

Short Message Service

Best Practice

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

 Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

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

Best Practice

How to Send Bulk Best Wishes Messages

How to Send SMS Verification Codes

Best Practice

How to Send Bulk Best Wishes Messages

Last updated : 2020-05-14 17:37:43

Best wishes for birthdays, holidays, and other matters is an important way for enterprises to retain existing customers. You can use Tencent Cloud SMS to send such messages to users on special days such as holidays, member birthdays (anniversaries), and days of major weather changes for customer care.

Best wishes messages are marketing SMS and can be sent only by **verified organizational users**. For more information, please see [Differences in Rights](#).

This document uses sending a Spring Festival best wishes message to members by company A as an example to describe how to send such messages quickly.

Preparations


- [Sign up for a Tencent Cloud account](#) and verify your organizational identity.
- [Purchase](#) an SMS package.
- Prepare SMS signature owner qualification certificates.
This document takes a business license as a qualification certificate for example.
- Understand the SMS body content review standards.

Step 1. Create a signature

After an SMS signature is submitted, it will be reviewed within two hours generally. You can [configure alarm contacts](#) to receive review result notifications.

1. Log in to the [SMS Console](#).

- Select **Mainland China SMS > Signature Management** on the left sidebar and click **Create Signature**.
- Set the following parameters as needed and according to the signature review standards:

Parameter	Sample Value
Signature use	For self-use (the signature is a company name, website, product name, or something else verified under the current account)
Signature type	Company
Signature content	A Co., Ltd.
Certificate type	Business license
Certificate upload	

- Click **OK**.
Wait for signature review. The SMS signature will be available only after its status changes to **approved**.

Step 2. Create a body template

After an SMS body template is submitted, it will be reviewed within two hours generally. You can [configure alarm contacts](#) to receive review result notifications.

1. Log in to the [SMS Console](#).
2. Select **Mainland China SMS > Template Management** on the left sidebar and click **Create Body Template**.
3. Set the following parameters as needed and according to the body template review standards:

Parameter	Sample Value
Template name	Best wishes message
SMS type	Marketing SMS
SMS content	Dear Ms./Mr. {1}, thank you for your always support and trust. We want to extend our Spring Festival greetings and best wishes to you and your beloved ones.

4. Click **OK**.

Wait for body template review. The body template will be available only after its status changes to **approved**.

Step 3. Send SMS

Before sending an SMS, you need to confirm that both the SMS signature and body template have been approved.

You can send an SMS through the console or [API](#). This document uses the console as an example.

1. Log in to the [SMS Console](#).
2. Select **Mainland China SMS > Bulk SMS** on the left sidebar and click **Create Bulk SMS Task**.
3. Configure the following parameters as needed:

Parameter	Sample Value
Signature name	Select the signature [A Co., Ltd.] created in step 1
Template name	Select the **Best wishes message** created in step 2
Sent time	Select **Send by Schedule** and specify a reasonable time point down to the

	second, such as **2020-01-25 00:00:00** . As only a time point within one month can be specified for schedule, please configure the sending task appropriately.
Recipient	Click **Download Standard Template** , enter recipient's mobile number and custom SMS content in the form by referring to the sample table , and click **Select File** to upload it. The maximum form size supported is 30 MB.

Below is a sample table:

Recipient's Mobile Number	SMS Content Variable 1
Example: 139xxxxxxxx Instructions: please enter the mobile numbers of recipients. All the mobile numbers in one single SMS sending must be registered in Mainland China. The cells need to be in a regular format, i.e., without any specific number formats.	Example: John Smith Instructions: please enter the first custom variable content according to the body template, i.e., replacing {1} in the template.

- Click **OK**.
- Check the number of recipients, indicate your consent to the prompt about fees, and click **Send**. You can view the status of the task in the Delivery Records list. When the status is **sent**, the task has been completed.

