusRequest()

    /* Settings of a basic parameter:
     * The SDK uses the pointer style to specify parameters, so even for basic para
     * The SDK provides encapsulation functions for importing the pointers of basic
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper: https://cloud.tencent.com/document/product/382/3773 */

    /* SMS application ID, which is the `SdkAppId` generated after an application i
    request.SmsSdkAppId = common.StringPtr("2400006666")
    /* 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
    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 trouble
    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"
    "os"

    "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/v20210111" //
)

func main() {
    // Required steps:
    // Instantiate an authentication object. The input parameters need Tencent Clou
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* Optional steps:
     * Instantiate a client configuration object. You can specify the timeout perio
    cpf := profile.NewClientProfile()

    /* The SDK uses the POST method by default
     * If you have to use the GET method, you can set it here, but the GET method c
    cpf.HttpProfile.ReqMethod = "POST"

    /* The SDK has a default timeout period. Do not adjust it unless absolutely nec
     * 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 t
     * you need to manually specify the domain name. For example, the SMS domain na
    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 the client object of the requested product (with SMS as an examp
     * The second parameter is the information on the region you select in Tencent
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* Instantiate a request object. You can further set the request parameters acc
     * You can directly check the SDK source code to determine which attributes of
     * An attribute may be of a basic type or import another data structure
     * We recommend you use the IDE for development where you can easily redirect t
    request := sms.NewSendStatusStatisticsRequest()

    /* Settings of a basic parameter:
     * The SDK uses the pointer style to specify parameters, so even for basic para
     * The SDK provides encapsulation functions for importing the pointers of basic
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper: https://cloud.tencent.com/document/product/382/3773 */

    /* SMS application ID, which is the `SdkAppId` generated after an application i
    request.SmsSdkAppId = common.StringPtr("2400006666")
    /* Maximum number of pulled entries. Maximum value: 100 */
    request.Limit = common.Uint64Ptr(10)
    /* Offset. Note: this parameter is currently fixed at 0 */
    request.Offset = common.Uint64Ptr(0)
    /* Start time of pull in the format of `yyyymmddhh` accurate to the hour */
    request.BeginTime = common.StringPtr("2019122400")
    /* End time of pull in the format of `yyyymmddhh` accurate to the hour
     * Note: `EndTime` must be after `BeginTime` */
    request.EndTime = common.StringPtr("2019122523")

    // Call the API you want to access through the client object. You need to pass
    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 trouble
    if err != nil {
        panic(err)
    }
    b, _ := json.Marshal(response.Response)
    // Print the returned JSON string
    fmt.Printf("%s", b)
}
```

Applying for SMS template

```
package main

import (
    "encoding/json"
    "fmt"
    "os"

    "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/v20210111" //
)

func main() {
    // Required steps:
    // Instantiate an authentication object. The input parameters need Tencent Clou
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // For SecretId, SecretKey lookup: https://console.tencentcloud.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* Optional steps:
     * Instantiate a client configuration object. You can specify the timeout perio

    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 c
    cpf.HttpProfile.ReqMethod = "POST"

    /* The SDK has a default timeout period. Do not adjust it unless absolutely nec
     * 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 t
     * For example, the SMS domain name of the Shanghai Finance Zone is `sms.ap-sha
    cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"
    /* The SDK uses `TC3-HMAC-SHA256` to sign by default. Do not modify this field
    cpf.SignMethod = "HmacSHA1"

