

短信

SDK 文档

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

SDK 文档

SDK 下载

Java SDK

PHP SDK

Python SDK

Node.js SDK

C# SDK

Go SDK

C++ SDK

SDK 文档

SDK 下载

最近更新时间：2024-01-18 16:27:43

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

您可以通过以下方式获得相关指引和 SDK 源码：

平台	SDK 文档	Github 地址
Java	Java SDK 文档	Java SDK
PHP	PHP SDK 文档	PHP SDK
Python	Python SDK 文档	Python SDK
JavaScript	Node.js SDK 文档	Node.js SDK
C#	C# SDK 文档	C# SDK
Go	Go SDK 文档	Go SDK
C++	C++ SDK 文档	C++ SDK

注意：

本章节短信 SDK 均为最新 SDK 3.0，后续短信相关新增功能都会在此章节更新。我们强烈推荐您使用最新的 SDK 3.0 以及配套的 API 3.0 接口。

现有2.0版本的 API 和 SDK 功能依然保持，未来可能停止维护。

Java SDK

最近更新时间：2024-06-27 15:48:01

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本。

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口。

一次群发请求最多支持200个号码。

签名、正文模板相关接口。

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口，请将“个人认证”变更为“企业认证”。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

通过 Maven 安装（推荐）

Github 仓库参见：[通过 Maven 安装（推荐）](#)

Gitee 仓库参见：[通过 Maven 安装（推荐）](#)

通过源码包安装

Github 仓库参见：[通过源码包安装](#)

Gitee 仓库参见：[通过源码包安装](#)

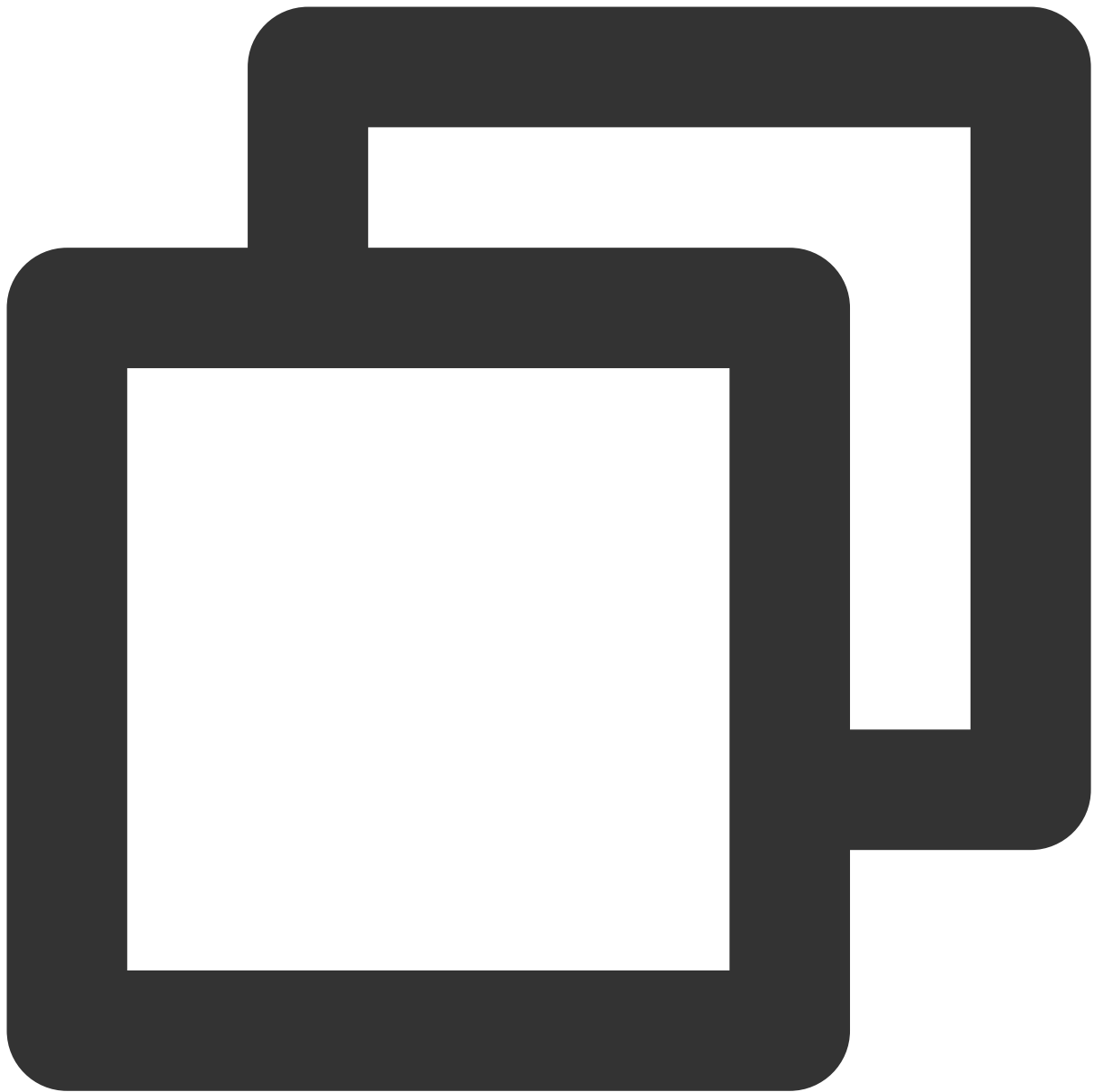
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