Step 4. View SMS delivery result

You can view the SMS delivery result in the following ways:

- On the **Mainland China SMS > Bulk SMS** page, click **Details & Report Analysis** on the line of the target task to view its detailed record and report analysis.
- Select **Statistics > Mainland China SMS** and you can filter and view the statistics and relevant analysis of Mainland China SMS by application, signature, body template, and time.

How to Send SMS Verification Codes

Last updated : 2020-10-16 11:49:24

Sending verification codes through SMS is the most popular and securest way to verify user identities. Currently, SMS verification codes are widely used in various application scenarios such as user registration, password reset, login protection, identity verification, random password generation, and transaction confirmation.

This document uses developing a verification code-based login and signup service based on [SCF](#) as an example to describe how to implement the SMS verification code feature.

Preparations

- You have [signed up for a Tencent Cloud account](#) and verified your organizational identity.
- You have purchased an SMS package.
- Prepare SMS signature owner qualification certificates.

This document takes a business license as a qualification certificate for example.

- Understand the SMS body content review standards.
- Get the `SDKAppID` of the SMS application.

Relevant Materials

- [Demo source code](#)
- Other products' documentation
 - [VPC documentation](#)
 - [TencentDB for MySQL documentation](#)
 - [NAT Gateway documentation](#)
 - [SCF documentation](#)

Step 1. Configure the SMS content

After an SMS signature or body template is submitted, it will be reviewed within two hours generally. You can [configure alarm contacts](#) to receive review result notifications.

Step 1.1. Create a signature

1. Log in to the [SMS Console](#).

2. Select **Mainland China SMS > Signature Management** on the left sidebar and click **Create Signature**.
3. Set the following parameters as needed:

Parameter	Sample Value
Signature use	For self-use (the signature is a company name, website, product name, or something else verified under the current account)
Signature type	Application
Signature content	Test demo
Certificate type	Screenshot of WeChat Mini Program settings page
Certificate upload	-

4. Click **OK**.
Wait for signature review. The SMS signature will be available only after its status changes to **approved**.

Step 1.2. Create a body template

1. Log in to the [SMS Console](#).
2. Select **Mainland China SMS > Template Management** on the left sidebar and click **Create Body Template**.
3. Set the following parameters as needed:

Parameter	Sample Value
Template name	Verification code SMS
SMS type	General SMS
SMS content	Your signup verification code is {1}. Please enter it within {2} minutes. If the signup was not initiated by you, please ignore this message.

4. Click **OK**.
Wait for body template review. The body template will be available only after its status changes to **approved**. Please note down the template ID.

Step 2. Set the SMS sending frequency limit (optional)

Note :

Individual users have no permission to modify the frequency limit. To use this feature, change "Individual Identity" to "Organizational Identity".

To ensure business and channel security and minimize potential financial losses caused by malicious calls of SMS APIs, you are recommended to [set the sending frequency limit](#). In addition, you can use Tencent Cloud Captcha to maximize the protection of your business security.

This document uses the default SMS sending frequency limit policy as an example.

- For SMS messages with the same content, a maximum of one such message can be sent to the same mobile number within 30 seconds.
- A maximum of 10 messages can be sent to the same mobile number on a calendar day.

Step 3. Configure the VPC and subnet

By default, SCF is deployed in the public network and can access public network only. If you need to access Tencent Cloud resources such as TencentDB instances, you need to build a VPC to ensure data and connection security.

1. [Plan the network design](#) as needed.
2. Create a VPC. For detailed directions, please see [Creating VPC](#).

Note :

The CIDRs of the VPC and subnet cannot be modified after creation.

Parameter	Sample Value
Region	South China (Guangzhou)
Name	Demo VPC
IPv4 CIDR	10.0.0.0/16
Subnet name	Demo subnet
IPv4 CIDR	10.0.0.0/16

AZ	Guangzhou Zone 3
----	------------------

Step 4. Configure a TencentDB for MySQL instance

The region and subnet AZ of the TencentDB for MySQL instance must be the same as those of the VPC configured in [step 3](#).

