

消息&短信

# 开发指南

文档版本 01  
发布日期 2025-02-13



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

---

# 目录

---

<b>1 新手必读</b> .....	<b>1</b>
<b>2 开发准备</b> .....	<b>5</b>
<b>3 代码样例</b> .....	<b>7</b>
3.1 特殊 AK/SK 认证（推荐）.....	7
3.1.1 Java.....	7
3.1.2 PHP.....	14
3.1.3 Python.....	18
3.1.4 C#.....	23
3.1.5 Node.js.....	28
3.1.6 Go.....	33
3.2 X-WSSE 认证.....	38
3.2.1 Java.....	38
3.2.2 PHP.....	48
3.2.3 Python.....	56
3.2.4 C#.....	63
3.2.5 Node.js.....	73
3.2.6 Go.....	78

# 1 新手必读

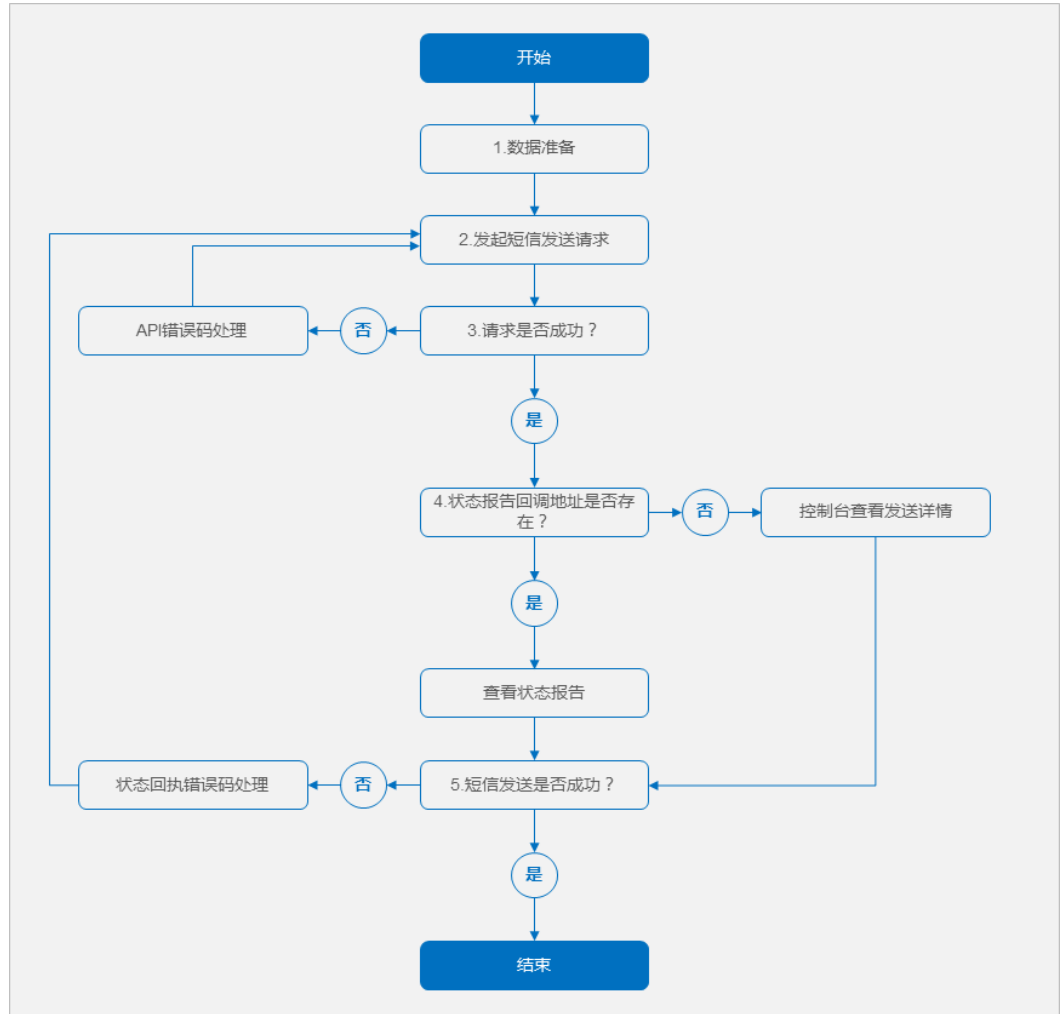
---

感谢您选用华为云消息&短信服务（Message&SMS），消息&短信是华为云携手全球多家优质运营商和渠道，为企业用户提供的优质通信服务。

本文将带您一步步地使用华为云短信服务进行二次开发。希望您能通过本文的介绍快速学习并灵活地使用短信平台进行二次代码开发和部署。

## 业务流程

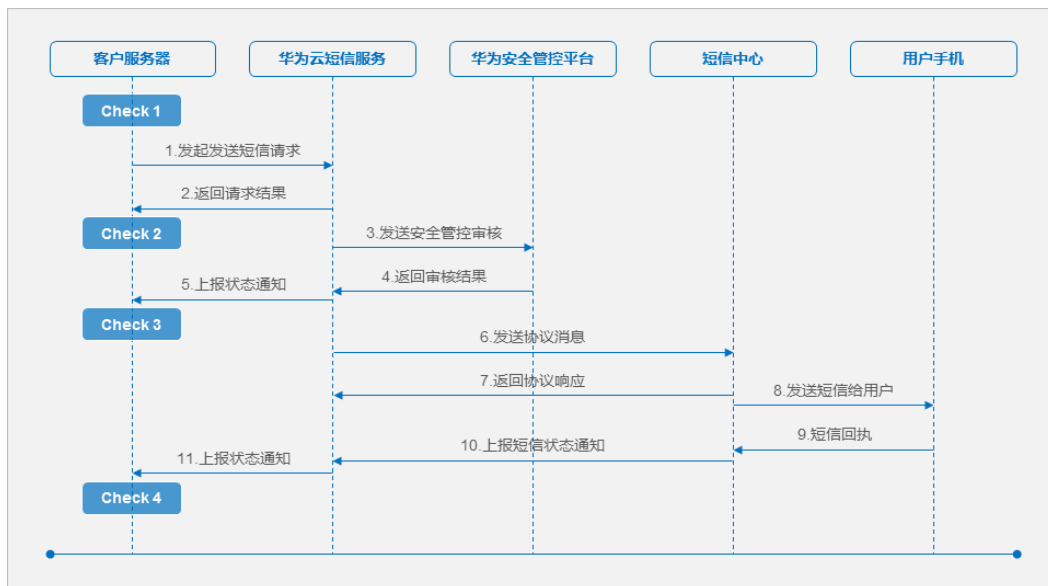
本节为您介绍新手开发者进行二次开发的业务流程。



1. 获取调用短信API的关联数据，关联数据可参考[开发准备](#)。
  2. 发起短信发送请求，参考[代码样例](#)调用短信API。
  3. 根据请求响应消息，判断请求是否成功。
    - 请求成功 => **4**
    - 请求失败 => 参考[API错误码](#)，修正后重新执行**2**
  4. 根据短信发送请求，判断请求是否通过“statusCallback”参数指定了状态报告回调地址。
    - 是 => 查看接收到的状态报告
    - 否 => 登录短信控制台，进入“发送详情”页查看发送结果、状态码
  5. 根据状态报告通知，判断短信是否发送成功。
    - 是 => 结束
    - 否 => 参考[状态回执错误码](#)，修正后重新执行**2**
- 如有任何疑问，请联系 [华为云智能客服](#)，将有专人为您解答。

## 调测指引

本节为您介绍新手开发者进行二次开发的调测点指引。



### 📖 说明

短信中心包括移动、联通、电信、广电、供应商（线路代理商）。

- **Check 1: 合法性检查。**在发起发送短信请求前，对请求参数合法性做必要的检查，如：
  - 请求Headers参数“Content-Type”在发送短信API中取值为application/x-www-form-urlencoded，在发送分批短信API中取值为application/json。
  - 请求Body参数“to”在发送短信API中取值为字符串，在发送分批短信API中取值为字符串数组。
  - 如果需要接收短信状态报告通知，则“statusCallback”参数值不能为空，且地址有效可达。
  - 如果“templateId”指定的模板类型为通用模板，则必须通过“signature”指定在通用模板短信内容前面补充的签名。
- **Check 2: 结果码解析。**获取请求结果时，请解析出响应结果码，并参考[API错误码](#)中的处理建议进行修正。

```
HTTP/1.1 200 OK
Date: Fri, 13 Apr 2018 06:29:08 GMT
Server: WebServer
Content-Type: application/json;charset=UTF-8
Content-Length: 220

{"result":[{"originTo":"+86155****5678","createTime":"2018-05-25T16:34:34Z","from":"1069-*****0012","smsMsgId":"d6e3cdd0-522b-4692-8304-a07553cdf591_8539659","status":"000000"}],"code":"000000","description":"Success"}
```

其中，“code”取值为“E000510”时，还需解析“status”参数值，进行定位分析。
- **Check 3: 解析拦截状态码。**所发短信内容触发华为平台拦截时，华为云短信服务会推送状态报告通知给客户，请解析出状态码，并参考[状态回执错误码](#)中“状态码来源”为“华为平台”的处理建议进行修正。

### 📖 说明

仅在[发送短信请求](#)中“statusCallback”参数已设置回调地址时，华为云短信服务才会推送状态报告通知。否则，请登录短信控制台，进入发送详情页查看状态码。

- **Check 4:** 解析送达状态码。华为云短信服务收到短信中心的短信状态通知时，会推送状态报告通知给客户，请解析出状态码，并参考[状态回执错误码](#)中的处理建议进行修正。

 **说明**

仅在[发送短信请求](#)中“statusCallback”参数已设置回调地址时，华为云短信服务才会推送状态报告通知。否则，请登录短信控制台，进入发送详情页查看状态码。

如有任何疑问，请联系[华为云智能客服](#)，将有专人为您解答。

# 2 开发准备

## 国内短信开发数据准备

国内短信启动开发前需要准备的数据如下：

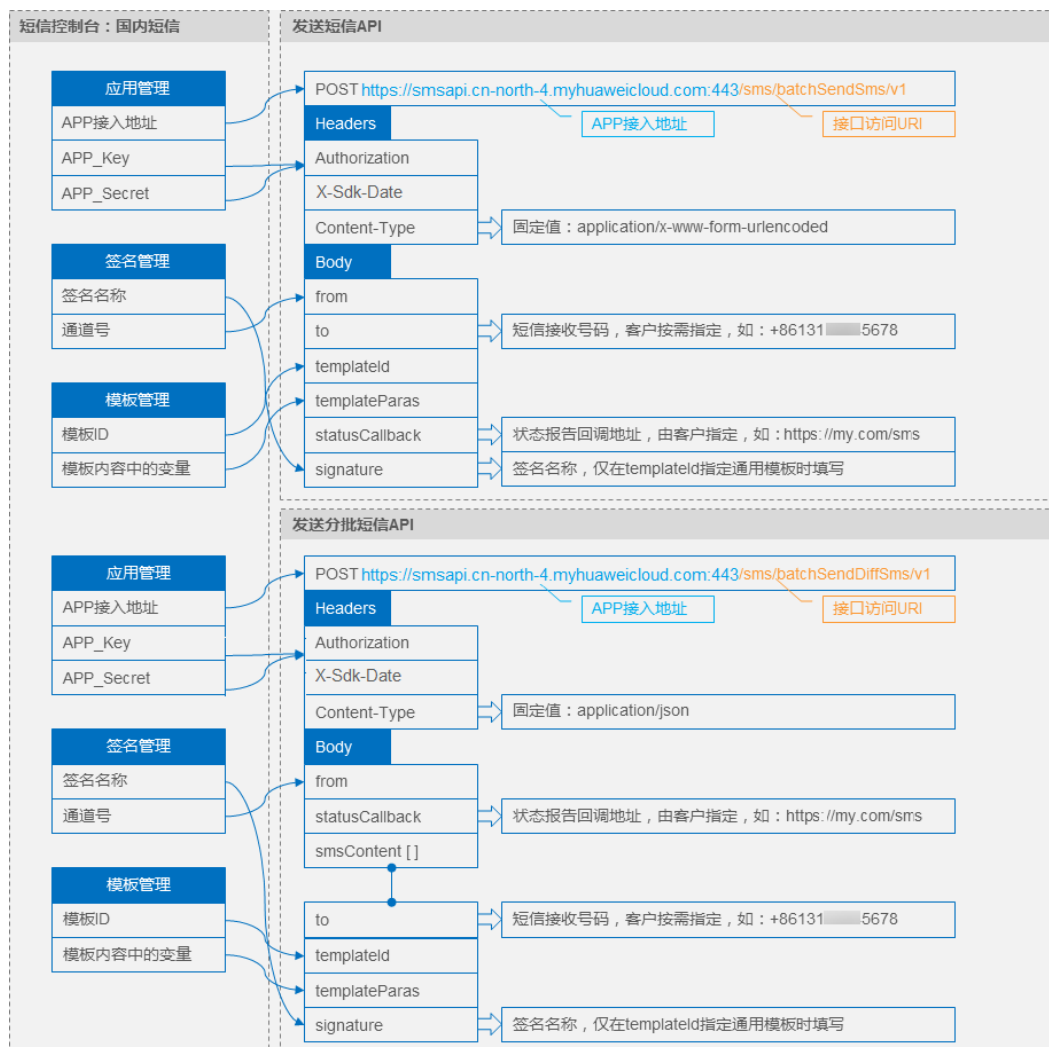
参数名	取值样例	获取方式	相关文档
APP_Key	c8RWg3gg*****4bf3Y7x1lle	登录管理控制台，从国内短信“ <a href="#">应用管理</a> ”页面获取。	<a href="#">创建短信应用</a>
APP_Secret	q4li87BhST*****zN80SfD7Al		
APP接入地址	https://example.cn-north-4.myhuaweicloud.com:443		
签名名称	华为云短信测试	登录管理控制台，从国内短信“ <a href="#">签名管理</a> ”页面获取。	<a href="#">申请短信签名</a>
签名通道号	csms12345678	<b>须知</b> 请根据“所属应用”关联获取。当模板ID指定的模板类型为通用模板时，必须准备签名名称。	
模板ID	8ff55eac1d0b47*****3c6a492300	登录管理控制台，从国内短信“ <a href="#">模板管理</a> ”页面获取。 <b>须知</b> 请根据“所属应用”和“所属签名”关联获取。发送分批短信时，可以指定多个模板ID。	<a href="#">申请短信模板</a>
访问URI	发送短信：/sms/ batchSendSms/v1 发送分批短信：/sms/ batchSendDiffSms/v1	从 <a href="#">发送短信API/发送分批短信API</a> 页面中的“接口类型”介绍部分获取。	<a href="#">发送短信</a> <a href="#">发送分批短信</a>



### 📖 说明

上表中的参数，除“访问URI”为固定值外，其他参数请根据对应的获取方式到控制台获取，取值样例仅为示例。

准备的数据与短信API请求参数关联关系如下：



如有任何疑问，请联系[华为云智能客服](#)，将有专人为您解答。

# 3 代码样例

## 3.1 特殊 AK/SK 认证（推荐）

### 3.1.1 Java

样例	<a href="#">发送短信示例</a> 、 <a href="#">发送分批短信示例</a> 、 <a href="#">接收状态报告示例</a> 、 <a href="#">接收上行短信示例</a>
环境要求	JDK1.8及以上版本。

#### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

### 发送短信示例

本示例代码为[调用API发送短信](#)，您还可以使用[发送短信SDK](#)进行短信发送。

需额外引入maven依赖，示例代码中使用了该依赖实现AK/SK签名，发送短信是调用API实现：

#### 注意

以下代码示例中的version值，请根据实际的SDK版本号进行替换。具体的SDK版本号请参见[SDK开发中心](#)。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-smsapi</artifactId>
  <version>3.1.125</version>
</dependency>

package com.huawei.smsapi;

import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.http.HttpMethod;
import com.huaweicloud.sdk.core.http.HttpRequest;
import com.huaweicloud.sdk.smsapi.utils.SmsAkSkSigner;

import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.ssl.SSLContextBuilder;
import org.apache.http.util.EntityUtils;

import javax.net.ssl.SSLContext;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Map;
import java.util.Objects;

import static java.nio.charset.StandardCharsets.UTF_8;

/**
 * Example of Invoking the batchSendSms Interface to Send an SMS Message.
 */
public class BatchSendSms {
    private static CloseableHttpClient httpClient = null;

    public static void main(String[] args) throws Exception {
        /**
         * Send an SMS message using a special AK/SK authentication algorithm.
         * When the MSGSMS is used to send SMS messages, the AK is app_key, and the SK is app_secret.
         * There will be security risks if the app_key/app_secret used for authentication is directly written into
         code.
         * We suggest encrypting the app_key/app_secret in the configuration file or environment variables for
         storage.
         * In this sample, the app_key/app_secret is stored in environment variables for identity authentication.
         * Before running this sample, set the environment variables CLOUD_SDK_MSGSMS_APPKEY and
         CLOUD_SDK_MSGSMS_APPSECRET.
         * CLOUD_SDK_MSGSMS_APPKEY indicates the application key (app_key), and
         CLOUD_SDK_MSGSMS_APPSECRET indicates the application secret (app_secret).
         * You can obtain the value from Application Management on the console or by calling the open API
         of Application Management.
         */
        String ak = System.getenv("CLOUD_SDK_MSGSMS_APPKEY");
        String sk = System.getenv("CLOUD_SDK_MSGSMS_APPSECRET");
        if (Objects.isNull(ak) || Objects.isNull(sk)) {
            System.out.println("the ak or sk is null. please config the environment
            CLOUD_SDK_MSGSMS_APPKEY and CLOUD_SDK_MSGSMS_APPSECRET first!");
            return;
        }

        // To prevent API invoking failures caused by HTTPS certificate authentication failures, ignore the
        certificate trust issue to simplify the sample code.
        httpClient = createIgnoreSSLHttpClient();
        sendSms(ak, sk);
    }
}

/**
```

```
* Example of Invoking the batchSendSms Interface to Send an SMS Message
* @param ak AK is the app_key of the SMS application.
* @param sk SK is the app_secret of the SMS application.
* @throws UnsupportedEncodingException
*/
private static void sendSms(String ak, String sk) throws UnsupportedEncodingException {
    // This address is of Beijing 4. Replace it with the actual value.
    String uri = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1";

    // Construct the message body of the sample code.
    // For details about how to construct a body, see the API description.
    StringBuilder body = new StringBuilder();
    appendToBody(body, "from=", "8824110605****");
    appendToBody(body, "&to=", "+86137****3774");
    appendToBody(body, "&templateId=", "e1440669a4354ccdb56ebf2283c6c****");
    appendToBody(body, "&templateParas=", "[\"1\", \"2\", \"3\"]");
    appendToBody(body, "&statusCallback=", "https://test/report");
    System.out.println("body:" + body);

    // Signature operation of the batchsendsms interface
    Map<String, String> signedHeaders = sign(ak, sk, body.toString(), uri);
    if (Objects.isNull(signedHeaders)) {
        System.out.println("sign failed!");
        return;
    }

    // Print Signature Results
    System.out.println("BatchSendSms sign result:" + signedHeaders);

    HttpPost postRequest = new HttpPost(uri);
    postRequest.setHeader("Accept", "application/json");
    postRequest.setHeader("Content-Type", "application/x-www-form-urlencoded");
    signedHeaders.forEach((key, value) -> postRequest.setHeader(key, value));
    postRequest.setEntity(new StringEntity(body.toString(), StandardCharsets.UTF_8));

    // Invoke /sms/batchSendSms/v1 api to send messages
    try (CloseableHttpResponse response = httpClient.execute(postRequest)) {
        // Print the response status code.
        System.out.println("Response Status Code: " + response.getStatusLine().getStatusCode());

        // Print the response content.
        System.out.println("Response Content: " + EntityUtils.toString(response.getEntity()));
    } catch (IOException e) {
        System.out.println("response exception:" + e);
    }
}

/**
 * Create an HTTP client that ignores the HTTPS certificate check.
 * @return HTTP client
 * @throws Exception
 */
private static CloseableHttpClient createIgnoreSSLHttpClient() throws Exception {
    SSLContext sslContext = new SSLContextBuilder().loadTrustMaterial(null, (x509CertChain, authType) -
> true).build();
    return HttpClients.custom().setSSLContextFactory(new SSLContextSocketFactory(sslContext,
    NoopHostnameVerifier.INSTANCE)).build();
}

/**
 * Add the parameter to the original body after URL encoding.
 * @param body Message body
 * @param key Parameter key
 * @param val Parameter value
 * @throws UnsupportedEncodingException
 */
private static void appendToBody(StringBuilder body, String key, String val) throws
UnsupportedEncodingException {
    if (null != val && !val.isEmpty()) {
```

```
        body.append(key).append(URLEncoder.encode(val, UTF_8.name()));
    }
}