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

//导入可选配置类
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// 导入对应SMS模块的client
import com.tencentcloudapi.sms.v20210111.SmsClient;

// 导入要请求接口对应的request response类
```

```
import com.tencentcloudapi.sms.v20210111.models.SendSmsRequest;
import com.tencentcloudapi.sms.v20210111.models.SendSmsResponse;

/**
 * Tencent Cloud Sms Sendsms
 *
 */
public class SendSms
{
    public static void main(String[] args)
    {
        try {
            // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId, SecretKey。
            // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中，请参考凭证管理 https
            // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
            // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID")

            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            // 从3.0.96版本开始，单独设置 HTTP 代理（无需要直接忽略）
            // httpProfile.setProxyHost("真实代理ip");
            // httpProfile.setProxyPort(真实代理端口);
            httpProfile.setReqMethod("GET"); // get请求(默认为post请求)
            httpProfile.setConnTimeout(10); // 请求连接超时时间，单位为秒(默认60秒)
            httpProfile.setWriteTimeout(10); // 设置写入超时时间，单位为秒(默认0秒)
            httpProfile.setReadTimeout(10); // 设置读取超时时间，单位为秒(默认0秒)

            /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com ，也支持指定
            httpProfile.setEndpoint("sms.tencentcloudapi.com");

            /* 非必要步骤：
            * 实例化一个客户端配置对象，可以指定超时时间等配置 */
            ClientProfile clientProfile = new ClientProfile();
            /* SDK默认用TC3-HMAC-SHA256进行签名
            * 非必要请不要修改这个字段 */
            clientProfile.setSignMethod("HmacSHA256");
            clientProfile.setHttpProfile(httpProfile);
            /* 实例化要请求产品(以sms为例)的client对象
            * 第二个参数是地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填入
            SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile);
            /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
            * 你可以直接查询SDK源码确定接口有哪些属性可以设置
            * 属性可能是基本类型，也可能引用了另一个数据结构
            * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
            SendSmsRequest req = new SendSmsRequest();
```

```
/* 填充请求参数,这里request对象的成员变量即对应接口的入参
 * 你可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义
 * 基本类型的设置:
 * 帮助链接:
 * 短信控制台: https://console.intl.cloud.tencent.com/smsv2
 * sms helper: https://intl.cloud.tencent.com/document/product/382/3773

/* 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2
String sdkAppId = "2400006666";
req.setSmsSdkAppId(sdkAppId);