    /* Instantiate an SMS client object
     * The second parameter is the information on the region you select in Tencent
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* Instantiate a request object. You can further set the request parameters acc
    * You can directly check the SDK source code to determine which attributes of t
     * An attribute may be of a basic type or import another data structure
     * We recommend you use the IDE for development where you can easily redirect t
    request := sms.NewAddSmsTemplateRequest()
    /* Settings of a basic parameter:
     * The SDK uses the pointer style to specify parameters, so even for basic para
     * The SDK provides encapsulation functions for importing the pointers of basic
     * Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     * sms helper：https://cloud.tencent.com/document/product/382/3773
     */
    /* Template name */
    request.TemplateName = common.StringPtr("Tencent")
    /* Template content */
    request.TemplateContent = common.StringPtr("Your login verification code is {1}
    /* SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS */
    request.SmsType = common.Uint64Ptr(3)
    /* 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
    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 trouble
    if err != nil {
        panic(err)
    }
    b, _ := json.Marshal(response.Response)
    // Print the returned JSON string
    fmt.Printf("%s", b)
}
```

FAQs

Proxy settings

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.

Enabling DNS cache

Currently, the SDK for Go always requests the DNS server without using the cache of nscd. You can export the environment variable `GODEBUG=netdns=cgo` or specify the `-tags 'netcgo'` parameter when compiling `go build` so as to get the nscd cache.

Ignoring server certificate verification

When the SDK is used to call a public cloud service, the server certificate must be verified to identify forged servers and ensure request security. However, in some extreme cases such as testing, you may need to ignore self-signed server certificates. Below is one of the possible methods:

```
import "crypto/tls"
...
    client, _ := cvm.NewClient(credential, regions.Guangzhou, cpf)
    tr := &http.Transport{
        TLSClientConfig: &tls.Config{InsecureSkipVerify: true},
    }
    client.WithHttpTransport(tr)
...
```

Note:

Unless you know what you are doing and understand the risks involved, do not try to disable server certificate verification.

Failure of package import with import

For example, if the error `imported and not used: "os"` is reported, it means that the package `os` is not used in the code. Therefore, simply remove it.

SDK for C++

Last updated：2024-06-27 15:48:00

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:

API version required for connecting to Tencent Cloud International::

SMS API v2021-01-11 is required. For details, see the sample code.

SMS delivery 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 only manage SMS signatures and SMS body templates in the SMS console. To use the APIs, change "Individual Verification" to "Organization Verification". For details, see Identity Verification Change Guide.

Prerequisites

You have learned about the concept of Region and selected a region as needed.

You have activated the SMS service and created a signature and SMS template that have been approved. For details, see Getting Started.

You have got the `SecretId` and `SecretKey` on the Manage API Key page in the CAM console.

`SecretId` is used to identify the API caller.

`SecretKey` is a key used to encrypt the signature and help the server verify the signature. **You should keep it private and avoid disclosure**.

The endpoint of the SMS service is `sms.tencentcloudapi.com`.

Reference

For more information on the APIs and their parameters, see API Documentation.

You can download the SDK source code from Github Repository.

Installing SDK

Environmental Dependencies

See Github repository: Environmental Dependencies

SDK build from source code

See Github repository: SDK build from source 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 its own request and response structures. This document only lists the sample code of several common features as shown below:

Sending SMS messages

```
#include <tencentcloud/core/TencentCloud.h>
#include <tencentcloud/core/profile/HttpProfile.h>
#include <tencentcloud/core/profile/ClientProfile.h>
#include <tencentcloud/core/Credential.h>
#include <tencentcloud/core/NetworkProxy.h>
#include <tencentcloud/core/AsyncCallerContext.h>
#include <tencentcloud/sms/v20210111/SmsClient.h>
#include <tencentcloud/sms/v20210111/model/SendSmsRequest.h>
#include <tencentcloud/sms/v20210111/model/SendSmsResponse.h>

#include <cstdlib>
#include <iostream>
#include <string>

using namespace TencentCloud;
using namespace TencentCloud::Sms::V20210111;
using namespace TencentCloud::Sms::V20210111::Model;
using namespace std;

int main()
{
    TencentCloud::InitAPI();

    // use the sdk
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // To ensure key security, it's recommended to set the keys in environment vari
    // Go to https://console.tencentcloud.com/cam/capi to obtain the API key Secret
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                 getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option.
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true);  // Specify whether to enable the keepalive fea
    httpProfile.SetEndpoint("sms.tencentcloudapi.com");  // Specify the endpoint. I
    httpProfile.SetReqTimeout(10);  // Specify the request timeout value, in second
    httpProfile.SetConnectTimeout(10); // Specify the response timeout value, in se