/**
 * The algorithm for generating the signature header field (Authorization, X-Sdk-Date, Host) is provided.
 * @param ak AK is the app_key of the SMS application.
 * @param sk SK is the app_secret of the SMS application.
 * @param body Message body to be signed
 * @param endpoint the api uri
 * @return signature header field (Authorization, X-Sdk-Date, Host)
 */
private static Map<String, String> sign(String ak, String sk, String body, String endpoint) {
    // The signature algorithm uses the AK and SK signature algorithms provided by HUAWEI CLOUD IAM
    and API Gateway.
    // Signature algorithm implementation. The capabilities provided by the SDK are used here.
    // Developers can also use the signature capability provided by HUAWEI CLOUD APIG. For details, see
    the following website: https://support.huaweicloud.com/devg-apisign/api-sign-sdk-java.html
    // For the signature operation of an interface, the signature must contain the body.
    return SmsAkSkSigner.sign(HttpRequest.newBuilder().withBodyAsString(body)
        .withMethod(HttpMethod.POST)
        .withEndpoint(endpoint)
        .addHeader("Content-Type", "application/x-www-form-urlencoded").build(),
        new BasicCredentials().withAk(ak).withSk(sk));
}
}
```

## 发送分批短信示例

本示例代码为[调用API发送分批短信](#)，您还可以使用[发送短信SDK](#)进行短信发送。

需额外引入maven依赖，示例代码中使用了该依赖实现AK/SK签名，发送短信是调用API实现：

### 注意

以下代码示例中的version值，请根据实际的SDK版本号进行替换。具体的SDK版本号请参见[SDK开发中心](#)。

```
<dependency>
  <groupId>com.huaweicloud.sdk</groupId>
  <artifactId>huaweicloud-sdk-smsapi</artifactId>
  <version>3.1.125</version>
</dependency>
```

```
package com.huawei.smsapi;

import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.http.HttpMethod;
import com.huaweicloud.sdk.core.http.HttpRequest;
import com.huaweicloud.sdk.smsapi.util.SmsAkSkSigner;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.ssl.NoopHostnameVerifier;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.ssl.SSLContextBuilder;
import org.apache.http.util.EntityUtils;

import javax.net.ssl.SSLContext;
import java.io.IOException;
```

```
import java.io.UnsupportedEncodingException;
import java.util.Map;
import java.util.Objects;

/**
 * Example of Invoking the batchSendDiffSms Interface to Send an SMS Message.
 */
public class BatchSendDiffSms {
    private static CloseableHttpClient httpClient = null;

    public static void main(String[] args) throws Exception {
        /**
         * Send an SMS message using a special AK/SK authentication algorithm.
         * When the MSGSMS is used to send SMS messages, the AK is app_key, and the SK is app_secret.
         * There will be security risks if the app_key/app_secret used for authentication is directly written into
         code.
         * We suggest encrypting the app_key/app_secret in the configuration file or environment variables for
         storage.
         * In this sample, the app_key/app_secret is stored in environment variables for identity authentication.
         * Before running this sample, set the environment variables CLOUD_SDK_MSGSMS_APPKEY and
         CLOUD_SDK_MSGSMS_APPSECRET.
         * CLOUD_SDK_MSGSMS_APPKEY indicates the application key (app_key), and
         CLOUD_SDK_MSGSMS_APPSECRET indicates the application secret (app_secret).
         * You can obtain the value from Application Management on the console or by calling the open API
         of Application Management.
         */
        String ak = System.getenv("CLOUD_SDK_MSGSMS_APPKEY");
        String sk = System.getenv("CLOUD_SDK_MSGSMS_APPSECRET");
        if (Objects.isNull(ak) || Objects.isNull(sk)) {
            System.out.println("the ak or sk is null. please config the environment
            CLOUD_SDK_MSGSMS_APPKEY and CLOUD_SDK_MSGSMS_APPSECRET first!");
            return;
        }

        // To prevent API invoking failures caused by HTTPS certificate authentication failures, ignore the
        certificate trust issue to simplify the sample code.
        httpClient = createIgnoreSSLHttpClient();
        sendSmsUsingDiff(ak, sk);
    }

    /**
     * Example of Invoking the batchSendDiffSms Interface to Send an SMS Message
     * @param ak AK is the app_key of the SMS application.
     * @param sk SK is the app_secret of the SMS application.
     * @throws UnsupportedEncodingException
     */
    private static void sendSmsUsingDiff(String ak, String sk) throws UnsupportedEncodingException {
        // This address is of Beijing 4. Replace it with the actual value.
        String uri = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1";

        /**
         * Construct the message body of the sample code.
         * For details about how to construct a body, see the API description.
         * Example is:
         {
         "from": "8824110605***",
         "statusCallback": "https://test/report",
         "smsContent": [{
         "to": ["+86137***3774", "+86137***3775"],
         "templateId": "e1440669a4354ccdb56ebf2283c6c***",
         "templateParas": ["1", "23", "e"]
         },
         {
         "to": ["+86137***3777"],
         "templateId": "e1440669a4354ccdb56ebf2283c6c***",
         "templateParas": ["4", "5", "6"]
         }
         ]}
         */
    }
}
```

```
String body = "{" +
    "\from\": \"8824110605***\"," +
    "\statusCallback\": \"https://test/report\"," +
    "\smsContent\": {" +
    "\to\": [\"+86137***3774\", \"+86137***3775\"]," +
    "\templateId\": \"e1440669a4354ccdb56ebf2285666***\"," +
    "\templateParas\": [\"1\", \"23\", \"e\"]" +
    }," +
    {" +
    "\to\": [\"+86137***3777\"]," +
    "\templateId\": \"e1440669a4354ccdb56ebf2283c6c***\"," +
    "\templateParas\": [\"4\", \"5\", \"6\"]" +
    }" +
    "}";
System.out.println("body:" + body);

// Signature operation of the batchsenddiffsms interface
Map<String, String> signedHeaders = sign(ak, sk, body, uri);
if (Objects.isNull(signedHeaders)) {
    System.out.println("sign failed!");
    return;
}

// Print Signature Results
System.out.println("BatchSendDiffSms sign result:" + signedHeaders);

// Invoke /sms/batchSendDiffSms/v1 api to send messages
HttpPost postRequest = new HttpPost(uri);
postRequest.setHeader("Accept", "application/json");
postRequest.setHeader("Content-Type", "application/json;charset=utf8");
signedHeaders.forEach((key, value) -> postRequest.setHeader(key, value));
postRequest.setEntity(new StringEntity(body.toString(), StandardCharsets.UTF_8));

try (CloseableHttpResponse response = httpClient.execute(postRequest)) {
    // Print the response status code.
    System.out.println("Response Status Code: " + response.getStatusLine().getStatusCode());

    // Print the response content.
    System.out.println("Response Content: " + EntityUtils.toString(response.getEntity()));
} catch (IOException e) {
    System.out.println("response exception:" + e);
}
}

/**
 * Create an HTTP client that ignores the HTTPS certificate check.
 * @return HTTP client
 * @throws Exception
 */
private static CloseableHttpClient createIgnoreSSLHttpClient() throws Exception {
    SSLContext sslContext = new SSLContextBuilder().loadTrustMaterial(null, (x509CertChain, authType) -
> true).build();
    return HttpClients.custom().setSSLSocketFactory(new SSLConnectionSocketFactory(sslContext,
    NoopHostnameVerifier.INSTANCE)).build();
}

/**
 * The algorithm for generating the signature header field (Authorization, X-Sdk-Date, Host) is provided.
 * @param ak AK is the app_key of the SMS application.
 * @param sk SK is the app_secret of the SMS application.
 * @param body Message body to be signed
 * @param endpoint the api uri
 * @return signature header field (Authorization, X-Sdk-Date, Host)
 */
private static Map<String, String> sign(String ak, String sk, String body, String endpoint) {
    // The signature algorithm uses the AK and SK signature algorithms provided by HUAWEI CLOUD IAM
    and API Gateway.
    // Signature algorithm implementation. The capabilities provided by the SDK are used here.
    // Developers can also use the signature capability provided by HUAWEI CLOUD APIG. For details, see
```

```
the following website: https://support.huaweicloud.com/devg-apisign/api-sign-sdk-java.html
return SmsAkSkSigner.sign(HttpRequest.newBuilder().withBodyAsString(body)
    .withMethod(HttpMethod.POST)
    .withEndpoint(endpoint)
    .addHeader("Content-Type", "application/json;charset=utf8").build(),
    new BasicCredentials().withAk(ak).withSk(sk));
}
}
```

## 接收状态报告

需要引入的maven依赖为：org.springframework:spring-web:5.3.21（样例版本）

```
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class DemoController {
    /**
     * 同步短信回执
     */
    @PostMapping("/report")
    public void smsHwReport(@RequestParam String smsMsgId, // 发送短信成功时返回的短信唯一标识。
        @RequestParam(required = false) String total, // 长短信拆分后的短信条数。当短信未
        拆分时该参数取值为1。
        @RequestParam(required = false) String sequence, // 长短信拆分后的短信序号，当total
        参数取值大于1时，该参数才有效。当短信未拆分时该参数取值为1。
        @RequestParam String status, // 短信状态报告枚举值，常见取值请参考“API参考”
        @RequestParam(required = false) String source, // 短信状态报告来源：1：短信平台自
        行产生的状态报告。2：短信中心返回的状态报告。3：华为平台产生的状态报告。
        @RequestParam(required = false) String updateTime, // 短信资源的更新时间，通常为短
        信平台接收短信状态报告的时间，为UTC时间，格式为：yyyy-MM-dd'T'HH:mm:ss'Z'，该时间会通过urlencode
        转义为%3a。// 当短信平台未收到短信中心上报的状态报告时，会自行构造状态报告，该状态报告中不携带
        “updateTime”参数。
        @RequestParam(required = false) String orgCode, // 透传南向网元状态码，仅国际/港
        澳台短信状态报告携带，国内短信不涉及。// 当南向网元未返回状态码时不携带该参数。
        @RequestParam(required = false) String extend, // 扩展字段，由用户在发送短信的请
        求中携带。若用户发送短信时未携带extend参数，则状态报告中也不会携带extend参数。
        @RequestParam(required = false) String to) { // 本条状态报告对应的短信的接收方号
        码，仅当状态报告中携带了extend参数时才会同时携带该参数。
        System.out.println("=====receive smsStatusReport =====");
        System.out.println("smsMsgId: " + smsMsgId);
        System.out.println("total: " + total);
        System.out.println("sequence: " + sequence);
        System.out.println("status: " + status);
        System.out.println("source: " + source);
        System.out.println("updateTime: " + updateTime);
        System.out.println("orgCode: " + orgCode);
        System.out.println("extend: " + extend);
        System.out.println("to: " + to);
    }
}
```

## 接收上行短信

需要引入的maven依赖为：org.springframework:spring-web:5.3.21（样例版本）

```
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.PostMapping;

@RestController
public class DemoController {
    /**
```



```

* @param smsMsgId 上行短信的唯一标识。
* @param from 上行短信发送方的号码。
* @param to 上行短信接收方的号码。
* @param body 上行短信发送的内容。
* @param createTime // 上行短信创建时间，即短信平台接收到用户发送的上行短信的时间（采用标准
UTC格式，例如：2018-02-12T15:30:20Z。不同编程语言中的时间格式转换方式不同，部分语言可参考表 不同编
程语言的时间格式）。
* // 在控制台创建应用时配置需要“接收上行短信回复时间”后，平台才会在Body中附带此字
段，该字段只对http协议的短信生效。
*/
@PostMapping("/upSMS")
public void receiveHwSms(@RequestParam String smsMsgId,
    @RequestParam String from,
    @RequestParam String to,
    @RequestParam String body,
    @RequestParam(required = false) String createTime) {
    System.out.println(" =====receive upSMS =====");
    // 打印上面的RequestParam
    System.out.println(" smsMsgId: " + smsMsgId);
    System.out.println(" from: " + from);
    System.out.println(" to: " + to);
    System.out.println(" body: " + body);
    System.out.println(" createTime: " + createTime);
}
}

```

### 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例：<a href="#">PHP/Python/C#/3.1.5-Node.js/3.1.6-Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

### 3.1.2 PHP

样例	<a href="#">发送短信示例</a> 、 <a href="#">发送分批短信示例</a> 、 <a href="#">接受状态报告示例</a> 、 <a href="#">接收上行短信示例</a>
环境要求	<a href="#">php-7.4.33-Win32-vc15-x64</a> IntelliJ IDEA 2021.1.3安装PHP插件（Oro PHPStorm Plugin）

**须知**

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

参考[API签名SDK与demo](#)，点击下载其中的SDK与Demo。将下列代码样例替换index.php文件内容即可。

**发送短信示例**

```
<html>
<head>
  <title>Send SMS test</title>
</head>
<body>
<pre>
  <?php
    require 'signer.php';
    $signer = new Signer();
    // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境
    变量中密文存放，使用时解密，确保安全；
    $signer->Key = 'c8RWg3ggEcyd4D3p94bf3Y7x1lle'; // app key
    $signer->Secret = "q4li87Bh*****80Sfd7Al"; // app secret

    $req = new Request('POST', 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1');
    //Add header parameters
    $req->headers = array(
      'content-type' => 'application/x-www-form-urlencoded',
    );
    $req->body = 'from=csms12345678&to=
    %2B8615112346788&templateId=8ff55eac1d0b478ab3c06c3c6a492300&templateParas=["1"]&statusCallbac
    k=https://your.server.com/rest/callback/statusReport';
    $curl = $signer->Sign($req);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, FALSE);
    curl_setopt($curl, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);
    var_dump($req->headers);
    var_dump($req);
    echo "-----\n";
    $response = curl_exec($curl);
    $status = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    if ($status == 0) {
      echo curl_error($curl);
    } else {
      echo $status . "\n";
      echo $response;
    }
    curl_close($curl);
  ?>
</pre>
</body>
</html>
```

**发送分批短信示例**

```
<html>
<head>
  <title>Send Diff SMS test</title>
</head>
<body>
```

```
<pre>
<?php
require 'signer.php';

//必填,请参考“开发准备”获取如下数据,替换为实际值
$url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1'; //APP接入地址(在控制台“应用管理”页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
$APP_KEY = 'c8RWg3ggEcyd4D3p94bf3Y7x1Ile'; //APP_Key
$APP_SECRET = 'q4Ii87Bh*****80SfD7Al'; //APP_Secret
$sender = 'csms12345678'; //国内短信签名通道号
$TEMPLATE_ID_1 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID1
$TEMPLATE_ID_2 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID2

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

$signature_1 = "华为云短信测试"; //签名名称1
$signature_2 = "华为云短信测试"; //签名名称2

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
$receiver_1 = ['+86151****6789', '+86152****7890']; //模板1的接收号码
$receiver_2 = ['+86151****6789', '+86152****7890']; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
$statusCallback = "";

/**
 * 选填,使用无变量模板时请赋空值 $TEMPLATE_PARAS = [];
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,$TEMPLATE_PARAS可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,$TEMPLATE_PARAS可填写为['3','人民公园正门']
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
$TEMPLATE_PARAS_1 = ['123456']; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
$TEMPLATE_PARAS_2 = ['234567']; //模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

//请求Body
$data = json_encode([
    'from' => $sender,
    'statusCallback' => $statusCallback,
    'smsContent' => [ //必填,请根据模板ID数量灵活增减
        //smsContent,不携带签名名称时,signature请填写"
        initDiffSms($receiver_1, $TEMPLATE_ID_1, $TEMPLATE_PARAS_1, $signature_1),
        initDiffSms($receiver_2, $TEMPLATE_ID_2, $TEMPLATE_PARAS_2, $signature_2)
    ]
]);

$signer = new Signer();
$signer->Key = $APP_KEY; // app key
$signer->Secret = $APP_SECRET; // app secret

$req = new Request('POST', $url);
//Add header parameters
$req->headers = array(
    'content-type' => 'application/json',
);
$req->body = $data;
$curl = $signer->Sign($req);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, FALSE);
curl_setopt($curl, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_1_1);
var_dump($req->headers);
var_dump($req);
echo "-----\n";

```

```
$response = curl_exec($curl);
$status = curl_getinfo($curl, CURLINFO_HTTP_CODE);
if ($status == 0) {
    echo curl_error($curl);
} else {
    echo $status . "\n";
    echo $response;
}
curl_close($curl);

/**
 * 构造smsContent参数值
 * @param array $receiver
 * @param string $templateId
 * @param array $templateParas
 * @param string $signature | 签名名称,使用国内短信通用模板时填写
 * @return string[]
 */
function initDiffSms(array $receiver, string $templateId, array $templateParas, string $signature)
{
    if (null != $signature && strlen($signature) > 0) {
        return ['to' => $receiver, 'templateId' => $templateId, 'templateParas' => $templateParas, 'signature'
=> $signature];
    }
    return ['to' => $receiver, 'templateId' => $templateId, 'templateParas' => $templateParas];
}

?>
</pre>
</body>
</html>
```

## 接收状态报告

```
<?php
// 短信平台上报状态报告数据样例(urlencode)
// $success_body =
'sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD';
$failed_body =
'sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=E200027';

// onSmsStatusReport($success_body);
onSmsStatusReport($failed_body);

/**
 * 解析状态报告数据
 *
 * @param string $data 短信平台上报的状态报告数据
 */
function onSmsStatusReport(string $data)
{
    $keyValues = [];
    parse_str(urldecode($data), $keyValues); //解析状态报告数据

    /**
     * Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理
     *
     * 'smsMsgId': 短信唯一标识
     * 'total': 长短信拆分条数
     * 'sequence': 拆分后短信序号
     * 'source': 状态报告来源
     * 'updateTime': 资源更新时间
     * 'status': 状态码
     */
    $status = $keyValues['status']; // 状态报告枚举值
    // 通过status判断短信是否发送成功
    if ('DELIVRD' === strtoupper($status)) {
```

```

        print 'Send sms success. smsMsgId: ' . $keyValues['smsMsgId'] . PHP_EOL;
    } else {
        // 发送失败,打印status和orgCode
        print 'Send sms failed. smsMsgId: ' . $keyValues['smsMsgId'] . PHP_EOL;
        print 'Failed status: ' . $status . PHP_EOL;
    }
}
?>

```

## 接收上行短信

```

<?php
// 上行短信通知样例(urlencode)
$upData = 'from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-c427-4525-8e73-cf4a6ac5b3f7';

onSmsUpData($upData);

/**
 * 解析上行短信通知数据
 *
 * @param string $data 短信平台推送的上行短信通知数据
 */
function onSmsUpData(string $data)
{
    $keyValues = [];
    parse_str(urldecode($data), $keyValues); //解析上行短信通知数据

    /**
     * Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理
     *
     * 'smsMsgId': 上行短信唯一标识
     * 'from': 上行短信发送方的号码
     * 'to': 上行短信接收方的号码
     * 'body': 上行短信发送的内容
     */
    $body = $keyValues['body']; // 上行短信发送的内容
    print 'Sms up data. Body: ' . $body . PHP_EOL;
}
?>

```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例: <a href="#">Java/Python/C#/3.1.5-Node.js/3.1.6-Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

### 3.1.3 Python

样例	<a href="#">发送短信示例</a> 、 <a href="#">发送分批短信示例</a> 、 <a href="#">接收状态报告示例</a> 、
----	--

## 环境要求

基于Python 3.7.0版本，要求Python 3.7及以上版本。

## 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

参考[API签名SDK与demo](#)，点击下载其中的SDK与Demo。将下列代码样例替换main.py文件内容即可。

## 发送短信示例

```
# coding=utf-8
import urllib.parse
import urllib.request
import requests
from apig_sdk import signer

if __name__ == '__main__':

    # 必填,请参考"开发准备"获取如下数据,替换为实际值
    url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1' #APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
    # 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
    APP_KEY = "c8RWg3ggEcyd4D3p94bf3Y7x1lle" #APP_Key
    APP_SECRET = "q4li87Bh*****80SfD7Al" #APP_Secret
    sender = "csms12345678" #国内短信签名通道号
    TEMPLATE_ID = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID

    #条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

    signature = "华为云短信测试" #签名名称

    # 必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
    receiver = "+86151****6789,+86152****7890" #短信接收人号码

    # 选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
    statusCallBack = ""

    ""
    选填,使用无变量模板时请赋空值 TEMPLATE_PARAM = "";
    单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为['"369751"']
    双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为['"3","人民公园正门"']
    模板中的每个变量都必须赋值，且取值不能为空
    查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
    ""
    TEMPLATE_PARAM = ['"369751"'] #模板变量，此处以单变量验证码短信为例，请客户自行生成6位验证码，并定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。

    formData = urllib.parse.urlencode({
        'from': sender,
        'to': receiver,
        'templateId': TEMPLATE_ID,
        'templateParas': TEMPLATE_PARAM,
```