1. Purchase a TencentDB for MySQL instance. For detailed directions, please see [Purchase Methods](#).

Parameter	Sample Value
Billing mode	Pay-as-you-go
Region	Guangzhou
Database version	MySQL 5.7
Architecture	High-Availability Edition
Master AZ	Guangzhou Zone 3
Slave AZ	Guangzhou Zone 4
Instance specification	4-core 8,000 MB MEM
Disk	200 GB
Network	Demo VPC and demo subnet
Instance name	Demo database
Purchase quantity	1

2. Initialize the TencentDB for MySQL instance. For detailed directions, please see [Initializing TencentDB for MySQL Instance](#).

Parameter	Sample Value
Supported character set	UTF-8
Table name case sensitivity	Yes

Custom port	3306
Root account and password	Set as needed
Confirm password	Enter the password again

- Log in to the TencentDB for MySQL instance. For detailed directions, please see [Logging in to phpMyAdmin](#).
- Create a table and fields for storing information such as user phone numbers, profile photos, and nicknames as needed. For detailed directions, please see [Creating Database and Table](#).

Step 5. Create a function

SCF currently supports development in Python, Node.js, PHP, Java, and Go. This document uses Node.js as an example.

- Create a function in the region of the VPC created in [step 3](#). For detailed directions, please see [Writing Function](#).

Parameter	Sample Value
Function name	Demo
Runtime environment	Node.js 8.9
Creation method	Template function: helloworld

- Deploy the function and set **API Gateway Trigger** as the trigger. For detailed directions, please see [Deploying Function](#).

Step 6. Enable public network access (optional)

- Functions deployed in a VPC before April 29, 2020 are isolated from the public network by default. If you want them to have access to both private network and public network, you can do so by enabling public network access.

Log in to the [SCF Console](#), select **Function Service**, click the name of the target function in the

function list to enter the function configuration page. Click **Edit**, check **Public Network Access**, and click **Save** to save the configuration.

- Functions deployed on or after April 29, 2020 have public network access enabled by default, and no additional operations are required.

Step 7. Deploy the SMS SDK

1. Run the following command to install the SDK:

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

2. Import the SMS module code into your code.
3. Configure the core logic for sending SMS messages.

```
/*
 * Feature: using SDK to send SMS messages
 * Parameter: mobile number and SMS verification code
 */
async function sendSms(phone, code) {
  const tencentcloud = require('tencentcloud-sdk-nodejs');
  const SmsClient = tencentcloud.sms.v20190711.Client;
  const Credential = tencentcloud.common.Credential;
  const ClientProfile = tencentcloud.common.ClientProfile;
  const HttpProfile = tencentcloud.common.HttpProfile;
  // `secretId` and `secretKey` of Tencent Cloud account, which should not be disclosed
  const secretId = "secretId";// Set it to your real `secretId`
  const secretKey = "secretKey";// Set it to your real `secretKey`

  let cred = new Credential(secretId, secretKey);
  let httpProfile = new HttpProfile();
  httpProfile.endpoint = "sms.tencentcloudapi.com";
  let clientProfile = new ClientProfile();
  clientProfile.httpProfile = httpProfile;
  let client = new SmsClient(cred, "ap-guangzhou", clientProfile);
  phone = "+86" + phone;// Mobile number in Mainland China

  let req = {
    PhoneNumberSet: [phone],// Mobile number to which the SMS message is sent
    TemplateID: "",// ID of the template created and recorded in step 1.2
    Sign: "",// Signature created in Step 1.1
    TemplateParamSet: [code],// Random verification code
  }
```

```
SmsSdkAppid: ""// SMS application ID
}

function smsPromise() {
  return new Promise((resolve, reject) => {
    client.SendSms(req, function(errMsg, response) {
      if (errMsg) {
        reject(errMsg)
      } else {
        if(response.SendStatusSet && response.SendStatusSet[0] && response.SendStatusSet[0].Code === 0) {
          resolve({
            errorCode: 0,
            errorMessage: response.SendStatusSet[0].Message,
            data: {
              codeStr: response.SendStatusSet[0].Code,
              requestId: response.RequestId
            }
          })
        } else {
          resolve({
            errorCode: -1003, // SMS verification code sending failed
            errorMessage: response.SendStatusSet[0].Message,
            data: {
              codeStr: response.SendStatusSet[0].Code,
              requestId: response.RequestId
            }
          })
        }
      }
    })
  })
}

let queryResult = await smsPromise()
return queryResult
}
```