    ClientProfile clientProfile = ClientProfile(httpProfile);

    SendSmsRequest req = SendSmsRequest();

    /* Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
      */
    /* SMS application ID: The SdkAppId generated after an application is added in
    req.SetSmsSdkAppId("2400006666");

    /* SMS signature content: You must enter an approved signature that is encoded
    req.SetSignName("Tencent");

    /* Template ID: You must enter the ID of an approved template.*/
    req.SetTemplateId("449739");

    /* Template parameter: The number of template parameters must be the same as th
    req.SetTemplateParamSet(std::vector<std::string>{"1234"});

    /* Target mobile number in the E.164 standard (+[country/region code][mobile nu
     * Example: +60198890000, which has a “+” sign followed by 60 (country/region c
    req.SetPhoneNumberSet(std::vector<std::string>{"+60198890000"});

    /* (Ignorable) User session content: Context information like the user ID can b
    req.SetSessionContext("");

    /* (Ignorable) SMS code number extension, which is not activated by default. If
    req.SetExtendCode("");

    /* (Ignorable) SenderId for Global SMS, which is not activated by default. If y
    req.SetSenderId("");

    /* Instantiate the client object of the product (with SMS as an example) to be
     * The second parameter is used to specify region information. You can enter a
    SmsClient sms_client = SmsClient(cred, "ap-singapore", clientProfile);

    // (Ignorable) Set a proxy.
    // NetworkProxy proxy = NetworkProxy(NetworkProxy::Type::HTTP, "localhost.proxy
    // sms_client.SetNetworkProxy(proxy);

    auto outcome = sms_client.SendSms(req);
    if (!outcome.IsSuccess())
    {
        cout << outcome.GetError().PrintAll() << endl;
        TencentCloud::ShutdownAPI();
        return -1;
    }
    SendSmsResponse rsp = outcome.GetResult();
    cout<<"RequestId="<<rsp.GetRequestId()<<endl;
    cout<<"SendSmsResponse="<<rsp.ToJsonString()<<endl;

    TencentCloud::ShutdownAPI();

    /* Below are some error codes and the corresponding solutions to them:
     * [FailedOperation.SignatureIncorrectOrUnapproved](https://www.tencentcloud.co
     * [FailedOperation.TemplateIncorrectOrUnapproved](https://www.tencentcloud.com
     * [UnauthorizedOperation.SmsSdkAppIdVerifyFail](https://www.tencentcloud.com/d
     * [UnsupportedOperation.ContainDomesticAndInternationalPhoneNumber](https://ww
     */

    return 0;
}
```

Pulling receipt status

```
#include <tencentcloud/core/TencentCloud.h>
#include <tencentcloud/core/profile/HttpProfile.h>
#include <tencentcloud/core/profile/ClientProfile.h>
#include <tencentcloud/core/Credential.h>
#include <tencentcloud/core/NetworkProxy.h>
#include <tencentcloud/core/AsyncCallerContext.h>
#include <tencentcloud/sms/v20210111/SmsClient.h>
#include <tencentcloud/sms/v20210111/model/PullSmsSendStatusRequest.h>
#include <tencentcloud/sms/v20210111/model/PullSmsSendStatusResponse.h>

#include <cstdlib>
#include <iostream>
#include <string>

using namespace TencentCloud;
using namespace TencentCloud::Sms::V20210111;
using namespace TencentCloud::Sms::V20210111::Model;
using namespace std;
```

```
int main()
{
    TencentCloud::InitAPI();