```
'statusCallback': statusCallBack,
# 'signature': signature #使用中国大陆短信通用模板时,必须填写签名名称
}).encode('ascii')
print(formData)

sig = signer.Signer()
sig.Key = APP_KEY
sig.Secret = APP_SECRET

r = signer.HttpRequest("POST", url)
r.headers = {"content-type": "application/x-www-form-urlencoded"}
r.body = formData

sig.Sign(r)
print(r.headers["X-Sdk-Date"])
print(r.headers["Authorization"])
resp = requests.request(r.method, r.scheme + "://" + r.host + r.uri, headers=r.headers, data=r.body,
verify=False)
print(resp.status_code, resp.reason)
print(resp.content)
```

## 发送分批短信示例

```
# coding=utf-8
import json

import requests

from apig_sdk import signer

if __name__ == '__main__':
    # 必填,请参考"开发准备"获取如下数据,替换为实际值
    url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1' # APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
    # 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
    APP_KEY = "c8RWg3ggEcyd4D3p94bf3Y7x1lle" # APP_Key
    APP_SECRET = "q4ii87Bh*****80SfD7Al" # APP_Secret
    sender = "csms12345678" # 国内短信签名通道号
    TEMPLATE_ID_1 = "8ff5eac1d0b478ab3c06c3c6a492300" # 模板ID1
    TEMPLATE_ID_2 = "8ff5eac1d0b478ab3c06c3c6a492300" # 模板ID2

    # 条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

    signature_1 = "华为云短信测试" # 签名名称1
    signature_2 = "华为云短信测试" # 签名名称2

    # 必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
    receiver_1 = ["+86151****6789", "+86152****7890"] # 模板1的接收号码
    receiver_2 = ["+86151****6789", "+86152****7890"] # 模板2的接收号码

    # 选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
    statusCallBack = ""

    """
    选填,使用无变量模板时请赋空值 TEMPLATE_PARAM = [];
    单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为["369751"]
    双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为["3","人民公园正门"]
    模板中的每个变量都必须赋值,且取值不能为空
    查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
    """
    TEMPLATE_PARAM_1 = ["123456"] # 模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
    TEMPLATE_PARAM_2 = ["234567"] # 模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
```

```

选填,使用无变量模板时请赋空值 TEMPLATE_PARAM = "";
单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为['369751']
双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为['3',"人民公园正
门"]
模板中的每个变量都必须赋值,且取值不能为空
查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
"""
TEMPLATE_PARAM = ['369751'] # 模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

# 请求Body
jsonData = json.dumps({'from': sender,
                      'statusCallback': statusCallBack,
                      'smsContent': [
                          {'to': receiver_1,
                           'templated': TEMPLATE_ID_1,
                           'templateParas': TEMPLATE_PARAM_1,
                           #
                           'signature':signature_1 #使用国内短信通用模板时,必须填写签名
                          },
                          {'to': receiver_2,
                           'templated': TEMPLATE_ID_2,
                           'templateParas': TEMPLATE_PARAM_2,
                           #
                           'signature':signature_2 #使用国内短信通用模板时,必须填写签名
                          }
                      ]}).encode('ascii')

sig = signer.Signer()
sig.Key = APP_KEY
sig.Secret = APP_SECRET

r = signer.HttpRequest("POST", url)
r.headers = {"content-type": "application/json"}
r.body = jsonData

sig.Sign(r)
print(r.headers["X-Sdk-Date"])
print(r.headers["Authorization"])
resp = requests.request(r.method, r.scheme + "://" + r.host + r.uri, headers=r.headers, data=r.body,
verify=False)
print(resp.status_code, resp.reason)
print(resp.content)

```

## 接收状态报告

```

# -*- coding: utf-8 -*-
import urllib.parse

# 短信平台上报状态报告数据样例(urlencode)
#success_body =
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";
failed_body =
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=E200027";

"""
解析状态报告数据
@param data: 短信平台上报的状态报告数据
@return:
"""
def onSmsStatusReport(data):
    keyValues = urllib.parse.parse_qs(data); #解析状态报告数据

    """
    Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理

    'smsMsgId': 短信唯一标识

```



```
'total': 长短信拆分条数
'sequence': 拆分后短信序号
'source': 状态报告来源
'updateTime': 资源更新时间
'status': 状态码
'''
status = keyValues.get('status'); #状态报告枚举值
# 通过status判断短信是否发送成功
if 'DELIVRD' == str.upper(status[0]):
    print('Send sms success. smsMsgId: ', keyValues.get('smsMsgId')[0]);
else:
    # 发送失败,打印status和orgCode
    print('Send sms failed. smsMsgId: ', keyValues.get('smsMsgId')[0]);
    print('Failed status: ', status[0]);

if __name__ == '__main__':
    # onSmsStatusReport(success_body)
    onSmsStatusReport(failed_body)
```

## 接收上行短信

```
# -*- coding: utf-8 -*-
import urllib.parse

# 上行短信通知样例(urlencode)
updata = "from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-c427-4525-8e73-cf4a6ac5b3f7";

'''
解析上行短信通知数据
@param data: 短信平台推送的上行短信通知数据
@return:
'''
def onSmsUpData(data):
    keyValues = urllib.parse.parse_qs(data); #解析上行短信通知数据

    '''
    Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理

    'smsMsgId': 上行短信唯一标识
    'from': 上行短信发送方的号码
    'to': 上行短信接收方的号码
    'body': 上行短信发送的内容
    '''

    body = keyValues.get('body'); #上行短信发送的内容
    print('Sms up data. Body: ', body[0]);

if __name__ == '__main__':
    onSmsUpData(updata)
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例: <a href="#">Java/PHP/Python/3.1.5-Node.js/3.1.6-Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

### 3.1.4 C#

样例	<a href="#">发送短信示例</a> 、 <a href="#">发送分批短信示例</a> 、 <a href="#">接收状态报告示例</a> 、 <a href="#">接收上行短信示例</a>
环境要求	.net 7.0, <a href="#">Visual Studio Code</a> 1.75.1 ( 插件C# v1.25.7, Code Runner v0.12.0 ( 非必选 ) )

#### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵守国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

参考[API签名SDK与demo](#)，点击下载其中的SDK与Demo。以Visual Studio Code为例，在本地资源管理器选择一个目录，比如D:/sms，使用终端进入。

1. 执行dotnet new console命令，产生一个工程。
2. 将Signer.cs、HttpEncoder.cs复制到D:/sms文件夹；然后在sms文件夹内新建SendSms.cs空文件，再将下列样例（比如“发送短信示例”）的内容复制到SendSms.cs中。
3. 用Visual Studio Code打开该工程，即可运行SendSms.cs。

#### 发送短信示例

```
using System;
using System.Net;
using System.IO;
using APIGATEWAY_SDK;
using System.Text;

namespace DEMO
{
    class Program
    {
        static void Main(string[] args)
        {
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls;

            //必填,请参考"开发准备"获取如下数据,替换为实际值
            string apiAddress = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1"; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
            // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
            string appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
            string appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
            string sender = "csms12345678"; //国内短信签名通道号
            string templateId = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID

            //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

            //string signature = "华为云短信测试"; //签名名称
```

```
//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
string receiver = "+86137****6781,+86137****6782"; //短信接收人号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
string statusCallBack = "";

/*
 * 选填,使用无变量模板时请赋空值 string templateParas = "";
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为["369751"]
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为["3","人民公园正门"]
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
 */
string templateParas = ["369751"]; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

Signer signer = new Signer();
signer.Key = appKey;
signer.Secret = appSecret;

HttpRequest r = new HttpRequest("POST", new Uri(apiAddress)); //APP接入地址(在控制台"应用管理"
页面获取)+接口访问URI

//请求Body
var body = new Dictionary<string, string>() {
    {"from", sender},
    {"to", receiver},
    {"templateId", templateId},
    {"templateParas", templateParas},
    {"statusCallback", statusCallBack},
    [{"signature", signature} //使用国内短信通用模板时,必须填写签名名称
};

r.body =new FormUrlEncodedContent(body).ReadAsStringAsync().Result;
r.headers.Add("Content-Type", "application/x-www-form-urlencoded");

HttpWebRequest req = signer.Sign(r);
Console.WriteLine(req.Headers.GetValues("x-sdk-date")[0]);
Console.WriteLine(string.Join(", ", req.Headers.GetValues("authorization")));
Console.WriteLine("body: " + r.body);

// 不校验证书
ServicePointManager.ServerCertificateValidationCallback = (sender, certificate, chain,
sslPolicyErrors) => true;
try
{
    var writer = new StreamWriter(req.GetRequestStream());
    writer.Write(r.body);
    writer.Flush();
    HttpWebResponse resp = (HttpWebResponse)req.GetResponse();
    var reader = new StreamReader(resp.GetResponseStream());
    Console.WriteLine(reader.ReadToEnd());
}
catch (WebException e)
{
    HttpWebResponse resp = (HttpWebResponse)e.Response;
    if (resp != null)
    {
        Console.WriteLine((int)resp.StatusCode + " " + resp.StatusDescription);
        var reader = new StreamReader(resp.GetResponseStream());
        Console.WriteLine(reader.ReadToEnd());
    }
    else
    {
        Console.WriteLine(e.Message);
    }
}
```

```
}  
}  
}  
}
```

## 发送分批短信示例

引用库：Newtonsoft.Json 11.0.2及以上版本，请参考<https://www.newtonsoft.com/json>获取。

```
using System;  
using System.Net;  
using System.IO;  
using APIGATEWAY_SDK;  
using System.Text;  
using System.Collections;  
using System.Collections.Generic;  
  
using Newtonsoft.Json;  
  
namespace DEMODiffSMS  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12 | SecurityProtocolType.Tls11 |  
SecurityProtocolType.Tls;  
  
            //必填,请参考“开发准备”获取如下数据,替换为实际值  
            string apiAddress = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/  
v1"; //APP接入地址(在控制台“应用管理”页面获取)+接口访问URI  
            // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或  
            // 者环境变量中密文存放,使用时解密,确保安全;  
            string appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key  
            string appSecret = "q4ii87Bh*****80SfD7Al"; //APP_Secret  
            string sender = "csms12345678"; //国内短信签名通道号  
            string templateId_1 = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID1  
            string templateId_2 = "8ff55eac1d0b478ab3c06c3c6a492301"; //模板ID2  
  
            //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与  
            // 模板类型一致的签名名称  
  
            string signature_1 = "华为云短信测试1"; //签名名称1  
            string signature_2 = "华为云短信测试2"; //签名名称2  
  
            //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔  
            string[] receiver_1 = { "+86151****6781", "+86151****6783" }; //模板1的接收号码  
            string[] receiver_2 = { "+86151****6782", "+86151****6784" }; //模板2的接收号码  
  
            //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告  
            string statusCallBack = "https://your.server.com/rest/callback/statusReport";  
  
            /*  
            * 选填,使用无变量模板时请赋空值 string[] templateParas = {};  
            * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为{"369751"}  
            * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为{"3","人民公园  
            正门"}  
            * 模板中的每个变量都必须赋值,且取值不能为空  
            * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知  
            */  
            string[] templateParas_1 = {"123456"}; //模板1变量,此处以单变量验证码短信为例,请客户自行生成  
            6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。  
            string[] templateParas_2 = {"234567"}; //模板2变量,此处以单变量验证码短信为例,请客户自行生成  
            6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。  
  
            ArrayList smsContent = new ArrayList  
            {
```

```
//smsContent,不携带签名名称时,signature请填null
InitDiffSms(receiver_1, templateId_1, templateParas_1, signature_1),
InitDiffSms(receiver_2, templateId_2, templateParas_2, signature_2)
};

//请求Body
var body = new Dictionary<string, object>{
    {"from", sender},
    {"statusCallback", statusCallBack},
    {"smsContent", smsContent}
};

Signer signer = new Signer();
signer.Key = appKey;
signer.Secret = appSecret;

HttpRequest r = new HttpRequest("POST", new Uri(apiAddress));
r.body = JsonConvert.SerializeObject(body);
r.headers.Add("Content-Type", "application/json");

HttpWebRequest req = signer.Sign(r);
Console.WriteLine(req.Headers.GetValues("x-sdk-date")[0]);
Console.WriteLine(string.Join(" ", req.Headers.GetValues("authorization")));
Console.WriteLine("body: " + r.body);

// 不校验证书
ServicePointManager.ServerCertificateValidationCallback = (sender, certificate, chain,
sslPolicyErrors) => true;
try
{
    var writer = new StreamWriter(req.GetRequestStream());
    writer.Write(r.body);
    writer.Flush();
    HttpWebResponse resp = (HttpWebResponse)req.GetResponse();
    var reader = new StreamReader(resp.GetResponseStream());
    Console.WriteLine(reader.ReadToEnd());
}
catch (WebException e)
{
    HttpWebResponse resp = (HttpWebResponse)e.Response;
    if (resp != null)
    {
        Console.WriteLine((int)resp.StatusCode + " " + resp.StatusDescription);
        var reader = new StreamReader(resp.GetResponseStream());
        Console.WriteLine(reader.ReadToEnd());
    }
    else
    {
        Console.WriteLine(e.Message);
    }
}
}

static Dictionary<string, object> InitDiffSms(string[] receiver, string templateId, string[] templateParas,
string signature)
{
    Dictionary<string, object> dic = new Dictionary<string, object>
    {
        {"to", receiver},
        {"templateId", templateId},
        {"templateParas", templateParas}
    };
    if (!signature.Equals(null) && signature.Length > 0)
    {
        dic.Add("signature", signature);
    }
}

return dic;
```

```
}  
}  
}
```

## 接收状态报告

```
using System;  
using System.Web;  
  
namespace msgsms_csharp_demo  
{  
    class Report  
    {  
        static void Main(string[] args)  
        {  
            //string success_body =  
            "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-  
            f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";  
            string failed_body =  
            "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-  
            f856-4376-afbf-570bd70a46ee_11840135&status=E200027";  
  
            //OnSmsStatusReport(success_body);  
            OnSmsStatusReport(failed_body);  
        }  
  
        /// <summary>  
        /// 解析状态报告数据  
        /// </summary>  
        /// <param name="data">短信平台上报的状态报告数据</param>  
        static void OnSmsStatusReport(string data)  
        {  
            var keyValues = HttpUtility.ParseQueryString(data); //解析状态报告数据  
  
            /**  
             * Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理  
             *  
             * 'smsMsgId': 短信唯一标识  
             * 'total': 长短信拆分条数  
             * 'sequence': 拆分后短信序号  
             * 'source': 状态报告来源  
             * 'updateTime': 资源更新时间  
             * 'status': 状态码  
             */  
            string status = keyValues.Get("status"); // 状态报告枚举值  
            // 通过status判断短信是否发送成功  
            if ("DELIVRD".Equals(status.ToUpper()))  
            {  
                Console.WriteLine("Send sms success. smsMsgId: " + keyValues.Get("smsMsgId"));  
            }  
            else  
            {  
                // 发送失败,打印status和orgCode  
                Console.WriteLine("Send sms failed. smsMsgId: " + keyValues.Get("smsMsgId"));  
                Console.WriteLine("Failed status: " + status);  
            }  
        }  
    }  
}
```

## 接收上行短信

```
using System;  
using System.Web;  
  
namespace msgsms_csharp_demo  
{  
    class UpData  
    {
```

```

static void Main(string[] args)
{
    string updata = "from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-
c427-4525-8e73-cf4a6ac5b3f7";

    onSmsUpData(updata);
}

/// <summary>
/// 解析上行短信通知数据
/// </summary>
/// <param name="data">短信平台推送的上行短信通知数据</param>
static void onSmsUpData(string data)
{
    var keyValues = HttpUtility.ParseQueryString(data); //解析上行短信通知数据

    /**
     * Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理
     *
     * 'smsMsgId': 上行短信唯一标识
     * 'from': 上行短信发送方的号码
     * 'to': 上行短信接收方的号码
     * 'body': 上行短信发送的内容
     */
    string body = keyValues.Get("body"); // 上行短信发送的内容
    Console.WriteLine("Sms up data. Body: " + body);
}
}
}

```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例: <a href="#">Java/PHP/Python/3.1.5-Node.js/3.1.6-Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.1.5 Node.js

样例	<a href="#">发送短信示例</a> 、 <a href="#">发送分批短信示例</a> 、 <a href="#">接受状态报告示例</a> 、 <a href="#">接收上行短信示例</a>
环境要求	IntelliJ IDEA 2021.1.3安装插件(Node.js (bundled 211.7628.21))

## 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信示例

```
var signer = require('./signer');
var https = require("https");
var url = require('url'); //引入url模块
var querystring = require('querystring'); // 引入querystring模块

//必填,请参考"开发准备"获取如下数据,替换为实际值
var realUrl = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1'; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
var appKey = 'c8RWg3ggEcyd4D3p94bf3Y7x1lle';
var appSecret = 'q4li87Bh*****80SfD7Al';
var sender = 'csms12345678'; //国内短信签名通道号
var templateId = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
var signature = "华为云短信测试"; //签名名称

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
var receiver = '+86151****6789,+86152****7890'; //短信接收人号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
var statusCallBack = "";

/**
 * 选填,使用无变量模板时请赋空值 var templateParas = "";
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为['3',"人民公园正门"]
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
 */
var templateParas = ['369751']; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

var urlObj = url.parse(realUrl); //解析realUrl字符串并返回一个 URL对象

var sig = new signer.Signer();
sig.Key = appKey;
sig.Secret = appSecret;

var r = new signer.HttpRequest("POST", realUrl);
r.headers = {"Content-Type": "application/x-www-form-urlencoded"};
r.body = buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature);

var opt = sig.Sign(r);
opt.hostname=urlObj.hostname
opt.port=urlObj.port
console.log(opt)

var req = https.request(opt, function (res) {
  console.log(res.statusCode);
  console.log("headers:", JSON.stringify(res.headers));
  res.on("data", function (chunk) {
```



```
        console.log(chunk.toString())
    })
});

req.on("error", function (err) {
    console.log(err.message)
});
req.write(r.body);
req.end();

function buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature){
    if (null !== signature && signature.length > 0) {
        return querystring.stringify({
            'from': sender,
            'to': receiver,
            'templateId': templateId,
            'templateParas': templateParas,
            'statusCallBack': statusCallBack,
            'signature': signature
        });
    }

    return querystring.stringify({
        'from': sender,
        'to': receiver,
        'templateId': templateId,
        'templateParas': templateParas,
        'statusCallBack': statusCallBack
    });
}
```

## 发送分批短信示例

```
var signer = require('./signer');
var https = require("https");
var url = require('url'); //引入url模块

//必填,请参考"开发准备"获取如下数据,替换为实际值
var realUrl = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1'; //APP接入地址
(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
var appKey = 'c8RWg3ggEcyd4D3p94bf3Y7x1Ile'; //APP_Key
var appSecret = 'q4li87Bh*****80SfD7Al'; //APP_Secret
var sender = 'csms12345678'; //国内短信签名通道号
var templateId1 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID1
var templateId2 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID2

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
var signature1 = "华为云短信测试"; //签名名称1
var signature2 = "华为云短信测试"; //签名名称2

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
var receiver1 = ['+86151****6789','+86152****7890']; //模板1的接收号码
var receiver2 = ['+86151****6789','+86152****7890']; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
var statusCallBack = "";

/**
 * 选填,使用无变量模板时请赋空值 var templateParas = [];
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为['3','人民公园正门']
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
 */
var templateParas1 = ['123456']; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并
```

定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。

var templateParas2 = ['234567']; //模板2变量，此处以单变量验证码短信为例，请客户自行生成6位验证码，并定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。

```
/**
 * 构造smsContent参数值
 *
 * @param receiver from
 * @param templateId to
 * @param templateParas template params value
 * @param signature | 签名名称,使用国内短信通用模板时填写
 * @returns
 */
function initDiffSms(receiver, templateId, templateParas, signature){
  if (null !== signature && signature.length > 0) {
    return {'to': receiver, 'templateId': templateId, 'templateParas': templateParas, 'signature': signature};
  }
  return {'to': receiver, 'templateId': templateId, 'templateParas': templateParas};
}

var body = JSON.stringify({ //请求Body
  'from': sender,
  'statusCallback': statusCallBack,
  'smsContent': [
    //smsContent,不携带签名名称时,signature请填写null
    initDiffSms(receiver1, templateId1, templateParas1, signature1),
    initDiffSms(receiver2, templateId2, templateParas2, signature2)
  ]
});

var urlObj = url.parse(realUrl); //解析realUrl字符串并返回一个 URL对象

var sig = new signer.Signer();
sig.Key = appKey;
sig.Secret = appSecret;

var r = new signer.HttpRequest("POST", realUrl);
r.headers = {"Content-Type": "application/json"};
r.body = body;

var opt = sig.Sign(r);
opt.hostname=urlObj.hostname
opt.port=urlObj.port
console.log(opt)

var req = https.request(opt, function (res) {
  console.log(res.statusCode);
  console.log('headers:', JSON.stringify(res.headers));
  res.on("data", function (chunk) {
    console.log(chunk.toString())
  })
});

req.on("error", function (err) {
  console.log(err.message)
});
req.write(r.body);
req.end();
```