Step 8. Verify the core logic of verification code sending

Verification codes have a high requirement for timeliness. You can store verification codes in the memory or TencentDB for Redis and use the mobile number as a key to store information such as sending time, verification code, number of verification attempts, and verification result. For the sake of security, you are recommended to set a limit on the number of verification attempts to prevent brute force attacks. In this document, a maximum of three attempts is used as an example.

```
/*
 * Feature: getting SMS verification code based on mobile number
 */
async function getSms(queryString) {
  const code = Math.random().toString().slice(-6); // Generate a random 6-digit verification code
  const sessionId = Math.random().toString().slice(-8); // Generate a random 8-digit verification code
  const sessionCode = {
    code: code,
    sessionId: sessionId,
    sendTime: new Date().getTime(),
    num: 0, // Number of verification attempts, which can be up to 3
    used: 1 // 1: not used; 2: used
  }
  clearCacheCode()

  cacheCode[queryString.phone] = sessionCode
}
```

Step 9. Configure the login module

The login module is mainly used for user signup or login. It stores user information such as mobile number, username, profile photo, and signup time upon the first login (i.e., signup).

```
/*
 * Feature: login
 */
async function loginSms(queryString) {
  const connection = mysql.createConnection({
    host: '', // TencentDB instance IP address
    user: '', // TencentDB instance username, such as `root`
    password: '', // TencentDB instance password
    database: '' // TencentDB database name
  });
  connection.connect();

  if(queryString.token) {
    return await verifyToken(connection, queryString)
  }

  if(!queryString.code || !queryString.sessionId) {
    return {
      errorCode: -1001,
      errorMessage: "Missing parameter"
    }
  }
}
```

```
}

let result = cacheCode[queryString.phone]
if(!result || result.used === 2 || result.num >= 3) {
  return {
    errorCode: -1100,
    errorMessage: "The verification code has expired"
  }
}
if(result.sessionId !== queryString.sessionId) {
  return {
    errorCode: -1103,
    errorMessage: "Unmatched sessionId"
  }
}

if(result.code == queryString.code) {
  cacheCode[queryString.phone].used = 2; // Update the verification code status to "used"
  const queryInfoSql = `select * from info where phone = ?`
  let queryInfoResult = await wrapPromise(connection, queryInfoSql, [queryString.phone])
  if(queryInfoResult.length === 0) { // No records are found. The user has not signed up.
    return await generateInfo(connection, queryString)
  } else {
    let infoResult = queryInfoResult[0]
    return {
      errorCode: 0,
      errorMessage: "Logged in successfully",
      data: {
        phone: infoResult.phone,
        token: getToken(infoResult.userId, infoResult),
        name: infoResult.name,
        avatar: infoResult.avatar,
        userId: infoResult.userId.toString()
      }
    }
  }
} else {
  updateCacheCode(queryString.phone, result)
  return {
    errorCode: -1102,
    errorMessage: "The verification code is incorrect. Please enter again."
  }
}
}
```

In addition, to make login easier, you can use the JSON web token standard to generate a token, which can be used to maintain the login status, so that the user can stay logged in for a short time

period with no need to enter a new SMS verification code.

```
/*  
 * Feature: using JSON web token to distribute token  
 */  
function getToken(userId, infoResult) {  
  return jwt.sign({  
    phone: infoResult.phone,  
    userId: userId,  
    name: infoResult.name,  
    avatar: infoResult.avatar  
  }, privateKey, {expiresIn: tokenExpireTime});  
}
```