    // use the sdk
    // Instantiate an authentication object. Enter Tencent Cloud SecretId and Secre
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // Go to https://console.tencentcloud.com/cam/capi to obtain the API key Secret
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                 getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option.
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true);  // Specify whether to enable the keepalive fea
    httpProfile.SetEndpoint("sms.tencentcloudapi.com");  // Specify the endpoint. I
    httpProfile.SetReqTimeout(30);  // Specify the request timeout value, in second
    httpProfile.SetConnectTimeout(30); // Specify the response timeout value, in se

    ClientProfile clientProfile = ClientProfile(httpProfile);

    PullSmsSendStatusRequest req = PullSmsSendStatusRequest();

    /* Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     */
    /* SMS application ID, which is the SdkAppId generated after an application is
    req.SetSmsSdkAppId("2400006666");
    // Set the max number of pulled entries. Max value: 100.
    req.SetLimit(100);

    SmsClient sms_client = SmsClient(cred, "ap-singapore", clientProfile);

    // set proxy.
    // NetworkProxy proxy = NetworkProxy(NetworkProxy::Type::HTTP, "localhost.proxy
    // cvm_client.SetNetworkProxy(proxy);

    auto outcome = sms_client.PullSmsSendStatus(req);
    if (!outcome.IsSuccess())
    {
        cout << outcome.GetError().PrintAll() << endl;
        TencentCloud::ShutdownAPI();
        return -1;
    }
    PullSmsSendStatusResponse rsp = outcome.GetResult();
    cout<<"RequestId="<<rsp.GetRequestId()<<endl;
    cout<<"PullSmsSendStatusResponse="<<rsp.ToJsonString()<<endl;

    TencentCloud::ShutdownAPI();

    return 0;
}
```

Collecting SMS sending statistics

```
#include <tencentcloud/core/TencentCloud.h>
#include <tencentcloud/core/profile/HttpProfile.h>
#include <tencentcloud/core/profile/ClientProfile.h>
#include <tencentcloud/core/Credential.h>
#include <tencentcloud/core/NetworkProxy.h>
#include <tencentcloud/core/AsyncCallerContext.h>
#include <tencentcloud/sms/v20210111/SmsClient.h>
#include <tencentcloud/sms/v20210111/model/SendStatusStatisticsRequest.h>
#include <tencentcloud/sms/v20210111/model/SendStatusStatisticsResponse.h>

#include <cstdlib>
#include <iostream>
#include <string>

using namespace TencentCloud;
using namespace TencentCloud::Sms::V20210111;
using namespace TencentCloud::Sms::V20210111::Model;
using namespace std;

int main()
{
    TencentCloud::InitAPI();

    // use the sdk
    // Instantiate an authentication object. Enter Tencent Cloud SecretId and Secre
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // Go to https://console.tencentcloud.com/cam/capi to obtain the API key Secret
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                 getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option.
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true);  // Specify whether to enable the keepalive fea
    httpProfile.SetEndpoint("sms.tencentcloudapi.com");  // Specify the endpoint. I
    httpProfile.SetReqTimeout(30);  // Specify the request timeout value, in second
    httpProfile.SetConnectTimeout(30); // Specify the response timeout value, in se
    ClientProfile clientProfile = ClientProfile(httpProfile);

    SendStatusStatisticsRequest req = SendStatusStatisticsRequest();

    /* Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     */
    /* SMS application ID, which is the SdkAppId generated after an application is
    req.SetSmsSdkAppId("2400006666");
    // Upper limit, which is currently fixed at 0.
    req.SetLimit(0);
    /* Offset, which is currently fixed at 0 */
    req.SetOffset(0);
    /* Start time in the format of “yyyymmddhh” accurate down to the hour, such as
    req.SetBeginTime("2019071100");
    /* End time in the format of “yyyymmddhh” accurate down to the hour, such as 20
        * Note: “EndTime” must be after “BeginTime”.*/
    req.SetEndTime("2019071123");

    SmsClient sms_client = SmsClient(cred, "ap-singapore", clientProfile);