## 接收状态报告

```
/*jshint esversion: 6 */

// 短信平台上报状态报告数据样例(urlencode)
//var success_body =
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";
var failed_body =
```

```
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-f856-4376-afbf-570bd70a46ee_11840135&status=E200027";

/**
 * 解析状态报告数据
 *
 * @param data 短信平台上报的状态报告数据
 * @returns
 */
function onSmsStatusReport(data) {
  var querystring = require('querystring');
  var keyValues = querystring.parse(data); // 解析状态报告数据

  /**
   * Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理
   *
   * 'smsMsgId': 短信唯一标识
   * 'total': 长短信拆分条数
   * 'sequence': 拆分后短信序号
   * 'source': 状态报告来源
   * 'updateTime': 资源更新时间
   * 'status': 状态码
   */
  var status = keyValues.status; // 状态报告枚举值
  // 通过status判断短信是否发送成功
  if ('DELIVRD' === status.toUpperCase()) {
    console.log('Send sms success. smsMsgId: ', keyValues.smsMsgId);
  } else {
    // 发送失败,打印status和orgCode
    console.log('Send sms failed. smsMsgId: ', keyValues.smsMsgId);
    console.log('Failed status: ', status);
  }
}

// onSmsStatusReport(success_body);
onSmsStatusReport(failed_body);
```

## 接收上行短信

```
/*jshint esversion: 6 */

// 上行短信通知样例(urlencode)
var updata = "from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-c427-4525-8e73-cf4a6ac5b3f7";

/**
 * 解析上行短信通知数据
 *
 * @param data 短信平台推送的上行短信通知数据
 * @returns
 */
function onSmsUpData(data) {
  var querystring = require('querystring');
  var keyValues = querystring.parse(data); // 解析上行短信通知数据

  /**
   * Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理
   *
   * 'smsMsgId': 上行短信唯一标识
   * 'from': 上行短信发送方的号码
   * 'to': 上行短信接收方的号码
   * 'body': 上行短信发送的内容
   */
  var body = keyValues.body; // 上行短信发送的内容
  console.log('Sms up data. Body: ', body);
}

onSmsUpData(updata);
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例: <a href="#">Java/PHP/Python/C#/3.1.6-Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.1.6 Go

样例	<a href="#">发送短信示例</a> 、 <a href="#">发送分批短信示例</a> 、 <a href="#">接受状态报告示例</a> 、 <a href="#">接收上行短信示例</a>
环境要求	<a href="#">go1.17.1</a> ，IntelliJ IDEA 2021.1.3安装插件（Go 211.7628.1）

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

参考[API签名SDK与demo](#)，点击下载其中的SDK与Demo。将下列代码样例替换demo.go文件内容即可。

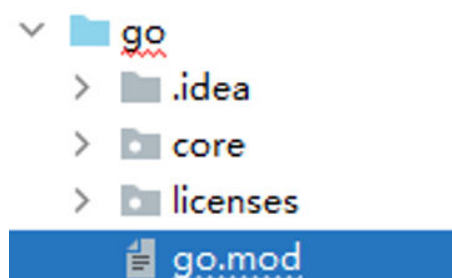
由于使用的是go1.17，请将首层go.mod文件内容修改为：

```
module huaweicloud.com/apig/go/signer

require huaweicloud.com/apig/signer v0.0.0

replace huaweicloud.com/apig/signer v0.0.0 => ./core

go 1.12
```



## 发送短信示例

```
package main

import (
    "bytes"
    "crypto/tls"
    "fmt"
    core "huaweicloud.com/apig/signer"
    "io/ioutil"
    "net/http"
    "net/url"
)

func main() {
    //必填,请参考"开发准备"获取如下数据,替换为实际值
    apInfo := core.Signer{
        // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境
        // 变量中密文存放,使用时解密,确保安全;
        Key: "c8RWg3ggEcyd4D3p94bf3Y7x1lle", //App Key
        Secret: "q4li87Bh*****80SfD7Al", //App Secret
    }
    apiAddress := "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1" //APP接入地址
    //在控制台"应用管理"页面获取+接口访问URI
    sender := "csms12345678" //国内短信签名通道号
    templateId := "8ff55eac1d0b478ab3c06c3c6a492300" //模板ID

    //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类
    //型一致的签名名称

    signature := "华为云短信测试" //签名名称

    //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
    receiver := "+86151****6789" //短信接收人号码

    //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
    statusCallBack := ""

    /*
    * 选填,使用无变量模板时请赋空值 string templateParas = "";
    * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为"[\\"369751\\"]"
    * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为"[\\"3\\"","\\"人民公园正
    门\\"]"
    * 模板中的每个变量都必须赋值,且取值不能为空
    * 查看更多 模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
    */
    templateParas := "[\\"369751\\"]" //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并
    定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

    body := buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature)
    resp, err := post(apiAddress, []byte(body), apInfo)

    if err != nil {
        return
    }
    fmt.Println(resp)
}

/**
 * sender,receiver,templateId不能为空
 */
func buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature string) string {
    param := "from=" + url.QueryEscape(sender) + "&to=" + url.QueryEscape(receiver) + "&templateId=" +
    url.QueryEscape(templateId)
    if templateParas != "" {
        param += "&templateParas=" + url.QueryEscape(templateParas)
    }
    if statusCallBack != "" {
        param += "&statusCallback=" + url.QueryEscape(statusCallBack)
    }
}
```

```
if signature != "" {
    param += "&signature=" + url.QueryEscape(signature)
}
return param
}

func post(url string, param []byte, appInfo core.Signer) (string, error) {
    if param == nil || appInfo == (core.Signer{}) {
        return "", nil
    }

    // 代码样例为了简便，设置了不进行证书校验，请在商用环境自行开启证书校验。
    tr := &http.Transport{
        TLSClientConfig: &tls.Config{InsecureSkipVerify: true},
    }
    client := &http.Client{Transport: tr}

    req, err := http.NewRequest("POST", url, bytes.NewBuffer(param))
    if err != nil {
        return "", err
    }

    // 对请求增加内容格式，固定头域
    req.Header.Add("Content-Type", "application/x-www-form-urlencoded")
    // 对请求进行HMAC算法签名，并将签名结果设置到Authorization头域。
    appInfo.Sign(req)

    fmt.Println(req.Header)
    // 发送短信请求
    resp, err := client.Do(req)
    if err != nil {
        fmt.Println(err)
    }

    // 获取短信响应
    defer resp.Body.Close()
    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        return "", err
    }
    return string(body), nil
}
```

## 发送分批短信示例

```
package main

import (
    "bytes"
    "crypto/tls"
    "encoding/json"
    "fmt"
    core "huaweicloud.com/apig/signer"
    "io/ioutil"
    "net/http"
)

func main() {
    //必填,请参考"开发准备"获取如下数据,替换为实际值
    apiAddress := "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1" //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
    // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
    appKey := "c8RWg3ggEcyd4D3p94bf3Y7x1lle" //APP_Key
    appSecret := "q4li87Bh*****80SfD7Al" //APP_Secret
    sender := "csms12345678" //国内短信签名通道号
    templateId1 := "8ff55eac1d0b478ab3c06c3c6a492300" //模板ID1
    templateId2 := "8ff55eac1d0b478ab3c06c3c6a492300" //模板ID2
}
```

```
//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

signature1 := "华为云短信测试" //签名名称1
signature2 := "华为云短信测试" //签名名称2

//必填,全局号码格式(包含国家码),示例:+8615123456789,多个号码之间用英文逗号分隔
receiver1 := []string{"+86151****6789", "+86152****7890"} //模板1的接收号码
receiver2 := []string{"+86151****6789", "+86152****7890"} //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
statusCallBack := ""

/**
 * 选填,使用无变量模板时请赋空值 templateParas := []string{}
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为[]string{"369751"}
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为[]string{"3","人民公园正门"}
 * ${DATE}${TIME}变量不允许取值为空,${TXT_20}变量可以使用英文空格或点号替代空值,${NUM_6}变量可以使用0替代空值
 * 查看更多 模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
templateParas1 := []string{"123456"} //模板1变量, 此处以单变量验证码短信为例, 请客户自行生成6位验证码, 并定义为字符串类型, 以杜绝首位0丢失的问题(例如: 002569变成了2569)。
templateParas2 := []string{"234567"} //模板2变量, 此处以单变量验证码短信为例, 请客户自行生成6位验证码, 并定义为字符串类型, 以杜绝首位0丢失的问题(例如: 002569变成了2569)。

item1 := initDiffSms(receiver1, templateId1, templateParas1, signature1)
item2 := initDiffSms(receiver2, templateId2, templateParas2, signature2)

item := []map[string]interface{}{item1, item2}
body := buildRequestBody(sender, item, statusCallBack)

appInfo := core.Signer{
    Key:   appKey, //App Key
    Secret: appSecret, //App Secret
}
resp, err := post(apiAddress, []byte(body), appInfo)
if err != nil {
    return
}
fmt.Println(resp)
}

func buildRequestBody(sender string, item []map[string]interface{}, statusCallBack string) []byte {
    body := make(map[string]interface{})
    body["smsContent"] = item
    body["from"] = sender
    if statusCallBack != "" {
        body["statusCallback"] = statusCallBack
    }
    res, _ := json.Marshal(body)
    return res
}

func initDiffSms(receiver []string, templateId string, templateParas []string, signature string) map[string]interface{} {
    diffSms := make(map[string]interface{})
    diffSms["to"] = receiver
    diffSms["templateId"] = templateId
    if templateParas != nil && len(templateParas) > 0 {
        diffSms["templateParas"] = templateParas
    }
    if signature != "" {
        diffSms["signature"] = signature
    }
    return diffSms
}
```

```
func post(url string, param []byte, applInfo core.Signer) (string, error) {
    if param == nil || applInfo == (core.Signer{}) {
        return "", nil
    }

    // 代码样例为了简便，设置了不进行证书校验，请在商用环境自行开启证书校验。
    tr := &http.Transport{
        TLSClientConfig: &tls.Config{InsecureSkipVerify: true},
    }
    client := &http.Client{Transport: tr}

    req, err := http.NewRequest("POST", url, bytes.NewBuffer(param))
    if err != nil {
        return "", err
    }

    // 对请求增加内容格式，固定头域
    req.Header.Add("Content-Type", "application/json")
    // 对请求进行HMAC算法签名，并将签名结果设置到Authorization头域。
    applInfo.Sign(req)

    fmt.Println(req.Header)
    // 发送短信请求
    resp, err := client.Do(req)
    if err != nil {
        fmt.Println(err)
    }

    // 获取短信响应
    defer resp.Body.Close()
    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        return "", err
    }
    return string(body), nil
}
```

## 接收状态报告

```
package main

import (
    "fmt"
    "net/url"
    "strings"
)

func main() {
    // 短信平台上报状态报告数据样例(urlencode)
    //success_body :=
    "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";
    failed_body :=
    "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-f856-4376-afbf-570bd70a46ee_11840135&status=E200027";
    //onSmsStatusReport(success_body);
    onSmsStatusReport(failed_body);
}

func onSmsStatusReport(data string) {
    ss, _ := url.QueryUnescape(data)
    params := strings.Split(ss, "&")
    keyValues := make(map[string]string)
    for i := range params {
        temp := strings.Split(params[i], "=")
        keyValues[temp[0]] = temp[1];
    }
    status := keyValues["status"];
    if status == "DELIVRD" {
```



```

    fmt.Println("Send sms success. smsMsgId: " + keyValues["smsMsgId"])
  } else {
    fmt.Println("Send sms failed. smsMsgId: " + keyValues["smsMsgId"])
    fmt.Println("Failed status: " + keyValues["status"])
  }
}

```

## 接收上行短信

```

package main

import (
    "fmt"
    "net/url"
    "strings"
)

func main() {
    upData := "from=%2B86151****6789&to=1069***2019&body=*****&smsMsgId=9692b5be-
c427-4525-8e73-cf4a6ac5b3f7";
    onSmsUpData(upData);
}

func onSmsUpData(data string) {
    ss, _ := url.QueryUnescape(data)
    params := strings.Split(ss, "&")
    keyValues := make(map[string]string)
    for i := range params {
        temp := strings.Split(params[i], "=")
        keyValues[temp[0]] = temp[1];
    }
    body := keyValues["body"];
    fmt.Println("Sms up data. Body: " + body)
}

```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例: <a href="#">Java/PHP/Python/C#/3.1.5-Node.js/</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.2 X-WSSE 认证

### 3.2.1 Java

样例	<a href="#">发送短信</a> <a href="#">发送分批短信</a> <a href="#">接收状态报告、接收上行短信</a>
----	---

环境要求	JDK 1.6及以上版本。
引用库	<a href="#">httpclient</a> 、 <a href="#">httpcore</a> 、 <a href="#">commons-codec</a> 、 <a href="#">commons-logging</a> 、 <a href="#">org.json</a>

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.io.Writer;
import java.net.URL;
import java.net.URLEncoder;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.text.SimpleDateFormat;
//如果JDK版本是1.8,可使用原生Base64类
import java.util.Base64;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.UUID;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

//如果JDK版本低于1.8,请使用三方库提供Base64类
//import org.apache.commons.codec.binary.Base64;

public class SendSms {

    //无需修改,用于格式化鉴权头域,给"X-WSSE"参数赋值
    private static final String WSSE_HEADER_FORMAT = "UsernameToken Username=\"%s\",PasswordDigest=
    \"%s\",Nonce=\"%s\",Created=\"%s\"";
    //无需修改,用于格式化鉴权头域,给"Authorization"参数赋值
    private static final String AUTH_HEADER_VALUE = "WSSE realm=\"SDP\",profile=\"UsernameToken
    \",type=\"Appkey\"";

    public static void main(String[] args) throws Exception {

        //必填,请参考"开发准备"获取如下数据,替换为实际值
        String url = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1"; //APP接入地址
        (在控制台"应用管理"页面获取)+接口访问URI
```

```
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境
// 变量中密文存放，使用时解密，确保安全；
String appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
String appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
String sender = "csms12345678"; //国内短信签名通道号
String templateId = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模
// 板类型一致的签名名称

String signature = "华为云短信测试"; //签名名称

//必填,全局号码格式(包含国家码),示例:+8615123456789,多个号码之间用英文逗号分隔
String receiver = "+86151****6789,+86152****7890"; //短信接收人号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
String statusCallBack = "";

/**
 * 选填,使用无变量模板时请赋空值 String templateParas = "";
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为"["369751\"]"
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为"["3","\\"人民公园
 * 正门\"]"
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
String templateParas = "["369751\"]"; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位
// 验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

//请求Body,不携带签名名称时,signature请填写null
String body = buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature);
if (null == body || body.isEmpty()) {
    System.out.println("body is null.");
    return;
}

//请求Headers中的X-WSSE参数值
String wsseHeader = buildWsseHeader(appKey, appSecret);
if (null == wsseHeader || wsseHeader.isEmpty()) {
    System.out.println("wsse header is null.");
    return;
}

Writer out = null;
BufferedReader in = null;
StringBuffer result = new StringBuffer();
URLConnection connection = null;
InputStream is = null;

HostnameVerifier hv = new HostnameVerifier() {

    @Override
    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
};
trustAllHttpsCertificates();

try {
    URL realUrl = new URL(url);
    connection = (URLConnection) realUrl.openConnection();

    connection.setHostnameVerifier(hv);
    connection.setDoOutput(true);
    connection.setDoInput(true);
    connection.setUseCaches(true);
    //请求方法
    connection.setRequestMethod("POST");
```

```
//请求Headers参数
connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
connection.setRequestProperty("Authorization", AUTH_HEADER_VALUE);
connection.setRequestProperty("X-WSSSE", wsseHeader);

connection.connect();
out = new OutputStreamWriter(connection.getOutputStream());
out.write(body); //发送请求Body参数
out.flush();
out.close();

int status = connection.getResponseCode();
if (200 == status) { //200
    is = connection.getInputStream();
} else { //400/401
    is = connection.getErrorStream();
}
in = new BufferedReader(new InputStreamReader(is, "UTF-8"));
String line = "";
while ((line = in.readLine()) != null) {
    result.append(line);
}
System.out.println(result.toString()); //打印响应消息实体
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (null != out) {
            out.close();
        }
        if (null != is) {
            is.close();
        }
        if (null != in) {
            in.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

/**
 * 构造请求Body体
 * @param sender
 * @param receiver
 * @param templateId
 * @param templateParas
 * @param statusCallBack
 * @param signature | 签名名称,使用国内短信通用模板时填写
 * @return
 */
static String buildRequestBody(String sender, String receiver, String templateId, String templateParas,
    String statusCallBack, String signature) {
    if (null == sender || null == receiver || null == templateId || sender.isEmpty() || receiver.isEmpty()
        || templateId.isEmpty()) {
        System.out.println("buildRequestBody(): sender, receiver or templateId is null.");
        return null;
    }
    Map<String, String> map = new HashMap<String, String>();

    map.put("from", sender);
    map.put("to", receiver);
    map.put("templateId", templateId);
    if (null != templateParas && !templateParas.isEmpty()) {
        map.put("templateParas", templateParas);
    }
    if (null != statusCallBack && !statusCallBack.isEmpty()) {
        map.put("statusCallback", statusCallBack);
    }
}
```

```
}
if (null != signature && !signature.isEmpty()) {
    map.put("signature", signature);
}

StringBuilder sb = new StringBuilder();
String temp = "";

for (String s : map.keySet()) {
    try {
        temp = URLEncoder.encode(map.get(s), "UTF-8");
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    sb.append(s).append("=").append(temp).append("&");
}

return sb.deleteCharAt(sb.length()-1).toString();
}

/**
 * 构造X-WSSE参数值
 * @param appKey
 * @param appSecret
 * @return
 */
static String buildWsseHeader(String appKey, String appSecret) {
    if (null == appKey || null == appSecret || appKey.isEmpty() || appSecret.isEmpty()) {
        System.out.println("buildWsseHeader(): appKey or appSecret is null.");
        return null;
    }
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'");
    String time = sdf.format(new Date()); //Created
    String nonce = UUID.randomUUID().toString().replace("-", ""); //Nonce

    MessageDigest md;
    byte[] passwordDigest = null;

    try {
        md = MessageDigest.getInstance("SHA-256");
        md.update((nonce + time + appSecret).getBytes());
        passwordDigest = md.digest();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }

    //如果JDK版本是1.8,请加载原生Base64类,并使用如下代码
    String passwordDigestBase64Str = Base64.getEncoder().encodeToString(passwordDigest); //
    PasswordDigest
    //如果JDK版本低于1.8,请加载三方库提供Base64类,并使用如下代码
    //String passwordDigestBase64Str = Base64.encodeBase64String(passwordDigest); //PasswordDigest
    //若passwordDigestBase64Str中包含换行符,请执行如下代码进行修正
    //passwordDigestBase64Str = passwordDigestBase64Str.replaceAll("[\\s*\t\n\r]", "");
    return String.format(W SSE_HEADER_FORMAT, appKey, passwordDigestBase64Str, nonce, time);
}

/** @throws Exception
 */
static void trustAllHttpsCertificates() throws Exception {
    TrustManager[] trustAllCerts = new TrustManager[] {
        new X509TrustManager() {
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
                return;
            }
            public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
                return;
            }
        }
    }
}
```

```
        public X509Certificate[] getAcceptedIssuers() {
            return null;
        }
    };
    SSLContext sc = SSLContext.getInstance("SSL");
    sc.init(null, trustAllCerts, null);
    HTTPSURLConnection.setDefaultSSLConnectionFactory(sc.getSocketFactory());
}
}
```

## 发送分批短信

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.Writer;
import java.net.URL;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
//如果JDK版本是1.8,可使用原生Base64类
import java.util.Base64;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

//如果JDK版本低于1.8,请使用三方库提供Base64类
//import org.apache.commons.codec.binary.Base64;
import org.json.JSONArray;
import org.json.JSONObject;

public class SendDiffSms {

    //无需修改,用于格式化鉴权头域,给"X-WSSE"参数赋值
    private static final String WSSE_HEADER_FORMAT = "UsernameToken Username=\"%s\",PasswordDigest=
    \"%s\",Nonce=\"%s\",Created=\"%s\"";
    //无需修改,用于格式化鉴权头域,给"Authorization"参数赋值
    private static final String AUTH_HEADER_VALUE = "WSSE realm=\"SDP\",profile=\"UsernameToken
    \",type=\"Appkey\"";

    public static void main(String[] args) throws Exception {

        //必填,请参考"开发准备"获取如下数据,替换为实际值
        String url = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1"; //APP接入
        //地址(在控制台"应用管理"页面获取)+接口访问URI
        //认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环
        //境变量中密文存放,使用时解密,确保安全;
        String appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
        String appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
        String sender = "csms12345678"; //国内短信签名通道号
        String templateId1 = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID1
        String templateId2 = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID2

        //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模
        //板类型一致的签名名称
```

```
String signature1 = "华为云短信测试"; //签名名称1
String signature2 = "华为云短信测试"; //签名名称2

//必填,全局号码格式(包含国家码),示例:+8615123456789,多个号码之间用英文逗号分隔
String[] receiver1 = {"+86151****6789", "+86152****7890"}; //模板1的接收号码
String[] receiver2 = {"+86151****6789", "+86152****7890"}; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
String statusCallBack = "";