/* 短信签名内容: 使用 UTF-8 编码, 必须填写已审核通过的签名, 签名信息可登录 [短信控
String signName = "签名内容";
req.setSignName(signName);

/* 国际/港澳台短信 SenderId: 中国大陆地区短信填空, 默认未开通, 如需开通请联系 [sms
String senderid = "";
req.setSenderId(senderid);

/* 用户的 session 内容: 可以携带用户侧 ID 等上下文信息, server 会原样返回 */
String sessionContext = "xxx";
req.setSessionContext(sessionContext);

/* 短信号码扩展号: 默认未开通, 如需开通请联系 [sms helper] */
String extendCode = "";
req.setExtendCode(extendCode);

/* 模板 ID: 必须填写已审核通过的模板 ID。模板ID可登录 [短信控制台] 查看 */
String templateId = "400000";
req.setTemplateId(templateId);

/* 下发手机号码, 采用 E.164 标准, +[国家或地区码][手机号]
 * 示例如: +60198890000, 其中前面有一个+号, 60为国家码, 198890000为手机号, 最
String[] phoneNumberSet = {"+60198890000", "+60198890001", "+6019889000
req.setPhoneNumberSet(phoneNumberSet);

/* 模板参数: 若无模板参数, 则设置为空 */
String[] templateParamSet = {"5678"};
req.setTemplateParamSet(templateParamSet);

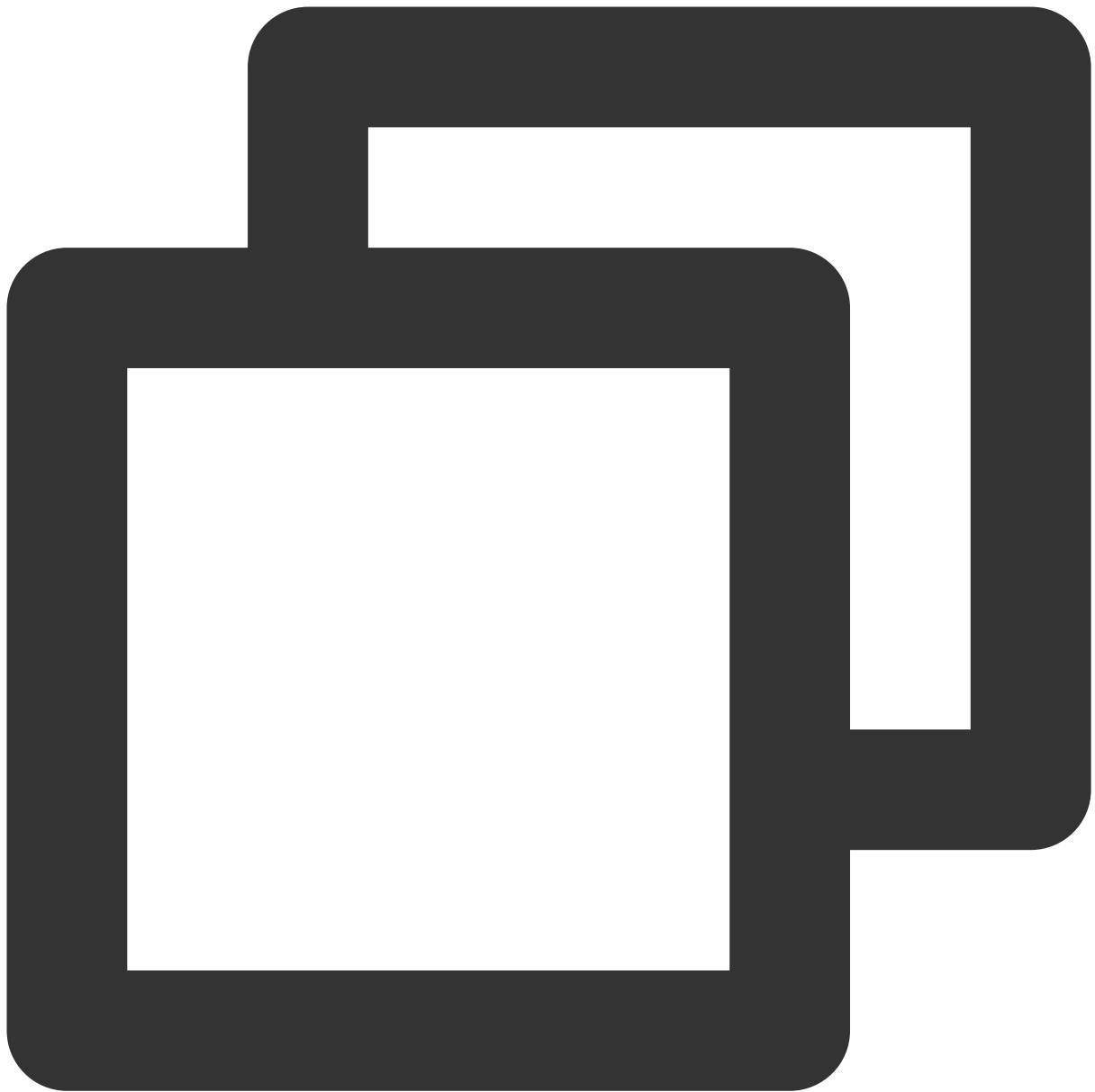
/* 通过 client 对象调用 SendSms 方法发起请求。注意请求方法名与请求对象是对应的
 * 返回的 res 是一个 SendSmsResponse 类的实例, 与请求对象对应 */
SendSmsResponse res = client.SendSms(req);