    // set proxy.
    // NetworkProxy proxy = NetworkProxy(NetworkProxy::Type::HTTP, "localhost.proxy
    // cvm_client.SetNetworkProxy(proxy);

    auto outcome = sms_client.SendStatusStatistics(req);
    if (!outcome.IsSuccess())
    {
        cout << outcome.GetError().PrintAll() << endl;
        TencentCloud::ShutdownAPI();
        return -1;
    }
    SendStatusStatisticsResponse rsp = outcome.GetResult();
    cout<<"RequestId="<<rsp.GetRequestId()<<endl;
    cout<<"SendStatusStatisticsResponse="<<rsp.ToJsonString()<<endl;

    TencentCloud::ShutdownAPI();

    return 0;
}
```

Applying for SMS template

```
#include <tencentcloud/core/TencentCloud.h>
#include <tencentcloud/core/profile/HttpProfile.h>
```

```
#include <tencentcloud/core/profile/ClientProfile.h>
#include <tencentcloud/core/Credential.h>
#include <tencentcloud/core/NetworkProxy.h>
#include <tencentcloud/core/AsyncCallerContext.h>
#include <tencentcloud/sms/v20210111/SmsClient.h>
#include <tencentcloud/sms/v20210111/model/AddSmsTemplateRequest.h>
#include <tencentcloud/sms/v20210111/model/AddSmsTemplateResponse.h>

#include <cstdlib>
#include <iostream>
#include <string>

using namespace TencentCloud;
using namespace TencentCloud::Sms::V20210111;
using namespace TencentCloud::Sms::V20210111::Model;
using namespace std;

int main()
{
    TencentCloud::InitAPI();

    // use the sdk
    // Instantiate an authentication object. Enter Tencent Cloud SecretId and Secre
    // To protect key security, it is suggested to set keys in environment variable
    // Hardcoding keys into the code might lead to exposure through leaked code, po
    // Go to https://console.tencentcloud.com/cam/capi to obtain the API key Secret
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                 getenv("TENCENTCLOUD_SECRET_KEY"));

    // (Optional) Instantiate an HTTP option.
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true);  // Specify whether to enable the keepalive fea
    httpProfile.SetEndpoint("sms.tencentcloudapi.com");  // Specify the endpoint. I
    httpProfile.SetReqTimeout(30);  // Specify the request timeout value, in second
    httpProfile.SetConnectTimeout(30); // Specify the response timeout value, in se

    ClientProfile clientProfile = ClientProfile(httpProfile);

    AddSmsTemplateRequest req = AddSmsTemplateRequest();

    /* Help link:
     * SMS console: https://console.tencentcloud.com/smsv2
     */
    /* Template name */
    req.SetTemplateName("Tencent");
    /* Template content */
    req.SetTemplateContent("{Your login verification code is {1}. Please enter it w
    /* SMS type. 1: Marketing SMS, 2: Notification SMS, 3: OTP SMS */
    req.SetSmsType(3);
    /* A parameter used to specify whether it is Global SMS:
    * 0: Chinese Mainland SMS
    * 1: Global SMS.*/
    req.SetInternational(0);
    /* Template remarks, such as the reason for application and its use cases.*/
    req.SetRemark("xxx");

    SmsClient sms_client = SmsClient(cred, "ap-singapore", clientProfile);

    // set proxy.
    // NetworkProxy proxy = NetworkProxy(NetworkProxy::Type::HTTP, "localhost.proxy
    // cvm_client.SetNetworkProxy(proxy);

    auto outcome = sms_client.AddSmsTemplate(req);
    if (!outcome.IsSuccess())
    {
        cout << outcome.GetError().PrintAll() << endl;
        TencentCloud::ShutdownAPI();
        return -1;
    }
    AddSmsTemplateResponse rsp = outcome.GetResult();
    cout<<"RequestId="<<rsp.GetRequestId()<<endl;
    cout<<"AddSmsTemplateResponse="<<rsp.ToJsonString()<<endl;

    TencentCloud::ShutdownAPI();

    return 0;
}
```