/**
 * 选填,使用无变量模板时请赋空值 String[] templateParas = {};
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为{"369751"}
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为{"3","人民公园正
门"}
 * ${DATE}${TIME}变量不允许取值为空,${TXT_20}变量可以使用英文空格或点号替代空值,${NUM_6}变量
可以使用0替代空值
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
String[] templateParas1 = {"123456"}; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位
验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
String[] templateParas2 = {"234567"}; //模板2变量,此处以单变量验证码短信为例,请客户自行生成6位
验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

//smsContent,不携带签名名称时,signature请填写null
List<Map<String, Object>> smsContent = new ArrayList<Map<String, Object>>();
Map<String, Object> item1 = initDiffSms(receiver1, templateId1, templateParas1, signature1);
Map<String, Object> item2 = initDiffSms(receiver2, templateId2, templateParas2, signature2);
if (null != item1 && !item1.isEmpty()) {
    smsContent.add(item1);
}
if (null != item2 && !item2.isEmpty()) {
    smsContent.add(item2);
}

//请求Body
String body = buildRequestBody(sender, smsContent, statusCallBack);
if (null == body || body.isEmpty()) {
    System.out.println("body is null.");
    return;
}

//请求Headers中的X-WSSE参数值
String wsseHeader = buildWsseHeader(appKey, appSecret);
if (null == wsseHeader || wsseHeader.isEmpty()) {
    System.out.println("wsse header is null.");
}

Writer out = null;
BufferedReader in = null;
StringBuffer result = new StringBuffer();
URLConnection connection = null;
InputStream is = null;

HostnameVerifier hv = new HostnameVerifier() {

    @Override
    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
};
trustAllHttpsCertificates();

try {
    URL realUrl = new URL(url);
    connection = (URLConnection) realUrl.openConnection();

    connection.setHostnameVerifier(hv);
```

```
connection.setDoOutput(true);
connection.setDoInput(true);
connection.setUseCaches(true);
//请求方法
connection.setRequestMethod("POST");
//请求Headers参数
connection.setRequestProperty("Content-Type", "application/json");
connection.setRequestProperty("Authorization", AUTH_HEADER_VALUE);
connection.setRequestProperty("X-WSSSE", wsseHeader);

connection.connect();
out = new OutputStreamWriter(connection.getOutputStream());
out.write(body); //发送请求Body参数
out.flush();
out.close();

int status = connection.getResponseCode();
if (200 == status) { //200
    is = connection.getInputStream();
} else { //400/401
    is = connection.getErrorStream();
}
in = new BufferedReader(new InputStreamReader(is, "UTF-8"));
String line = "";
while ((line = in.readLine()) != null) {
    result.append(line);
}
System.out.println(result.toString()); //打印响应消息实体
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (null != out) {
            out.close();
        }
        if (null != is) {
            is.close();
        }
        if (null != in) {
            in.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

/**
 * 构造smsContent参数值
 * @param receiver
 * @param templateId
 * @param templateParas
 * @param signature | 签名名称,使用国内短信通用模板时填写
 * @return
 */
static Map<String, Object> initDiffSms(String[] receiver, String templateId, String[] templateParas,
String signature) {
    if (null == receiver || null == templateId || receiver.length == 0 || templateId.isEmpty()) {
        System.out.println("initDiffSms(): receiver or templateId is null.");
        return null;
    }
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("to", receiver);
    map.put("templateId", templateId);
    if (null != templateParas && templateParas.length > 0) {
        map.put("templateParas", templateParas);
    }
    if (null != signature && !signature.isEmpty()) {
        map.put("signature", signature);
    }
}
```



```
    }

    return map;
}

/**
 * 构造请求Body体
 * @param sender
 * @param smsContent
 * @param statusCallBack
 * @return
 */
static String buildRequestBody(String sender, List<Map<String, Object>> smsContent,
    String statusCallBack) {
    if (null == sender || null == smsContent || sender.isEmpty() || smsContent.isEmpty()) {
        System.out.println("buildRequestBody(): sender or smsContent is null.");
        return null;
    }
    JSONArray jsonArr = new JSONArray();

    for(Map<String, Object> it: smsContent){
        jsonArr.put(it);
    }

    Map<String, Object> data = new HashMap<String, Object>();
    data.put("from", sender);
    data.put("smsContent", jsonArr);
    if (null != statusCallBack && !statusCallBack.isEmpty()) {
        data.put("statusCallback", statusCallBack);
    }

    return new JSONObject(data).toString();
}

static String buildWsseHeader(String appKey, String appSecret) {
    if (null == appKey || null == appSecret || appKey.isEmpty() || appSecret.isEmpty()) {
        System.out.println("buildWsseHeader(): appKey or appSecret is null.");
        return null;
    }
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'");
    String time = sdf.format(new Date()); //Created
    String nonce = UUID.randomUUID().toString().replace("-", ""); //Nonce

    MessageDigest md;
    byte[] passwordDigest = null;

    try {
        md = MessageDigest.getInstance("SHA-256");
        md.update((nonce + time + appSecret).getBytes());
        passwordDigest = md.digest();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }

    //如果JDK版本是1.8,请加载原生Base64类,并使用如下代码
    String passwordDigestBase64Str = Base64.getEncoder().encodeToString(passwordDigest); //
    PasswordDigest
    //如果JDK版本低于1.8,请加载三方库提供Base64类,并使用如下代码
    //String passwordDigestBase64Str = Base64.encodeBase64String(passwordDigest); //PasswordDigest
    //若passwordDigestBase64Str中包含换行符,请执行如下代码进行修正
    //passwordDigestBase64Str = passwordDigestBase64Str.replaceAll("[\\s*\t\n\r]", "");
    return String.format(W SSE_HEADER_FORMAT, appKey, passwordDigestBase64Str, nonce, time);
}

/**
 * @throws Exception
 */
static void trustAllHttpsCertificates() throws Exception {
    TrustManager[] trustAllCerts = new TrustManager[] {
```

```
        new X509TrustManager() {
            public void checkClientTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
                return;
            }
            public void checkServerTrusted(X509Certificate[] chain, String authType) throws
CertificateException {
                return;
            }
            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        }
    };
    SSLContext sc = SSLContext.getInstance("SSL");
    sc.init(null, trustAllCerts, null);
    HTTPSURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
}
}
```

## 接收状态报告

需要引入的maven依赖为：org.springframework:spring-web:5.3.21（样例版本）

```
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class DemoController {
    /**
     * 同步短信回执
     */
    @PostMapping("/report")
    public void smsHwReport(@RequestParam String smsMsgId, // 发送短信成功时返回的短信唯一标识。
        @RequestParam(required = false) String total, // 长短信拆分后的短信条数。当短信未
拆分时该参数取值为1。
        @RequestParam(required = false) String sequence, // 长短信拆分后的短信序号，当total
参数取值大于1时，该参数才有效。当短信未拆分时该参数取值为1。
        @RequestParam String status, // 短信状态报告枚举值，常见取值请参考“API参考”
        @RequestParam(required = false) String source, // 短信状态报告来源：1：短信平台自
行产生的状态报告。2：短信中心返回的状态报告。3：华为平台产生的状态报告。
        @RequestParam(required = false) String updateTime, // 短信资源的更新时间，通常为短
信平台接收短信状态报告的时间，为UTC时间，格式为：yyyy-MM-dd'T'HH:mm:ss'Z'，该时间会通过urlencode
转义为%3a。// 当短信平台未收到短信中心上报的状态报告时，会自行构造状态报告，该状态报告中不携带
“updateTime”参数。
        @RequestParam(required = false) String orgCode, // 透传南向网元状态码，仅国际/港
澳台短信状态报告携带，国内短信不涉及。// 当南向网元未返回状态码时不携带该参数。
        @RequestParam(required = false) String extend, // 扩展字段，由用户在发送短信的请
求中携带。若用户发送短信时未携带extend参数，则状态报告中也不会携带extend参数。
        @RequestParam(required = false) String to) { // 本条状态报告对应的短信的接收方号
码，仅当状态报告中携带了extend参数时才会同时携带该参数。
        System.out.println("=====receive smsStatusReport =====");
        System.out.println("smsMsgId: " + smsMsgId);
        System.out.println("total: " + total);
        System.out.println("sequence: " + sequence);
        System.out.println("status: " + status);
        System.out.println("source: " + source);
        System.out.println("updateTime: " + updateTime);
        System.out.println("orgCode: " + orgCode);
        System.out.println("extend: " + extend);
        System.out.println("to: " + to);
    }
}
```

## 接收上行短信

需要引入的maven依赖为：org.springframework:spring-web:5.3.21（样例版本）

```
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.PostMapping;

@RestController
public class DemoController {
    /**
     * @param smsMsgId 上行短信的唯一标识。
     * @param from 上行短信发送方的号码。
     * @param to 上行短信接收方的号码。
     * @param body 上行短信发送的内容。
     * @param createTime // 上行短信创建时间，即短信平台接收到用户发送的上行短信的时间（采用标准
    UTC格式，例如：2018-02-12T15:30:20Z。不同编程语言中的时间格式转换方式不同，部分语言可参考表 不同编
    程语言的时间格式）。
     * // 在控制台创建应用时配置需要“接收上行短信回复时间”后，平台才会在Body中附带此字
    段，该字段只对http协议的短信生效。
     */
    @PostMapping("/upSMS")
    public void receiveHwSms(@RequestParam String smsMsgId,
        @RequestParam String from,
        @RequestParam String to,
        @RequestParam String body,
        @RequestParam(required = false) String createTime) {
        System.out.println(" =====receive upSMS =====");
        // 打印上面的RequestParam
        System.out.println(" smsMsgId: " + smsMsgId);
        System.out.println(" from: " + from);
        System.out.println(" to: " + to);
        System.out.println(" body: " + body);
        System.out.println(" createTime: " + createTime);
    }
}
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例：<a href="#">PHP/Python/C#/Node.js/Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.2.2 PHP

样例	<a href="#">发送短信（示例1）</a> 、 <a href="#">发送分批短信（示例1）</a> <a href="#">发送短信（示例2）</a> 、 <a href="#">发送分批短信（示例2）</a> <a href="#">接收状态报告</a> 、 <a href="#">接收上行短信</a>
----	---

环境要求	基于PHP 7.2.9版本，要求PHP 7.0及以上版本。
依赖配置	Composer、Guzzle（仅示例1依赖） 1. 请参考： <a href="https://getcomposer.org/download/">https://getcomposer.org/download/</a> ，安装配置Composer。 2. 在项目文件夹下，创建文件composer.json，内容如下： <pre>{   "require": {     "guzzlehttp/guzzle": "~6.0@dev"   } }</pre> 3. 打开命令行窗口，进入项目文件夹，执行 <i>composer install</i> ，安装guzzle。 注：安装完成，会在项目文件夹下自动生成“vendor/autoload.php”文件。

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信（示例 1）

```
<?php
require 'vendor/autoload.php'; //执行composer install时自动生成

use GuzzleHttp\Psr7;
use GuzzleHttp\Client;
use GuzzleHttp\Exception\RequestException;

//必填,请参考"开发准备"获取如下数据,替换为实际值
$url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1'; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
$APP_KEY = 'c8RWg3ggEcyd4D3p94bf3Y7x1lle'; //APP_Key
$APP_SECRET = 'q4li87Bh*****80SfD7Al'; //APP_Secret
$sender = 'csms12345678'; //国内短信签名通道号
$TEMPLATE_ID = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
$signature = "华为云短信测试"; //签名名称

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
$receiver = '+86151****6789,+86152****7890'; //短信接收人号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
$statusCallback = "";

/**
 * 选填,使用无变量模板时请赋空值 $TEMPLATE_PARAS = "";
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,$TEMPLATE_PARAS可填写为 ['369751']
 */
```

```
* 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,$TEMPLATE_PARAS可填写为["3","人民公园正
门"]
* 模板中的每个变量都必须赋值,且取值不能为空
* 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
* @var string $TEMPLATE_PARAS
*/
$TEMPLATE_PARAS = ["369751"]; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并
定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

$client = new Client();
try {
    $response = $client->request('POST', $url, [
        'form_params' => [
            'from' => $sender,
            'to' => $receiver,
            'templated' => $TEMPLATE_ID,
            'templateParas' => $TEMPLATE_PARAS,
            'statusCallback' => $statusCallback,
//            'signature' => $signature //使用国内短信通用模板时,必须填写签名名称
        ],
        'headers' => [
            'Authorization' => 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
            'X-WSSE' => buildWsseHeader($APP_KEY, $APP_SECRET)
        ],
        'verify' => false //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    ]);
    echo Psr7\str($response); //打印响应信息
} catch (RequestException $e) {
    echo $e;
    echo Psr7\str($e->getRequest()), "\n";
    if ($e->hasResponse()) {
        echo Psr7\str($e->getResponse());
    }
}

/**
 * 构造X-WSSE参数值
 * @param string $appKey
 * @param string $appSecret
 * @return string
 */
function buildWsseHeader(string $appKey, string $appSecret){
    $now = date('Y-m-d\TH:i:sZ'); //Created
    $nonce = uniqid(); //Nonce
    $base64 = base64_encode(hash('sha256', ($nonce . $now . $appSecret))); //PasswordDigest
    return sprintf("UsernameToken Username=\"%s\",PasswordDigest=\"%s\",Nonce=\"%s\",Created=\"%s\"",
        $appKey, $base64, $nonce, $now);
}
?>
```

## 发送短信 (示例 2)

```
<?php
//必填,请参考开发准备获取如下数据,替换为实际值
$url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1'; //APP接入地址(在控制台"
应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变
量中密文存放,使用时解密,确保安全;
$APP_KEY = 'c8RWg3ggEcyd4D3p94bf3Y7x11le'; //APP_Key
$APP_SECRET = 'q4li87Bh*****80SfD7Al'; //APP_Secret
$sender = 'csms12345678'; //国内短信签名通道号
$TEMPLATE_ID = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID

//条件必填,国内短信关注,当templated指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类
型一致的签名名称
$signature = '华为云短信测试'; //签名名称

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
$receiver = '+8615123456789,+86152****7890'; //短信接收人号码
```

```
//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
$statusCallback = '';

/**
 * 选填,使用无变量模板时请赋空值 $TEMPLATE_PARAS = '';
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,$TEMPLATE_PARAS可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,$TEMPLATE_PARAS可填写为['3',"人民公园正
 门']
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
 * @var string $TEMPLATE_PARAS
 */
$TEMPLATE_PARAS = ['369751']; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并
定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

//请求Headers
$headers = [
    'Content-Type: application/x-www-form-urlencoded',
    'Authorization: WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
    'X-WSSE: ' . buildWsseHeader($APP_KEY, $APP_SECRET)
];
//请求Body
$data = http_build_query([
    'from' => $sender,
    'to' => $receiver,
    'templateId' => $TEMPLATE_ID,
    'templateParas' => $TEMPLATE_PARAS,
    'statusCallback' => $statusCallback,
    // 'signature' => $signature //使用国内短信通用模板时,必须填写签名名称
]);

$context_options = [
    'http' => ['method' => 'POST', 'header' => $headers, 'content' => $data, 'ignore_errors' => true],
    'ssl' => ['verify_peer' => false, 'verify_peer_name' => false] //为防止因HTTPS证书认证失败造成API调用失
败,需要先忽略证书信任问题
];
print_r($context_options) . PHP_EOL; //打印请求信息

$response = file_get_contents($url, false, stream_context_create($context_options));
print_r($response) . PHP_EOL; //打印响应信息

/**
 * 构造X-WSSE参数值
 * @param string $appKey
 * @param string $appSecret
 * @return string
 */
function buildWsseHeader(string $appKey, string $appSecret){
    date_default_timezone_set('Asia/Shanghai');
    $now = date('Y-m-d\TH:i:sZ'); //Created
    $nonce = uniqid(); //Nonce
    $base64 = base64_encode(hash('sha256', ($nonce . $now . $appSecret))); //PasswordDigest
    return sprintf("UsernameToken Username=\"%s\",PasswordDigest=\"%s\",Nonce=\"%s\",Created=\"%s\""",
        $appKey, $base64, $nonce, $now);
}
?>
```

## 发送分批短信 ( 示例 1 )

```
<?php
require 'vendor/autoload.php'; //执行composer install时自动生成

use GuzzleHttp\Psr7;
use GuzzleHttp\Client;
use GuzzleHttp\Exception\RequestException;

//必填,请参考"开发准备"获取如下数据,替换为实际值
$url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1'; //APP接入地址(在控制
```

```
台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
$APP_KEY = 'c8RWg3ggEcyd4D3p94bf3Y7x1lle'; //APP_Key
$APP_SECRET = 'q4li87Bh*****80SfD7Al'; //APP_Secret
$sender = 'csms12345678'; //国内短信签名通道号
$TEMPLATE_ID_1 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID1
$TEMPLATE_ID_2 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID2

//条件必填,国内短信关注,当templated指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
$signature_1 = "华为云短信测试"; //签名名称1
$signature_2 = "华为云短信测试"; //签名名称2

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
$receiver_1 = ['+86151****6789', '+86152****7890']; //模板1的接收号码
$receiver_2 = ['+86151****6789', '+86152****7890']; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
$statusCallback = "";

/**
 * 选填,使用无变量模板时请赋空值 $TEMPLATE_PARAS = [];
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,$TEMPLATE_PARAS可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,$TEMPLATE_PARAS可填写为['3','人民公园正门']
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
$TEMPLATE_PARAS_1 = ['123456']; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
$TEMPLATE_PARAS_2 = ['234567']; //模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

//必填,请根据模板ID数量灵活增减
$smsContent = [
    //smsContent,不携带签名名称时,signature请填写null
    initDiffSms($receiver_1, $TEMPLATE_ID_1, $TEMPLATE_PARAS_1, $signature_1),
    initDiffSms($receiver_2, $TEMPLATE_ID_2, $TEMPLATE_PARAS_2, $signature_2)
];

$client = new Client();
try {
    $response = $client->request('POST', $url, [
        'body' => json_encode([
            'from' => $sender,
            'statusCallback' => $statusCallback,
            'smsContent' => $smsContent
        ]),
        'headers' => [
            'Authorization' => 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
            'X-WSSE' => buildWsseHeader($APP_KEY, $APP_SECRET),
            'Content-Type' => 'application/json'
        ],
        'verify' => false //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    ]);
    echo Psr7\str($response); //打印响应信息
} catch (RequestException $e) {
    echo $e;
    echo Psr7\str($e->getRequest()), "\n";
    if ($e->hasResponse()) {
        echo Psr7\str($e->getResponse());
    }
}

/**
 * 构造smsContent参数值
 * @param array $receiver
 * @param string $templated
 * @param array $templateParas
```

```
* @param string $signature | 签名名称,使用国内短信通用模板时填写
* @return string[]
*/
function initDiffSms(array $receiver, string $templatedId, array $templateParas, string $signature) {
    if (null != $signature && strlen($signature) > 0) {
        return ['to' => $receiver, 'templatedId' => $templatedId, 'templateParas' => $templateParas, 'signature'
=> $signature];
    }
    return ['to' => $receiver, 'templatedId' => $templatedId, 'templateParas' => $templateParas];
}

/**
 * 构造X-WSSE参数值
 * @param string $appKey
 * @param string $appSecret
 * @return string
 */
function buildWsseHeader(string $appKey, string $appSecret){
    $now = date('Y-m-d\TH:i:sZ'); //Created
    $nonce = uniqid(); //Nonce
    $base64 = base64_encode(hash('sha256', ($nonce . $now . $appSecret))); //PasswordDigest
    return sprintf("UsernameToken Username=\"%s\",PasswordDigest=\"%s\",Nonce=\"%s\",Created=\"%s\"",
        $appKey, $base64, $nonce, $now);
}
?>
```