// 输出json格式的字符串回包
System.out.println(SendSmsResponse.toJsonString(res));
```

```
// 也可以取出单个值，你可以通过官网接口文档或跳转到response对象的定义处查看返回字段
System.out.println(res.getRequestId());

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

拉取回执状态



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

//导入可选配置类
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// 导入对应SMS模块的client
import com.tencentcloudapi.sms.v20210111.SmsClient;

// 导入要请求接口对应的request response类
```

```
import com.tencentcloudapi.sms.v20210111.models.PullSmsReplyStatusRequest;
import com.tencentcloudapi.sms.v20210111.models.PullSmsReplyStatusResponse;

/**
 * Tencent Cloud Sms PullSmsSendStatus
 *
 */
public class PullSmsSendStatus {
    public static void main(String[] args) {
        try {
            // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId, SecretKey。
            // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中，请参考凭证管理 https://cloud.tencent.com/document/product/598/38136
            // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
            // SecretId、SecretKey 查询: https://console.intl.cloud.tencent.com/cam/credentials
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"), System.getenv("TENCENTCLOUD_SECRET_KEY"));

            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            // 从3.0.96版本开始，单独设置 HTTP 代理（无需要直接忽略）
            // httpProfile.setProxyHost("真实代理ip");
            // httpProfile.setProxyPort(真实代理端口);
            httpProfile.setReqMethod("GET"); // get请求(默认为post请求)
            httpProfile.setConnTimeout(30); // 请求连接超时时间，单位为秒(默认60秒)
            httpProfile.setWriteTimeout(30); // 设置写入超时时间，单位为秒(默认0秒)
            httpProfile.setReadTimeout(30); // 设置读取超时时间，单位为秒(默认0秒)

            /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com，也支持指定
            httpProfile.setEndpoint("sms.tencentcloudapi.com"); */

            /* 非必要步骤：
            * 实例化一个客户端配置对象，可以指定超时时间等配置 */
            ClientProfile clientProfile = new ClientProfile();
            /* SDK默认用TC3-HMAC-SHA256进行签名
            * 非必要请不要修改这个字段 */
            clientProfile.setSignMethod("HmacSHA256");
            clientProfile.setHttpProfile(httpProfile);

            /* 实例化要请求产品(以sms为例)的client对象
            * 第二个参数是地域信息，根据您的选择的国际站地域，如您选择的是新加坡国际站，则应该填入
            SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile); */