## 发送分批短信（示例 2）

```
<?php
//必填,请参考"开发准备"获取如下数据,替换为实际值
$url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1'; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
$APP_KEY = 'c8RWg3ggEcyd4D3p94bf3Y7x11le'; //APP_Key
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
$APP_SECRET = 'q4li87Bh*****80SfD7Al'; //APP_Secret
$sender = 'csms12345678'; //国内短信签名通道号
$TEMPLATE_ID_1 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID1
$TEMPLATE_ID_2 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID2

//条件必填,国内短信关注,当templatedId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
$signature_1 = "华为云短信测试"; //签名名称1
$signature_2 = "华为云短信测试"; //签名名称2

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
$receiver_1 = ['+86151****6789', '+86152****7890']; //模板1的接收号码
$receiver_2 = ['+86151****6789', '+86152****7890']; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
$statusCallback = "";

/**
 * 选填,使用无变量模板时请赋空值 $TEMPLATE_PARAS = [];
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,$TEMPLATE_PARAS可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,$TEMPLATE_PARAS可填写为['3','人民公园正门']
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
$TEMPLATE_PARAS_1 = ['123456']; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
$TEMPLATE_PARAS_2 = ['234567']; //模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

//请求Headers
$headers = [
    'Content-Type: application/json',
    'Authorization: WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
    'X-WSSE: ' . buildWsseHeader($APP_KEY, $APP_SECRET)
];
```



```
];
//请求Body
$data = json_encode([
    'from' => $sender,
    'statusCallback' => $statusCallback,
    'smsContent' => [ //必填,请根据模板ID数量灵活增减
        //smsContent,不携带签名名称时,signature请填写
        initDiffSms($receiver_1, $TEMPLATE_ID_1, $TEMPLATE_PARAS_1, $signature_1),
        initDiffSms($receiver_2, $TEMPLATE_ID_2, $TEMPLATE_PARAS_2, $signature_2)
    ]
]);

$context_options = [
    'http' => ['method' => 'POST', 'header' => $headers, 'content' => $data, 'ignore_errors' => true],
    'ssl' => ['verify_peer' => false, 'verify_peer_name' => false] //为防止因HTTPS证书认证失败造成API调用失败,
    需要先忽略证书信任问题
];
print_r($context_options) . PHP_EOL; //打印请求信息

$response = file_get_contents($url, false, stream_context_create($context_options));
print_r($response) . PHP_EOL; //打印响应信息

/**
 * 构造smsContent参数值
 * @param array $receiver
 * @param string $templateId
 * @param array $templateParas
 * @param string $signature | 签名名称,使用国内短信通用模板时填写
 * @return string[]
 */
function initDiffSms(array $receiver, string $templateId, array $templateParas, string $signature) {
    if (null != $signature && strlen($signature) > 0) {
        return ['to' => $receiver, 'templateId' => $templateId, 'templateParas' => $templateParas, 'signature'
=> $signature];
    }
    return ['to' => $receiver, 'templateId' => $templateId, 'templateParas' => $templateParas];
}

/**
 * 构造X-WSSE参数值
 * @param string $appKey
 * @param string $appSecret
 * @return string
 */
function buildWsseHeader(string $appKey, string $appSecret){
    date_default_timezone_set('Asia/Shanghai');
    $now = date("Y-m-d\TH:i:sZ"); //Created
    $nonce = uniqid(); //Nonce
    $base64 = base64_encode(hash('sha256', ($nonce . $now . $appSecret))); //PasswordDigest
    return sprintf("UsernameToken Username=\"%s\",PasswordDigest=\"%s\",Nonce=\"%s\",Created=\"%s\"",
        $appKey, $base64, $nonce, $now);
}
?>
```

## 接收状态报告

```
<?php
// 短信平台上报状态报告数据样例(urlencode)
// $success_body =
'sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD';
$failed_body =
'sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=E200027';

// onSmsStatusReport($success_body);
onSmsStatusReport($failed_body);

/**
```

```
* 解析状态报告数据
*
* @param string $data 短信平台上报的状态报告数据
*/
function onSmsStatusReport(string $data)
{
    $keyValues = [];
    parse_str(urldecode($data), $keyValues); //解析状态报告数据

    /**
     * Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理
     *
     * 'smsMsgId': 短信唯一标识
     * 'total': 长短信拆分条数
     * 'sequence': 拆分后短信序号
     * 'source': 状态报告来源
     * 'updateTime': 资源更新时间
     * 'status': 状态码
     */
    $status = $keyValues['status']; // 状态报告枚举值
    // 通过status判断短信是否发送成功
    if ('DELIVRD' === strtoupper($status)) {
        print 'Send sms success. smsMsgId: ' . $keyValues['smsMsgId'] . PHP_EOL;
    } else {
        // 发送失败,打印status和orgCode
        print 'Send sms failed. smsMsgId: ' . $keyValues['smsMsgId'] . PHP_EOL;
        print 'Failed status: ' . $status . PHP_EOL;
    }
}
?>
```

## 接收上行短信

```
<?php
// 上行短信通知样例(urlencode)
$upData = 'from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-c427-4525-8e73-cf4a6ac5b3f7';

onSmsUpData($upData);

/**
 * 解析上行短信通知数据
 *
 * @param string $data 短信平台推送的上行短信通知数据
 */
function onSmsUpData(string $data)
{
    $keyValues = [];
    parse_str(urldecode($data), $keyValues); //解析上行短信通知数据

    /**
     * Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理
     *
     * 'smsMsgId': 上行短信唯一标识
     * 'from': 上行短信发送方的号码
     * 'to': 上行短信接收方的号码
     * 'body': 上行短信发送的内容
     */
    $body = $keyValues['body']; // 上行短信发送的内容
    print 'Sms up data. Body: ' . $body . PHP_EOL;
}
?>
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>• 代码样例: <a href="#">Java/Python/C#/Node.js/Go</a></li> <li>• <a href="#">发送短信API</a></li> <li>• <a href="#">发送分批短信API</a></li> <li>• <a href="#">接收状态报告API</a></li> <li>• <a href="#">接收上行短信API</a></li> <li>• <a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.2.3 Python

样例	<a href="#">发送短信（示例1）</a> 、 <a href="#">发送分批短信（示例1）</a> <a href="#">发送短信（示例2）</a> 、 <a href="#">发送分批短信（示例2）</a> <a href="#">接收状态报告</a> 、 <a href="#">接收上行短信</a>
环境要求	基于Python 3.7.0版本，要求Python 3.7及以上版本。
引用库	requests 2.18.1（仅示例1引用） 1. 请自行下载安装Python 3.7，并完成环境配置。 2. 打开命令行窗口，执行 <b>pip install requests</b> 命令。 3. 执行 <b>pip list</b> 查看安装结果。

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信（示例 1）

```
# -*- coding: utf-8 -*-
import time
import uuid
import hashlib
import base64
import requests #需要先使用pip install requests命令安装依赖

# 必填,请参考"开发准备"获取如下数据,替换为实际值
url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1' #APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全；
APP_KEY = "c8RWg3ggEcyd4D3p94bf3Y7x1lle" #APP_Key
```

```
APP_SECRET = "q4li87Bh*****80SfD7Al" #APP_Secret
sender = "csms12345678" #国内短信签名通道号
TEMPLATE_ID = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID

#条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
signature = "华为云短信测试" #签名名称

# 必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
receiver = "+86151****6789,+86152****7890" #短信接收人号码

# 选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
statusCallBack = ""

"""
选填,使用无变量模板时请赋值 TEMPLATE_PARAM = ";
单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为["369751"]
双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为["3","人民公园正门"]
模板中的每个变量都必须赋值,且取值不能为空
查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
"""
TEMPLATE_PARAM = '["369751"]' #模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

"""
构造X-WSSE参数值
@param appKey: string
@param appSecret: string
@return: string
"""
def buildWSSEHeader(appKey, appSecret):
    now = time.strftime('%Y-%m-%dT%H:%M:%SZ') #Created
    nonce = str(uuid.uuid4()).replace('-', '') #Nonce
    digest = hashlib.sha256((nonce + now + appSecret).encode()).hexdigest()

    digestBase64 = base64.b64encode(digest.encode()).decode() #PasswordDigest
    return 'UsernameToken Username="{0}",PasswordDigest="{1}",Nonce="{2}",Created="{3}"'.format(appKey,
    digestBase64, nonce, now)

def main():
    # 请求Headers
    header = {'Authorization': 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
             'X-WSSE': buildWSSEHeader(APP_KEY, APP_SECRET)}
    # 请求Body
    formData = {'from': sender,
               'to': receiver,
               'templateId': TEMPLATE_ID,
               'templateParas': TEMPLATE_PARAM,
               'statusCallback': statusCallBack,
               'signature': signature #使用国内短信通用模板时,必须填写签名名称
    }
    print(header)

    # 为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    r = requests.post(url, data=formData, headers=header, verify=False)
    print(r.text) #打印响应信息

if __name__ == '__main__':
    main()
```

## 发送短信（示例 2）

```
# -*- coding: utf-8 -*-
import time
import uuid
import hashlib
import base64
import ssl
```

```
import urllib.parse
import urllib.request

# 必填,请参考"开发准备"获取如下数据,替换为实际值
url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1' #APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
APP_KEY = "c8RWg3ggEcyd4D3p94bf3Y7x1lle" #APP_Key
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
APP_SECRET = "q4li87Bh*****80SfD7Al" #APP_Secret
sender = "csms12345678" #国内短信签名通道号
TEMPLATE_ID = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID

#条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
signature = "华为云短信测试" #签名名称

# 必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
receiver = "+86151****6789,+86152****7890" #短信接收人号码

# 选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
statusCallBack = ""

"""
选填,使用无变量模板时请赋值 TEMPLATE_PARAM = "";
单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为['369751']
双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为['3',"人民公园正门"]
模板中的每个变量都必须赋值,且取值不能为空
查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
"""
TEMPLATE_PARAM = ['369751'] #模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

"""
构造X-WSSSE参数值
@param appKey: string
@param appSecret: string
@return: string
"""
def buildWSSSEHeader(appKey, appSecret):
    now = time.strftime('%Y-%m-%dT%H:%M:%SZ') #Created
    nonce = str(uuid.uuid4()).replace('-', '') #Nonce
    digest = hashlib.sha256((nonce + now + appSecret).encode()).hexdigest()

    digestBase64 = base64.b64encode(digest.encode()).decode() #PasswordDigest
    return 'UsernameToken Username="{0}",PasswordDigest="{1}",Nonce="{2}",Created="{3}"'.format(appKey,
    digestBase64, nonce, now)

def main():
    # 请求Body
    formData = urllib.parse.urlencode({
        'from': sender,
        'to': receiver,
        'templateId': TEMPLATE_ID,
        'templateParas': TEMPLATE_PARAM,
        'statusCallback': statusCallBack,
#
        'signature': signature #使用国内短信通用模板时,必须填写签名名称
    }).encode('ascii')

    req = urllib.request.Request(url=url, data=formData, method='POST') #请求方法为POST
    # 请求Headers参数
    req.add_header('Authorization', 'WSSE realm="SDP",profile="UsernameToken",type="Appkey")
    req.add_header('X-WSSSE', buildWSSSEHeader(APP_KEY, APP_SECRET))
    req.add_header('Content-Type', 'application/x-www-form-urlencoded')

    # 为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    ssl_create_default_https_context = ssl.create_unverified_context
```

```
try:
    r = urllib.request.urlopen(req) #发送请求
    print(r.read().decode('utf-8')) #打印响应信息
except urllib.error.HTTPError as e:
    print(e.code)
    print(e.read().decode('utf-8'))
except urllib.error.URLError as e:
    print(e.reason)

if __name__ == '__main__':
    main()
```

## 发送分批短信（示例 1）

```
# -*- coding: utf-8 -*-
import time
import uuid
import hashlib
import base64
import requests #需要先使用pip install requests命令安装依赖

# 必填,请参考"开发准备"获取如下数据,替换为实际值
url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1' #APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
APP_KEY = "c8RWg3ggEcyd4D3p94bf3Y7x1lle" #APP_Key
APP_SECRET = "q4li87Bh*****80SfD7Al" #APP_Secret
sender = "csms12345678" #国内短信签名通道号
TEMPLATE_ID_1 = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID1
TEMPLATE_ID_2 = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID2

#条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
signature_1 = "华为云短信测试" #签名名称1
signature_2 = "华为云短信测试" #签名名称2

# 必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
receiver_1 = ["+86151****6789", "+86152****7890"] #模板1的接收号码
receiver_2 = ["+86151****6789", "+86152****7890"] #模板2的接收号码

# 必填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
statusCallBack = ""

"""
选填,使用无变量模板时请赋空值 TEMPLATE_PARAM = [];
单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为["369751"]
双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为["3","人民公园正门"]
模板中的每个变量都必须赋值,且取值不能为空
查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
"""
TEMPLATE_PARAM_1 = ["123456"] #模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
TEMPLATE_PARAM_2 = ["234567"] #模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

"""
构造X-WSSE参数值
@param appKey: string
@param appSecret: string
@return: string
"""
def buildWSSEHeader(appKey, appSecret):
    now = time.strftime('%Y-%m-%dT%H:%M:%SZ') #Created
    nonce = str(uuid.uuid4()).replace('-', '') #Nonce
    digest = hashlib.sha256((nonce + now + appSecret).encode()).hexdigest()

    digestBase64 = base64.b64encode(digest.encode()).decode() #PasswordDigest
    return 'UsernameToken Username="{0}",PasswordDigest="{1}",Nonce="{2}",Created="{3}"'.format(appKey,
```

```
digestBase64, nonce, now)

def main():
    # 请求Headers
    header = {'Authorization': 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
              'X-WSSE': buildWSSEHeader(APP_KEY, APP_SECRET)}
    # 请求Body
    jsonData = {'from': sender,
               'statusCallback': statusCallBack,
               'smsContent': [
                   {'to': receiver_1,
                    'templateId': TEMPLATE_ID_1,
                    'templateParas': TEMPLATE_PARAM_1,
                    'signature': signature_1 #使用国内短信通用模板时,必须填写签名名称
                   },
                   {'to': receiver_2,
                    'templateId': TEMPLATE_ID_2,
                    'templateParas': TEMPLATE_PARAM_2,
                    'signature': signature_2 #使用国内短信通用模板时,必须填写签名名称
                   }
               ]
    }
    print(header)

    # 为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    r = requests.post(url, json=jsonData, headers=header, verify=False)
    print(r.text) #打印响应信息

if __name__ == '__main__':
    main()
```

## 发送分批短信 ( 示例 2 )

```
# -*- coding: utf-8 -*-
import time
import uuid
import hashlib
import base64
import json
import ssl
import urllib.request

# 必填,请参考"开发准备"获取如下数据,替换为实际值
url = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1' #APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
APP_KEY = "c8RWg3ggEcyd4D3p94bf3Y7x1lle" #APP_Key
APP_SECRET = "q4li87Bh*****80SfD7Al" #APP_Secret
sender = "csms12345678" #国内短信签名通道号
TEMPLATE_ID_1 = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID1
TEMPLATE_ID_2 = "8ff55eac1d0b478ab3c06c3c6a492300" #模板ID2

#条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
signature_1 = "华为云短信测试" #签名名称1
signature_2 = "华为云短信测试" #签名名称2

# 必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
receiver_1 = ["+86151****6789", "+86152****7890"] #模板1的接收号码
receiver_2 = ["+86151****6789", "+86152****7890"] #模板2的接收号码

# 选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
statusCallBack = ""

"""
选填,使用无变量模板时请赋空值 TEMPLATE_PARAM = [];
单变量模板示例:模板内容为"您的验证码是${1}"时,TEMPLATE_PARAM可填写为["369751"]
双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,TEMPLATE_PARAM可填写为["3","人民公园正门"]
模板中的每个变量都必须赋值,且取值不能为空
"""
```

查看更多[模板规范](#)和[变量规范](#):产品介绍>短信模板须知和短信变量须知

```
"""
TEMPLATE_PARAM_1 = ["123456"] #模板1变量, 此处以单变量验证码短信为例, 请客户自行生成6位验证码,
并定义为字符串类型, 以杜绝首位0丢失的问题(例如: 002569变成了2569)。
TEMPLATE_PARAM_2 = ["234567"] #模板2变量, 此处以单变量验证码短信为例, 请客户自行生成6位验证码,
并定义为字符串类型, 以杜绝首位0丢失的问题(例如: 002569变成了2569)。

"""
构造X-WSSE参数值
@param appKey: string
@param appSecret: string
@return: string
"""
def buildWSSEHeader(appKey, appSecret):
    now = time.strftime('%Y-%m-%dT%H:%M:%SZ') #Created
    nonce = str(uuid.uuid4()).replace('-', '') #Nonce
    digest = hashlib.sha256((nonce + now + appSecret).encode()).hexdigest()

    digestBase64 = base64.b64encode(digest.encode()).decode() #PasswordDigest
    return 'UsernameToken Username="{0}",PasswordDigest="{1}",Nonce="{2}",Created="{3}"'.format(appKey,
    digestBase64, nonce, now)

def main():
    # 请求Body
    jsonData = json.dumps({'from': sender,
        'statusCallback': statusCallBack,
        'smsContent':[
            {'to':receiver_1,
            'templateId':TEMPLATE_ID_1,
            'templateParas':TEMPLATE_PARAM_1,
            'signature':signature_1 #使用国内短信通用模板时,必须填写签名名称
            },
            {'to':receiver_2,
            'templateId':TEMPLATE_ID_2,
            'templateParas':TEMPLATE_PARAM_2,
            'signature':signature_2 #使用国内短信通用模板时,必须填写签名名称
            }
        ])
    req = urllib.request.Request(url=url, data=jsonData, method='POST') #请求方法为POST
    # 请求Headers参数
    req.add_header('Authorization', 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"')
    req.add_header('X-WSSE', buildWSSEHeader(APP_KEY, APP_SECRET))
    req.add_header('Content-Type', 'application/json')

    # 为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    ssl_create_default_https_context = ssl._create_unverified_context

    try:
        r = urllib.request.urlopen(req) #发送请求
        print(r.read().decode('utf-8')) #打印响应信息
    except urllib.error.HTTPError as e:
        print(e.code)
        print(e.read().decode('utf-8'))
    except urllib.error.URLError as e:
        print(e.reason)

if __name__ == '__main__':
    main()
```

## 接收状态报告

```
## coding: utf-8 ##
import urllib.parse

# 短信平台上报状态报告数据样例(urlencode)
#success_body =
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";
```



```
failed_body =
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
f856-4376-afbf-570bd70a46ee_11840135&status=E200027";

"""
解析状态报告数据
@param data: 短信平台上报的状态报告数据
@return:
"""
def onSmsStatusReport(data):
    keyValues = urllib.parse.parse_qs(data); #解析状态报告数据

    """
    Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理

    'smsMsgId': 短信唯一标识
    'total': 长短信拆分条数
    'sequence': 拆分后短信序号
    'source': 状态报告来源
    'updateTime': 资源更新时间
    'status': 状态码
    """

    status = keyValues.get('status'); #状态报告枚举值
    # 通过status判断短信是否发送成功
    if 'DELIVRD' == str.upper(status[0]):
        print('Send sms success. smsMsgId: ', keyValues.get('smsMsgId')[0]);
    else:
        # 发送失败,打印status和orgCode
        print('Send sms failed. smsMsgId: ', keyValues.get('smsMsgId')[0]);
        print('Failed status: ', status[0]);

if __name__ == '__main__':
    # onSmsStatusReport(success_body)
    onSmsStatusReport(failed_body)
```

## 接收上行短信

```
# -*- coding: utf-8 -*-
import urllib.parse

# 上行短信通知样例(urlencode)
updata = "from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-c427-4525-8e73-
cf4a6ac5b3f7";

"""
解析上行短信通知数据
@param data: 短信平台推送的上行短信通知数据
@return:
"""
def onSmsUpData(data):
    keyValues = urllib.parse.parse_qs(data); #解析上行短信通知数据

    """
    Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理

    'smsMsgId': 上行短信唯一标识
    'from': 上行短信发送方的号码
    'to': 上行短信接收方的号码
    'body': 上行短信发送的内容
    """

    body = keyValues.get('body'); #上行短信发送的内容
    print('Sms up data. Body: ', body[0]);

if __name__ == '__main__':
    onSmsUpData(updata)
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>• 代码样例: <a href="#">Java/PHP/C#/Node.js/Go</a></li> <li>• <a href="#">发送短信API</a></li> <li>• <a href="#">发送分批短信API</a></li> <li>• <a href="#">接收状态报告API</a></li> <li>• <a href="#">接收上行短信API</a></li> <li>• <a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.2.4 C#

样例	<a href="#">发送短信（示例1）</a> 、 <a href="#">发送分批短信（示例1）</a> <a href="#">发送短信（示例2）</a> 、 <a href="#">发送分批短信（示例2）</a> <a href="#">接收状态报告</a> 、 <a href="#">接收上行短信</a>
环境要求	示例1：要求.NET Core 2.0及以上版本或.NET Framework 4.6及以上版本。 示例2：要求.NET Core 1.0及以上版本或.NET Framework 2.0及以上版本。
引用库	Newtonsoft.Json 11.0.2及以上版本，请参考 <a href="https://www.newtonsoft.com/json">https://www.newtonsoft.com/json</a> 获取。

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信（示例1）

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Security.Cryptography;
using System.Text;

namespace msgsms_csharp_demo
{
    class SendSms
    {
        static void Main(string[] args)
```

```
{
    //必填,请参考"开发准备"获取如下数据,替换为实际值
    string apiAddress = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1"; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
    // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
    string appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
    string appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
    string sender = "csms12345678"; //国内短信签名通道号
    string templateId = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID

    //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

    //string signature = "华为云短信测试"; //签名名称

    //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
    string receiver = "+86151****6789,+86152****7890"; //短信接收人号码

    //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
    string statusCallBack = "";

    /*
    * 选填,使用无变量模板时请赋空值 string templateParas = "";
    * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为"[\\"369751\\"]"
    * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为"[\\"3\\","\\"人民公园正门\\"]"
    * 模板中的每个变量都必须赋值,且取值不能为空
    * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
    */
    string templateParas = "[\\"369751\\"]"; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

    try
    {
        //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
        HttpClient client = new HttpClient();
        ServicePointManager.ServerCertificateValidationCallback = delegate { return true; };