            /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
            * 你可以直接查询SDK源码确定接口有哪些属性可以设置
            * 属性可能是基本类型，也可能引用了另一个数据结构
            * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
            PullSmsSendStatusRequest req = new PullSmsSendStatusRequest();
```

```
/* 填充请求参数,这里request对象的成员变量即对应接口的入参
 * 你可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义
 * 基本类型的设置:
 * 帮助链接:
 * 短信控制台: https://console.intl.cloud.tencent.com/smsv2
 * sms helper: https://intl.cloud.tencent.com/document/product/382/3773

/* 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2
String sdkAppId = "2400006666";
req.setSmsSdkAppId(sdkAppId);

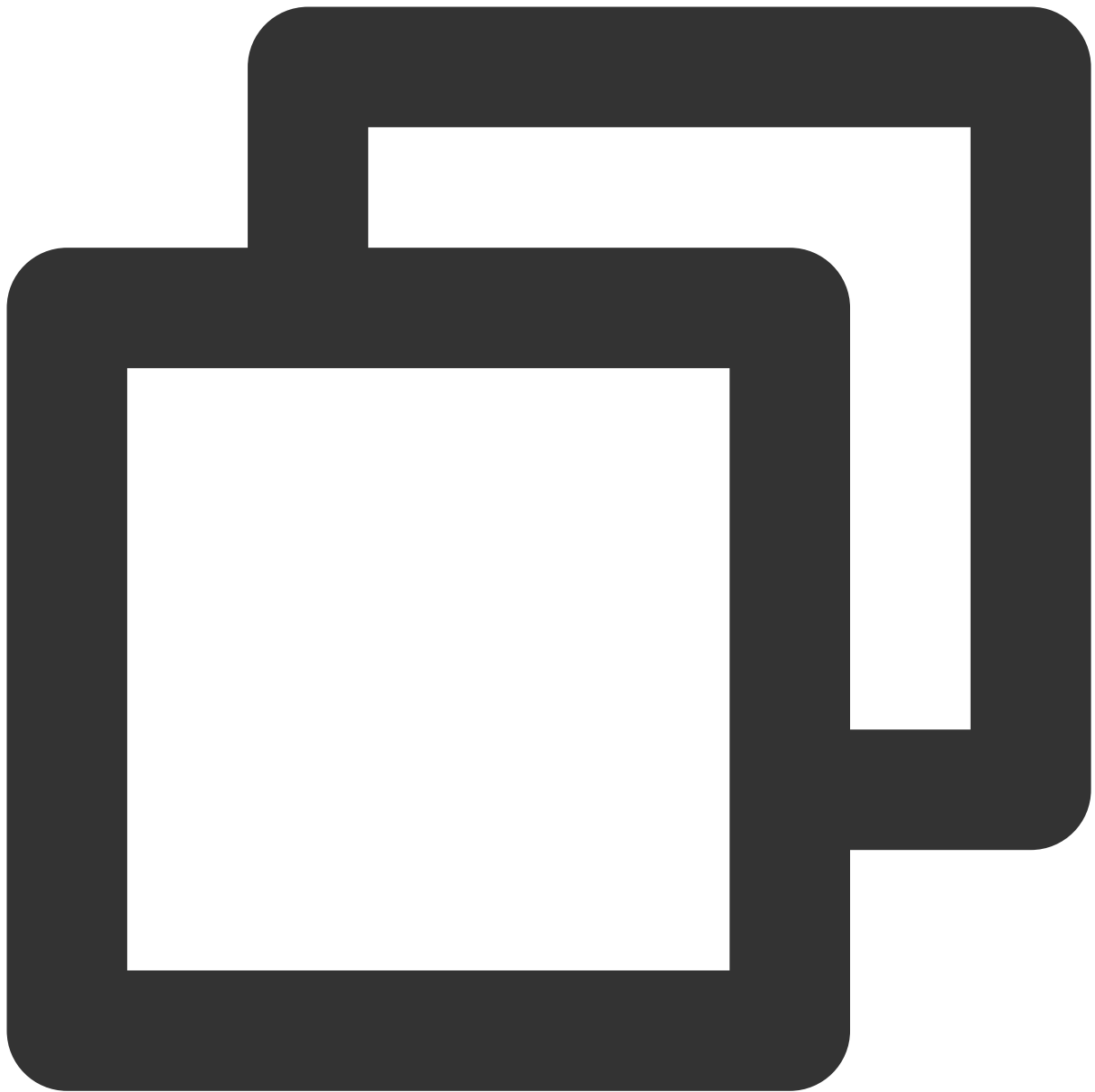
// 设置拉取最大条数, 最多100条
Long limit = 5L;
req.setLimit(limit);