        //请求Headers
        client.DefaultRequestHeaders.Add("Authorization", "WSSE realm=\\"SDP\\",profile=\\"UsernameToken\\",type=\\"Appkey\\"");
        client.DefaultRequestHeaders.Add("X-WSSE", BuildWSSEHeader(appKey, appSecret));
        //请求Body
        var body = new Dictionary<string, string>() {
            {"from", sender},
            {"to", receiver},
            {"templateId", templateId},
            {"templateParas", templateParas},
            {"statusCallback", statusCallBack},
            //{"signature", signature} //使用国内短信通用模板时,必须填写签名名称
        };

        HttpContent content = new FormUrlEncodedContent(body);

        var response = client.PostAsync(apiAddress, content).Result;
        Console.WriteLine(response.StatusCode); //打印响应结果码
        var res = response.Content.ReadAsStringAsync().Result;
        Console.WriteLine(res); //打印响应信息
    }
    catch (Exception e)
    {
        Console.WriteLine(e.StackTrace);
        Console.WriteLine(e.Message);
    }
}

/// <summary>
/// 构造X-WSSE参数值
```

```
/// </summary>
/// <param name="appKey"></param>
/// <param name="appSecret"></param>
/// <returns></returns>
static string BuildWSSEHeader(string appKey, string appSecret)
{
    string now = DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ"); //Created
    string nonce = Guid.NewGuid().ToString().Replace("-", ""); //Nonce

    byte[] material = Encoding.UTF8.GetBytes(nonce + now + appSecret);
    byte[] hashed = SHA256Managed.Create().ComputeHash(material);
    string hexdigest = BitConverter.ToString(hashed).Replace("-", "");
    string base64 = Convert.ToBase64String(Encoding.UTF8.GetBytes(hexdigest)); //PasswordDigest

    return String.Format("UsernameToken Username=\"{0}\",PasswordDigest=\"{1}\",Nonce=
\"{2}\",Created=\"{3}\"",
        appKey, base64, nonce, now);
}
}
```

## 发送短信（示例 2）

```
using System;
using System.Collections.Specialized;
using System.IO;
using System.Net;
using System.Security.Cryptography;
using System.Text;
using System.Web;

namespace msgsms_csharp_demo
{
    class SendSms
    {
        static void Main(string[] args)
        {
            //必填,请参考"开发准备"获取如下数据,替换为实际值
            string apiAddress = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/
v1"; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
            // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或
            者环境变量中密文存放,使用时解密,确保安全;
            string appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
            string appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
            string sender = "csms12345678"; //国内短信签名通道号
            string templateId = "8ff55eac1d0b478ab3c06c3c6a492300"; //模板ID

            //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与
            模板类型一致的签名名称

            //string signature = "华为云短信测试"; //签名名称

            //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
            string receiver = "+86151****6789,+86152****7890"; //短信接收人号码

            //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
            string statusCallBack = "";

            /*
            * 选填,使用无变量模板时请赋空值 string templateParas = "";
            * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为"[\\"369751\\"]"
            * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为"[\\"3\\","\\"人民公
            园正门\\"]"
            * 模板中的每个变量都必须赋值,且取值不能为空
            * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
            */
            string templateParas = "[\\"369751\\"]"; //模板变量,此处以单变量验证码短信为例,请客户自行生成6
            位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
        }
    }
}
```

```
try
{
    //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    ServicePointManager.ServerCertificateValidationCallback = delegate { return true; };
    //使用Tls1.2 = 3072
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | (SecurityProtocolType)3072;

    HttpWebRequest myReq = (HttpWebRequest)WebRequest.Create(apiAddress);
    //请求方法
    myReq.Method = "POST";
    //请求Headers
    myReq.ContentType = "application/x-www-form-urlencoded";
    myReq.Headers.Add("Authorization", "WSSE realm=\"SDP\",profile=\"UsernameToken\",type=
\"Appkey\"");
    myReq.Headers.Add("X-WSSE", BuildWSSEHeader(appKey, appSecret));
    //请求Body
    NameValueCollection keyValues = new NameValueCollection
    {
        {"from", sender},
        {"to", receiver},
        {"templateId", templateId},
        {"templateParas", templateParas},
        {"statusCallback", statusCallBack},
        /*{"signature", signature } //使用国内短信通用模板时,必须填写签名名称
    };
    string body = BuildQueryString(keyValues);

    //发送请求数据
    StreamWriter req = new StreamWriter(myReq.GetRequestStream());
    req.Write(body);
    req.Close();

    //获取响应数据
    HttpResponseMessage myResp = (HttpResponseMessage)myReq.GetResponse();
    StreamReader resp = new StreamReader(myResp.GetResponseStream());
    string result = resp.ReadToEnd();
    myResp.Close();
    resp.Close();

    Console.WriteLine(result);
}
catch (Exception e)
{
    Console.WriteLine(e.StackTrace);
    Console.WriteLine(e.Message);
}
}

/// <summary>
/// 构造X-WSSE参数值
/// </summary>
/// <param name="appKey"></param>
/// <param name="appSecret"></param>
/// <returns></returns>
static string BuildWSSEHeader(string appKey, string appSecret)
{
    string now = DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ"); //Created
    string nonce = Guid.NewGuid().ToString().Replace("-", ""); //Nonce

    byte[] material = Encoding.UTF8.GetBytes(nonce + now + appSecret);
    byte[] hashed = SHA256Managed.Create().ComputeHash(material);
    string hexdigest = BitConverter.ToString(hashed).Replace("-", "");
    string base64 = Convert.ToBase64String(Encoding.UTF8.GetBytes(hexdigest)); //PasswordDigest

    return String.Format("UsernameToken Username=\"{0}\",PasswordDigest=\"{1}\",Nonce=
\"{2}\",Created=\"{3}\"",
        appKey, base64, nonce, now);
}
```

```
/// <summary>
/// 构造请求body
/// </summary>
/// <param name="keyValues"></param>
/// <returns></returns>
static string BuildQueryString(NameValueCollection keyValues)
{
    StringBuilder temp = new StringBuilder();
    foreach (string item in keyValues.Keys)
    {
temp.Append(item).Append("=").Append(HttpUtility.UrlEncode(keyValues.Get(item))).Append("&");
    }
    return temp.Remove(temp.Length - 1, 1).ToString();
}
}
```

## 发送分批短信（示例 1）

```
using Newtonsoft.Json;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Security.Cryptography;
using System.Text;

namespace msgsms_csharp_demo
{
    class SendDiffSms
    {
        static void Main(string[] args)
        {
            //必填,请参考“开发准备”获取如下数据,替换为实际值
            string apiAddress = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1"; //APP接入地址(在控制台“应用管理”页面获取)+接口访问URI
            // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
            string appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
            string appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
            string sender = "csms12345678"; //国内短信签名通道号
            string templateId_1 = "979b639cbd0b4b6b88e0fd5de4ad6f85"; //模板ID1
            string templateId_2 = "979b639cbd0b4b6b88e0fd5de4ad6f85"; //模板ID2

            //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

            string signature_1 = "华为云短信测试"; //签名名称1
            string signature_2 = "华为云短信测试"; //签名名称2

            //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
            string[] receiver_1 = { "+86151****6789", "+86152****7890" }; //模板1的接收号码
            string[] receiver_2 = { "+86151****6789", "+86152****7890" }; //模板2的接收号码

            //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
            string statusCallBack = "";

            /*
            * 选填,使用无变量模板时请赋空值 string[] templateParas = {};
            * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为{"369751"}
            * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为{"3","人民公园
            正门"}
            * 模板中的每个变量都必须赋值,且取值不能为空
            * 查看更多模板规范和变量规范.产品介绍>短信模板须知和短信变量须知
            */
            string[] templateParas_1 = {"123456"}; //模板1变量,此处以单变量验证码短信为例,请客户自行生成
        }
    }
}
```

6位验证码，并定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。  
string[] templateParas\_2 = {"234567"}; //模板2变量，此处以单变量验证码短信为例，请客户自行生成  
6位验证码，并定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。

```
ArrayList smsContent = new ArrayList
{
    //smsContent,不携带签名名称时,signature请填写null
    InitDiffSms(receiver_1, templateId_1, templateParas_1, signature_1),
    InitDiffSms(receiver_2, templateId_2, templateParas_2, signature_2)
};

try
{
    //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    HttpClient client = new HttpClient();
    ServicePointManager.ServerCertificateValidationCallback = delegate { return true; };

    //请求Headers
    client.DefaultRequestHeaders.Add("Authorization", "WSSE realm=\"SDP\",profile=
\"UsernameToken\",type=\"AppKey\"");
    client.DefaultRequestHeaders.Add("X-WSSE", BuildWSSEHeader(appKey, appSecret));
    //请求Body
    var body = new Dictionary<string, object>{
        {"from", sender},
        {"statusCallback", statusCallBack},
        {"smsContent", smsContent}
    };

    HttpContent content = new StringContent(JsonConvert.SerializeObject(body));
    //请求Headers中的Content-Type参数
    content.Headers.ContentType = new MediaTypeHeaderValue("application/json");

    var response = client.PostAsync(apiAddress, content).Result;
    Console.WriteLine(response.StatusCode); //打印响应结果码
    var res = response.Content.ReadAsStringAsync().Result;
    Console.WriteLine(res); //打印响应信息
}
catch (Exception e)
{
    Console.WriteLine(e.StackTrace);
    Console.WriteLine(e.Message);
}

/// <summary>
/// 构造smsContent参数值
/// </summary>
/// <param name="receiver"></param>
/// <param name="templateId"></param>
/// <param name="templateParas"></param>
/// <param name="signature">签名名称,使用国内短信通用模板时填写</param>
/// <returns></returns>
static Dictionary<string, object> InitDiffSms(string[] receiver, string templateId, string[] templateParas,
string signature)
{
    Dictionary<string, object> dic = new Dictionary<string, object>
    {
        {"to", receiver},
        {"templateId", templateId},
        {"templateParas", templateParas}
    };
    if (!signature.Equals(null) && signature.Length > 0)
    {
        dic.Add("signature", signature);
    }

    return dic;
}
```

```
/// <summary>
/// 构造X-WSSE参数值
/// </summary>
/// <param name="appKey"></param>
/// <param name="appSecret"></param>
/// <returns></returns>
static string BuildWSSEHeader(string appKey, string appSecret)
{
    string now = DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ"); //Created
    string nonce = Guid.NewGuid().ToString().Replace("-", ""); //Nonce

    byte[] material = Encoding.UTF8.GetBytes(nonce + now + appSecret);
    byte[] hashed = SHA256Managed.Create().ComputeHash(material);
    string hexdigest = BitConverter.ToString(hashed).Replace("-", "");
    string base64 = Convert.ToBase64String(Encoding.UTF8.GetBytes(hexdigest)); //PasswordDigest

    return String.Format("UsernameToken Username=\"{0}\",PasswordDigest=\"{1}\",Nonce=
    \"{2}\",Created=\"{3}\"",
        appKey, base64, nonce, now);
}
}
```

## 发送分批短信（示例 2）

```
using Newtonsoft.Json;
using System;
using System.Collections;
using System.Collections.Generic;
using System.IO;
using System.Net;
using System.Security.Cryptography;
using System.Text;

namespace msgsms_csharp_demo
{
    class SendDiffSms
    {
        static void Main(string[] args)
        {
            //必填,请参考“开发准备”获取如下数据,替换为实际值
            string apiAddress = "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/
            v1"; //APP接入地址(在控制台“应用管理”页面获取)+接口访问URI
            // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或
            //者环境变量中密文存放,使用时解密,确保安全;
            string appKey = "c8RWg3ggEcyd4D3p94bf3Y7x1lle"; //APP_Key
            string appSecret = "q4li87Bh*****80SfD7Al"; //APP_Secret
            string sender = "csms12345678"; //国内短信签名通道号
            string templateId_1 = "979b639cbd0b4b6b88e0fd5de4ad6f85"; //模板ID1
            string templateId_2 = "979b639cbd0b4b6b88e0fd5de4ad6f85"; //模板ID2

            //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与
            //模板类型一致的签名名称

            string signature_1 = "华为云短信测试"; //签名名称1
            string signature_2 = "华为云短信测试"; //签名名称2

            //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
            string[] receiver_1 = { "+86151****6789", "+86152****7890" }; //模板1的接收号码
            string[] receiver_2 = { "+86151****6789", "+86152****7890" }; //模板2的接收号码

            //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
            string statusCallBack = "";

            /*
            * 选填,使用无变量模板时请赋空值 string[] templateParas = {};
            * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为{"369751"}
            * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为{"3","人民公园
            正门"}
            */
        }
    }
}
```



```
* 模板中的每个变量都必须赋值，且取值不能为空
* 查看更多模板规范和变量规范。产品介绍>短信模板须知和短信变量须知
*/
string[] templateParas_1 = {"123456"}; //模板1变量，此处以单变量验证码短信为例，请客户自行生成
6位验证码，并定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。
string[] templateParas_2 = {"234567"}; //模板2变量，此处以单变量验证码短信为例，请客户自行生成
6位验证码，并定义为字符串类型，以杜绝首位0丢失的问题（例如：002569变成了2569）。

ArrayList smsContent = new ArrayList
{
    //smsContent,不携带签名名称时,signature请填写null
    InitDiffSms(receiver_1, templateId_1, templateParas_1, signature_1),
    InitDiffSms(receiver_2, templateId_2, templateParas_2, signature_2)
};

try
{
    //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
    ServicePointManager.ServerCertificateValidationCallback = delegate { return true; };
    //使用Tls1.2 = 3072
    ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | (SecurityProtocolType)3072;

    HttpWebRequest myReq = (HttpWebRequest)WebRequest.Create(apiAddress);
    //请求方法
    myReq.Method = "POST";
    //请求Headers
    myReq.ContentType = "application/json";
    myReq.Headers.Add("Authorization", "WSSE realm=\"SDP\",profile=\"UsernameToken\",type=
\"Appkey\"");
    myReq.Headers.Add("X-WSSE", BuildWSSEHeader(appKey, appSecret));
    //请求Body
    var body = new Dictionary<string, object>() {
        {"from", sender},
        {"statusCallback", statusCallBack},
        {"smsContent", smsContent}
    };

    string jsonData = JsonConvert.SerializeObject(body);

    //发送请求数据
    StreamWriter req = new StreamWriter(myReq.GetRequestStream());
    req.Write(jsonData);
    req.Close();

    //获取响应数据
    HttpResponseMessage myResp = (HttpResponse)myReq.GetResponse();
    StreamReader resp = new StreamReader(myResp.GetResponseStream());
    string result = resp.ReadToEnd();
    myResp.Close();
    resp.Close();

    Console.WriteLine(result);
}
catch (Exception e)
{
    Console.WriteLine(e.StackTrace);
    Console.WriteLine(e.Message);
}
}

/// <summary>
/// 构造smsContent参数值
/// </summary>
/// <param name="receiver"></param>
/// <param name="templateId"></param>
/// <param name="templateParas"></param>
/// <param name="signature">签名名称,使用国内短信通用模板时填写</param>
/// <returns></returns>
static Dictionary<string, object> InitDiffSms(string[] receiver, string templateId, string[] templateParas,
```

```
string signature)
{
    Dictionary<string, object> dic = new Dictionary<string, object>
    {
        {"to", receiver},
        {"templateId", templateId},
        {"templateParas", templateParas}
    };
    if (!signature.Equals(null) && signature.Length > 0)
    {
        dic.Add("signature", signature);
    }

    return dic;
}

/// <summary>
/// 构造X-WSSE参数值
/// </summary>
/// <param name="appKey"></param>
/// <param name="appSecret"></param>
/// <returns></returns>
static string BuildWSSEHeader(string appKey, string appSecret)
{
    string now = DateTime.Now.ToString("yyyy-MM-ddTHH:mm:ssZ"); //Created
    string nonce = Guid.NewGuid().ToString().Replace("-", ""); //Nonce

    byte[] material = Encoding.UTF8.GetBytes(nonce + now + appSecret);
    byte[] hashed = SHA256Managed.Create().ComputeHash(material);
    string hexdigest = BitConverter.ToString(hashed).Replace("-", "");
    string base64 = Convert.ToBase64String(Encoding.UTF8.GetBytes(hexdigest)); //PasswordDigest

    return String.Format("UsernameToken Username=\"{0}\",PasswordDigest=\"{1}\",Nonce=
\"{2}\",Created=\"{3}\"",
        appKey, base64, nonce, now);
}
}
```

## 接收状态报告

```
using System;
using System.Web;

namespace msgsms_csharp_demo
{
    class Report
    {
        static void Main(string[] args)
        {
            //string success_body =
            "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
            f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";
            string failed_body =
            "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-
            f856-4376-afbf-570bd70a46ee_11840135&status=E200027";

            //OnSmsStatusReport(success_body);
            OnSmsStatusReport(failed_body);
        }

        /// <summary>
        /// 解析状态报告数据
        /// </summary>
        /// <param name="data">短信平台上报的状态报告数据</param>
        static void OnSmsStatusReport(string data)
        {
            var keyValues = HttpUtility.ParseQueryString(data); //解析状态报告数据
        }
    }
}
```

```
/**
 * Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理
 *
 * 'smsMsgId': 短信唯一标识
 * 'total': 长短信拆分条数
 * 'sequence': 拆分后短信序号
 * 'source': 状态报告来源
 * 'updateTime': 资源更新时间
 * 'status': 状态码
 */
string status = keyValues.Get("status"); // 状态报告枚举值
// 通过status判断短信是否发送成功
if ("DELIVRD".Equals(status.ToUpper()))
{
    Console.WriteLine("Send sms success. smsMsgId: " + keyValues.Get("smsMsgId"));
}
else
{
    // 发送失败,打印status和orgCode
    Console.WriteLine("Send sms failed. smsMsgId: " + keyValues.Get("smsMsgId"));
    Console.WriteLine("Failed status: " + status);
}
}
}
```

## 接收上行短信

```
using System;
using System.Web;

namespace msgsms_csharp_demo
{
    class UpData
    {
        static void Main(string[] args)
        {
            string updata = "from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-
c427-4525-8e73-cf4a6ac5b3f7";

            onSmsUpData(updata);
        }

        /// <summary>
        /// 解析上行短信通知数据
        /// </summary>
        /// <param name="data">短信平台推送的上行短信通知数据</param>
        static void onSmsUpData(string data)
        {
            var keyValues = HttpUtility.ParseQueryString(data); //解析上行短信通知数据

            /**
             * Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理
             *
             * 'smsMsgId': 上行短信唯一标识
             * 'from': 上行短信发送方的号码
             * 'to': 上行短信接收方的号码
             * 'body': 上行短信发送的内容
             */
            string body = keyValues.Get("body"); // 上行短信发送的内容
            Console.WriteLine("Sms up data. Body: " + body);
        }
    }
}
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>• 代码样例: <a href="#">Java/PHP/Python/Node.js/Go</a></li> <li>• <a href="#">发送短信API</a></li> <li>• <a href="#">发送分批短信API</a></li> <li>• <a href="#">接收状态报告API</a></li> <li>• <a href="#">接收上行短信API</a></li> <li>• <a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.2.5 Node.js

样例	<a href="#">发送短信</a> 、 <a href="#">发送分批短信</a> 、 <a href="#">接收状态报告</a> 、 <a href="#">接收上行短信</a>
环境要求	基于Node.js 8.12.0版本，要求Node.js 8.12.0及以上版本。

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信

```

/*jshint esversion: 6 */
var https = require('https'); //引入https模块
var url = require('url'); //引入url模块
var querystring = require('querystring'); // 引入querystring模块

//必填,请参考"开发准备"获取如下数据,替换为实际值
var realUrl = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1'; //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
var appKey = 'c8RWg3ggEcyd4D3p94bf3Y7x1lle'; //APP_Key
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
var appSecret = 'q4li87Bh*****80SfD7Al'; //APP_Secret
var sender = 'csms12345678'; //国内短信签名通道号
var templateId = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
var signature = "华为云短信测试"; //签名名称

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
var receiver = '+86151****6789,+86152****7890'; //短信接收人号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
var statusCallBack = ";

```

```
/**
 * 选填,使用无变量模板时请赋空值 var templateParas = "";
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为["369751"]
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为["3","人民公园正门"]
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
var templateParas = ["369751"]; //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并
定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

/**
 * 构造请求Body体
 *
 * @param sender
 * @param receiver
 * @param templateId
 * @param templateParas
 * @param statusCallBack
 * @param signature | 签名名称,使用国内短信通用模板时填写
 * @returns
 */
function buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature){
  if (null !== signature && signature.length > 0) {
    return querystring.stringify({
      'from': sender,
      'to': receiver,
      'templateId': templateId,
      'templateParas': templateParas,
      'statusCallback': statusCallBack,
      'signature': signature
    });
  }

  return querystring.stringify({
    'from': sender,
    'to': receiver,
    'templateId': templateId,
    'templateParas': templateParas,
    'statusCallback': statusCallBack
  });
}

/**
 * 构造X-WSSE参数值
 *
 * @param appKey
 * @param appSecret
 * @returns
 */
function buildWsseHeader(appKey, appSecret){
  var crypto = require('crypto');
  var util = require('util');

  var time = new Date(Date.now()).toISOString().replace(/.[0-9]+\Z/, 'Z'); //Created
  var nonce = crypto.randomBytes(64).toString('hex'); //Nonce
  var passwordDigestBase64Str = crypto.createHash('sha256').update(nonce + time +
  appSecret).digest('base64'); //PasswordDigest

  return util.format('UsernameToken Username="%s",PasswordDigest="%s",Nonce="%s",Created="%s"',
  appKey, passwordDigestBase64Str, nonce, time);
}

var urlObj = url.parse(realUrl); //解析realUrl字符串并返回一个 URL对象

var options = {
  host: urlObj.hostname, //主机名
  port: urlObj.port, //端口
  path: urlObj.pathname, //URI
}
```

```
method: 'POST', //请求方法为POST
headers: { //请求Headers
  'Content-Type': 'application/x-www-form-urlencoded',
  'Authorization': 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
  'X-WSSE': buildWsseHeader(appKey, appSecret)
},
rejectUnauthorized: false //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
};
// 请求Body,不携带签名名称时,signature请填写null
var body = buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature);

var req = https.request(options, (res) => {
  console.log('statusCode:', res.statusCode); //打印响应码

  res.setEncoding('utf8'); //设置响应数据编码格式
  res.on('data', (d) => {
    console.log('resp:', d); //打印响应数据
  });
});
req.on('error', (e) => {
  console.error(e.message); //请求错误时,打印错误信息
});
req.write(body); //发送请求Body数据
req.end(); //结束请求
```

## 发送分批短信

```
/*jshint esversion: 6 */
var https = require('https'); //引入https模块
var url = require('url'); //引入url模块

//必填,请参考"开发准备"获取如下数据,替换为实际值
var realUrl = 'https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1'; //APP接入地址
//在控制台"应用管理"页面获取+接口访问URI
var appKey = 'c8RWg3ggEcyd4D3p94bf3Y7x1lle'; //APP_Key
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
var appSecret = 'q4li87Bh*****80SfD7Al'; //APP_Secret
var sender = 'csms12345678'; //国内短信签名通道号
var templateId1 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID1
var templateId2 = '8ff55eac1d0b478ab3c06c3c6a492300'; //模板ID2

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称
var signature1 = "华为云短信测试"; //签名名称1
var signature2 = "华为云短信测试"; //签名名称2

//必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
var receiver1 = ['+86151****6789','+86152****7890']; //模板1的接收号码
var receiver2 = ['+86151****6789','+86152****7890']; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
var statusCallBack = "";

/**
 * 选填,使用无变量模板时请赋空值 var templateParas = [];
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为['369751']
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为['3','人民公园正门']
 * 模板中的每个变量都必须赋值,且取值不能为空
 * 查看更多模板和变量规范:产品介绍>模板和变量规范
 */
var templateParas1 = ['123456']; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
var templateParas2 = ['234567']; //模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

/**
 * 构造smsContent参数值
 */
```

```
* @param receiver
* @param templateId
* @param templateParas
* @param signature | 签名名称,使用国内短信通用模板时填写
* @returns
*/
function initDiffSms(receiver, templateId, templateParas, signature){
  if (null !== signature && signature.length > 0) {
    return {'to': receiver, 'templateId': templateId, 'templateParas': templateParas, 'signature': signature};
  }
  return {'to': receiver, 'templateId': templateId, 'templateParas': templateParas};
}

/**
* 构造X-WSSE参数值
*
* @param appKey
* @param appSecret
* @returns
*/
function buildWsseHeader(appKey, appSecret){
  var crypto = require('crypto');
  var util = require('util');

  var time = new Date(Date.now()).toISOString().replace(/.[0-9]+\Z/, 'Z'); //Created
  var nonce = crypto.randomBytes(64).toString('hex'); //Nonce
  var passwordDigestBase64Str = crypto.createHash('sha256').update(nonce + time +
  appSecret).digest('base64'); //PasswordDigest

  return util.format('UsernameToken Username="%s",PasswordDigest="%s",Nonce="%s",Created="%s"',
  appKey, passwordDigestBase64Str, nonce, time);
}

var body = JSON.stringify({ //请求Body
  'from': sender,
  'statusCallback': statusCallBack,
  'smsContent': [
    //smsContent,不携带签名名称时,signature请填写null
    initDiffSms(receiver1, templateId1, templateParas1, signature1),
    initDiffSms(receiver2, templateId2, templateParas2, signature2)
  ]
});

var urlObj = url.parse(realUrl); //解析realUrl字符串并返回一个 URL对象

var options = {
  host: urlObj.hostname, //主机名
  port: urlObj.port, //端口
  path: urlObj.pathname, //URI
  method: 'POST', //请求方法为POST
  headers: { //请求Headers
    'Content-Type': 'application/json',
    'Authorization': 'WSSE realm="SDP",profile="UsernameToken",type="Appkey"',
    'X-WSSE': buildWsseHeader(appKey, appSecret)
  },
  rejectUnauthorized: false //为防止因HTTPS证书认证失败造成API调用失败,需要先忽略证书信任问题
};

var req = https.request(options, (res) => {
  console.log('statusCode:', res.statusCode); //打印响应码

  res.setEncoding('utf8'); //设置响应数据编码格式
  res.on('data', (d) => {
    console.log('resp:', d); //打印响应数据
  });
});
req.on('error', (e) => {
  console.error(e.message); //请求错误时,打印错误信息
});
```

```
req.write(body); //发送请求Body数据  
req.end(); //结束请求
```

## 接收状态报告

```
/*jshint esversion: 6 */  
  
// 短信平台上报状态报告数据样例(urlencode)  
//var success_body =  
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-  
f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";  
var failed_body =  
"sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-  
f856-4376-afbf-570bd70a46ee_11840135&status=E200027";  
  
/**  
 * 解析状态报告数据  
 *  
 * @param data 短信平台上报的状态报告数据  
 * @returns  
 */  
function onSmsStatusReport(data) {  
    var querystring = require('querystring');  
    var keyValues = querystring.parse(data); // 解析状态报告数据  
  
    /**  
     * Example: 此处已解析status为例,请按需解析所需参数并自行实现相关处理  
     */  
    * 'smsMsgId': 短信唯一标识  
    * 'total': 长短信拆分条数  
    * 'sequence': 拆分后短信序号  
    * 'source': 状态报告来源  
    * 'updateTime': 资源更新时间  
    * 'status': 状态码  
    */  
    var status = keyValues.status; // 状态报告枚举值  
    // 通过status判断短信是否发送成功  
    if ('DELIVRD' === status.toUpperCase()) {  
        console.log('Send sms success. smsMsgId: ', keyValues.smsMsgId);  
    } else {  
        // 发送失败,打印status和orgCode  
        console.log('Send sms failed. smsMsgId: ', keyValues.smsMsgId);  
        console.log('Failed status: ', status);  
    }  
}  
  
// onSmsStatusReport(success_body);  
onSmsStatusReport(failed_body);
```

## 接收上行短信

```
/*jshint esversion: 6 */  
  
// 上行短信通知样例(urlencode)  
var updata = "from=%2B86151****6789&to=1069****019&body=*****&smsMsgId=9692b5be-  
c427-4525-8e73-cf4a6ac5b3f7";  
  
/**  
 * 解析上行短信通知数据  
 *  
 * @param data 短信平台推送的上行短信通知数据  
 * @returns  
 */  
function onSmsUpData(data) {  
    var querystring = require('querystring');  
    var keyValues = querystring.parse(data); // 解析上行短信通知数据  
  
    /**  
     * Example: 此处已解析body为例,请按需解析所需参数并自行实现相关处理  
     */
```



```

*
* 'smsMsgId': 上行短信唯一标识
* 'from': 上行短信发送方的号码
* 'to': 上行短信接收方的号码
* 'body': 上行短信发送的内容
*/
var body = keyValues.body; // 上行短信发送的内容
console.log('Sms up data. Body: ', body);
}
onSmsUpData(updata);

```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>代码样例: <a href="#">Java/PHP/Python/C#/Go</a></li> <li><a href="#">发送短信API</a></li> <li><a href="#">发送分批短信API</a></li> <li><a href="#">接收状态报告API</a></li> <li><a href="#">接收上行短信API</a></li> <li><a href="#">调测问题处理</a></li> </ul>	<a href="#">语音通话</a> <a href="#">隐私保护通话</a>

## 3.2.6 Go

样例	<a href="#">发送短信、发送分批短信</a> <a href="#">接收状态报告、接收上行短信</a>
环境要求	go1.11及以上版本
引用库	<a href="https://github.com/satori/go.uuid">github.com/satori/go.uuid</a>

### 须知

- 发送短信为单模板群发短信示例，发送分批短信为多模板群发短信示例。
- 本文档所述Demo在提供服务的过程中，可能会涉及个人数据的使用，建议您遵从国家的相关法律采取足够的措施，以确保用户的个人数据受到充分的保护。
- 本文档所述Demo仅用于功能演示，不允许客户直接进行商业使用。
- 本文档信息仅供参考，不构成任何要约或承诺。

## 发送短信

```

package main

import (
    "bytes"
    "crypto/sha256"
    "crypto/tls"
    "encoding/base64"

```

```
"fmt"
"github.com/satori/go.uuid"
"io/ioutil"
"net/http"
"net/url"
"strings"
"time"
)

//无需修改,用于格式化鉴权头域,给"X-WSSSE"参数赋值
const WSSE_HEADER_FORMAT = "UsernameToken Username=\"%s\",PasswordDigest=\"%s\",Nonce=\"%s\",Created=\"%s\""
//无需修改,用于格式化鉴权头域,给"Authorization"参数赋值
const AUTH_HEADER_VALUE = "WSSE realm=\"SDP\",profile=\"UsernameToken\",type=\"Appkey\""

func main() {
    //必填,请参考"开发准备"获取如下数据,替换为实际值
    apiAddress := "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendSms/v1" //APP接入地址
    //在控制台"应用管理"页面获取+接口访问URI
    // 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
    appKey := "c8RWg3ggEcyd4D3p94bf3Y7x1lle" //APP_Key
    appSecret := "q4li87Bh*****80SfD7Al" //APP_Secret
    sender := "csms12345678" //国内短信签名通道号
    templateId := "8ff55eac1d0b478ab3c06c3c6a492300" //模板ID

    //条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

    signature := "华为云短信测试" //签名名称

    //必填,全局号码格式(包含国家码),示例:+86151****6789,多个号码之间用英文逗号分隔
    receiver := "+86151****6789,+86152****7890" //短信接收人号码

    //选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
    statusCallBack := ""

    /*
    * 选填,使用无变量模板时请赋空值 string templateParas = "";
    * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为"[\\"369751\"]"
    * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为"[\\"3\\","\\"人民公园正门\"]"
    * 模板中的每个变量都必须赋值,且取值不能为空
    * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
    */
    templateParas := "[]" //模板变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

    body := buildRequestBody(sender,receiver,templateId,templateParas,statusCallBack,signature)
    headers := make(map[string]string)
    headers["Content-Type"] = "application/x-www-form-urlencoded"
    headers["Authorization"] = AUTH_HEADER_VALUE;
    headers["X-WSSSE"] = buildWsseHeader(appKey, appSecret);
    resp, err := post(apiAddress, []byte(body),headers)
    if err != nil {
        return
    }
    fmt.Println(resp);
}

/**
 * sender,receiver,templateId不能为空
 */
func buildRequestBody(sender, receiver, templateId, templateParas, statusCallBack, signature string) string {
    param := "from=" + url.QueryEscape(sender) + "&to=" + url.QueryEscape(receiver) + "&templateId=" + url.QueryEscape(templateId)
    if templateParas != "" {
        param += "&templateParas=" + url.QueryEscape(templateParas)
    }
}
```

```
    if statusCallBack != "" {
        param += "&statusCallBack=" + url.QueryEscape(statusCallBack)
    }
    if signature != "" {
        param += "&signature=" + url.QueryEscape(signature)
    }
    return param
}

func post(url string, param []byte, headers map[string]string)(string,error) {
    tr := &http.Transport{
        TLSClientConfig: &tls.Config{InsecureSkipVerify: true},
    }
    client := &http.Client{Transport: tr}

    req, err := http.NewRequest("POST",url, bytes.NewBuffer(param));
    if err != nil {
        return "", err
    }
    for key, header := range headers {
        req.Header.Set(key, header)
    }

    resp, err := client.Do(req)
    defer resp.Body.Close()
    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        return "", err
    }
    return string(body), nil;
}

func buildWsseHeader(appKey,appSecret string)string {
    var cTime = time.Now().Format("2006-01-02T15:04:05Z")
    var nonce = uuid.NewV4().String()
    nonce = strings.ReplaceAll(nonce,"-", "")

    h := sha256.New()
    h.Write([]byte(nonce + cTime + appSecret))
    passwordDigestBase64Str := base64.StdEncoding.EncodeToString(h.Sum(nil))

    return fmt.Sprintf(W SSE_HEADER_FORMAT,appKey,passwordDigestBase64Str,nonce, cTime);
}
```

## 发送分批短信

```
package main

import (
    "bytes"
    "crypto/sha256"
    "crypto/tls"
    "encoding/base64"
    "encoding/json"
    "fmt"
    "github.com/satori/go.uuid"
    "io/ioutil"
    "net/http"
    "strings"
    "time"
)

//无需修改,用于格式化鉴权头域,给"X-WSSE"参数赋值
const WSSE_HEADER_FORMAT = "UsernameToken Username=\"%s\",PasswordDigest=\"%s\",Nonce=\"%s\
\",Created=\"%s\"""
//无需修改,用于格式化鉴权头域,给"Authorization"参数赋值
const AUTH_HEADER_VALUE = "WSSE realm=\"SDP\",profile=\"UsernameToken\",type=\"Appkey\"""

func main() {
```

```
//必填,请参考"开发准备"获取如下数据,替换为实际值
url := "https://smsapi.cn-north-4.myhuaweicloud.com:443/sms/batchSendDiffSms/v1" //APP接入地址(在控制台"应用管理"页面获取)+接口访问URI
// 认证用的appKey和appSecret硬编码到代码中或者明文存储都有很大的安全风险,建议在配置文件或者环境变量中密文存放,使用时解密,确保安全;
appKey := "c8RWg3ggEcyd4D3p94bf3Y7x1lle" //APP_Key
appSecret := "q4li87Bh*****80SfD7Al" //APP_Secret
sender := "csms12345678" //国内短信签名通道号
templateId1 := "8ff55eac1d0b478ab3c06c3c6a492300" //模板ID1
templateId2 := "8ff55eac1d0b478ab3c06c3c6a492300" //模板ID2

//条件必填,国内短信关注,当templateId指定的模板类型为通用模板时生效且必填,必须是已审核通过的,与模板类型一致的签名名称

signature1 := "华为云短信测试" //签名名称1
signature2 := "华为云短信测试" //签名名称2

//必填,全局号码格式(包含国家码),示例:+8615123456789,多个号码之间用英文逗号分隔
receiver1 := []string{"+86151****6789", "+86152****7890"}; //模板1的接收号码
receiver2 := []string{"+86151****6789", "+86152****7890"}; //模板2的接收号码

//选填,短信状态报告接收地址,推荐使用域名,为空或者不填表示不接收状态报告
statusCallBack := "";

/**
 * 选填,使用无变量模板时请赋空值 templateParas := []string{}
 * 单变量模板示例:模板内容为"您的验证码是${1}"时,templateParas可填写为[]string{"369751"}
 * 双变量模板示例:模板内容为"您有${1}件快递请到${2}领取"时,templateParas可填写为[]string{"3","人民公园正门"}
 * ${DATE}${TIME}变量不允许取值为空,${TXT_20}变量可以使用英文空格或点号替代空值,${NUM_6}变量可以使用0替代空值
 * 查看更多模板规范和变量规范:产品介绍>短信模板须知和短信变量须知
 */
templateParas1 := []string{"123456"}; //模板1变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。
templateParas2 := []string{"234567"}; //模板2变量,此处以单变量验证码短信为例,请客户自行生成6位验证码,并定义为字符串类型,以杜绝首位0丢失的问题(例如:002569变成了2569)。

item1 := initDiffSms(receiver1, templateId1, templateParas1, signature1);
item2 := initDiffSms(receiver2, templateId2, templateParas2, signature2);

item := []map[string]interface{}{item1,item2}
body := buildRequestBody(sender, item, statusCallBack)

headers := make(map[string]string)
headers["Content-Type"] = "application/json;charset=utf-8"
headers["Authorization"] = AUTH_HEADER_VALUE;
headers["X-WSSSE"] = buildWsseHeader(appKey, appSecret);
resp, err := post(url,body,headers)
if err != nil {
    return
}
fmt.Println(resp);
}

func buildRequestBody(sender string, item []map[string]interface{}, statusCallBack string) []byte{
    body := make(map[string]interface{})
    body["smsContent"] = item
    body["from"] = sender
    if statusCallBack != "" {
        body["statusCallback"] = statusCallBack
    }
    res, _ := json.Marshal(body)
    return res;
}

func initDiffSms(reveiver []string, templateId string, templateParas []string, signature string)
map[string]interface{} {
    diffSms := make(map[string]interface{});
```

```
diffSms["to"] = receiver
diffSms["templateId"] = templateId
if templateParas != nil && len(templateParas) > 0 {
    diffSms["templateParas"] = templateParas
}
if signature != "" {
    diffSms["signature"] = signature
}
return diffSms;
}

func post(url string, param []byte, headers map[string]string)(string,error) {
    tr := &http.Transport{
        TLSClientConfig: &tls.Config{InsecureSkipVerify: true},
    }
    client := &http.Client{Transport: tr}

    req, err := http.NewRequest("POST",url, bytes.NewBuffer(param));
    if err != nil {
        return "", err
    }
    for key, header := range headers {
        req.Header.Set(key, header)
    }

    resp, err := client.Do(req)
    defer resp.Body.Close()
    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        return "", err
    }
    return string(body), nil;
}

func buildWsseHeader(appKey,appSecret string)string {

    var cTime = time.Now().Format("2006-01-02T15:04:05Z")
    var nonce = uuid.NewV4().String()
    nonce = strings.ReplaceAll(nonce,"-", "")

    h := sha256.New()
    h.Write([]byte(nonce + cTime + appSecret))
    passwordDigestBase64Str := base64.StdEncoding.EncodeToString(h.Sum(nil))

    return fmt.Sprintf(WSSSE_HEADER_FORMAT,appKey,passwordDigestBase64Str,nonce, cTime);
}
```

## 接收状态报告

```
package main

import (
    "fmt"
    "net/url"
    "strings"
)

func main() {
    // 短信平台上报状态报告数据样例(urlencode)
    //success_body :=
    "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-f856-4376-afbf-570bd70a46ee_11840135&status=DELIVRD";
    failed_body :=
    "sequence=1&total=1&updateTime=2018-10-31T08%3A43%3A41Z&source=2&smsMsgId=2ea20735-f856-4376-afbf-570bd70a46ee_11840135&status=E200027";
    //onSmsStatusReport(success_body);
    onSmsStatusReport(failed_body);
}
```

```
func onSmsStatusReport(data string) {
    ss, _ := url.QueryUnescape(data)
    params := strings.Split(ss, "&")
    keyValues := make(map[string]string)
    for i := range params {
        temp := strings.Split(params[i], "=")
        keyValues[temp[0]] = temp[1];
    }
    status := keyValues["status"];
    if status == "DELIVRD" {
        fmt.Println("Send sms success. smsMsgId: " + keyValues["smsMsgId"])
    } else {
        fmt.Println("Send sms failed. smsMsgId: " + keyValues["smsMsgId"])
        fmt.Println("Failed status: " + keyValues["status"])
    }
}
```

## 接收上行短信

```
package main

import (
    "fmt"
    "net/url"
    "strings"
)

func main() {
    upData := "from=%2B86151****6789&to=1069***2019&body=*****&smsMsgId=9692b5be-
c427-4525-8e73-cf4a6ac5b3f7";
    onSmsUpData(upData);
}

func onSmsUpData(data string) {
    ss, _ := url.QueryUnescape(data)
    params := strings.Split(ss, "&")
    keyValues := make(map[string]string)
    for i := range params {
        temp := strings.Split(params[i], "=")
        keyValues[temp[0]] = temp[1];
    }
    body := keyValues["body"];
    fmt.Println("Sms up data. Body: " + body)
}
```

## 相关推荐

相关文档	相关产品
<ul style="list-style-type: none"> <li>• 代码样例: <a href="#">Java/PHP/Python/C#/Node.js</a></li> <li>• 发送短信API</li> <li>• 发送分批短信API</li> <li>• 接收状态报告API</li> <li>• 接收上行短信API</li> <li>• 调测问题处理</li> </ul>	语音通话 隐私保护通话