/* 通过 client 对象调用 PullSmsSendStatus 方法发起请求。注意请求方法名与请求对象
 * 返回的 res 是一个 PullSmsSendStatusResponse 类的实例, 与请求对象对应 */
PullSmsSendStatusResponse res = client.PullSmsSendStatus(req);

// 输出json格式的字符串回包
System.out.println(PullSmsSendStatusResponse.toJsonString(res));

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

统计短信发送数据



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

//导入可选配置类
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;

// 导入对应SMS模块的client
import com.tencentcloudapi.sms.v20210111.SmsClient;

// 导入要请求接口对应的request response类
```

```
import com.tencentcloudapi.sms.v20210111.models.SendStatusStatisticsRequest;
import com.tencentcloudapi.sms.v20210111.models.SendStatusStatisticsResponse;

/**
 * Tencent Cloud Sms SendStatusStatistics
 *
 */
public class SendStatusStatistics {
    public static void main(String[] args) {
        try {
            // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId, SecretKey。
            // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中，请参考凭证管理 https://cloud.tencent.com/document/product/598/38138
            // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
            // SecretId、SecretKey 查询: https://console.intl.cloud.tencent.com/cam/credentials
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"), System.getenv("TENCENTCLOUD_SECRET_KEY"));

            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            // 从3.0.96版本开始，单独设置 HTTP 代理（无需要直接忽略）
            // httpProfile.setProxyHost("真实代理ip");
            // httpProfile.setProxyPort(真实代理端口);
            httpProfile.setReqMethod("GET"); // get请求(默认为post请求)
            httpProfile.setConnTimeout(30); // 请求连接超时时间，单位为秒(默认60秒)
            httpProfile.setWriteTimeout(30); // 设置写入超时时间，单位为秒(默认0秒)
            httpProfile.setReadTimeout(30); // 设置读取超时时间，单位为秒(默认0秒)

            /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com，也支持指定
            httpProfile.setEndpoint("sms.tencentcloudapi.com"); */

            /* 非必要步骤：
            * 实例化一个客户端配置对象，可以指定超时时间等配置 */
            ClientProfile clientProfile = new ClientProfile();
            /* SDK默认用TC3-HMAC-SHA256进行签名
            * 非必要请不要修改这个字段 */
            clientProfile.setSignMethod("HmacSHA256");
            clientProfile.setHttpProfile(httpProfile);

            /* 实例化要请求产品(以sms为例)的client对象
            * 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入
            SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile); */

            /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
            * 你可以直接查询SDK源码确定接口有哪些属性可以设置
            * 属性可能是基本类型，也可能引用了另一个数据结构
            * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
            SendStatusStatisticsRequest req = new SendStatusStatisticsRequest();
```

```
/* 填充请求参数,这里request对象的成员变量即对应接口的入参
 * 你可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义
 * 基本类型的设置:
 * 帮助链接:
 * 短信控制台: https://console.intl.cloud.tencent.com/smsv2
 * sms helper: https://intl.cloud.tencent.com/document/product/382/3773

/* 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2
String sdkAppId = "2400006666";
req.setSmsSdkAppId(sdkAppId);

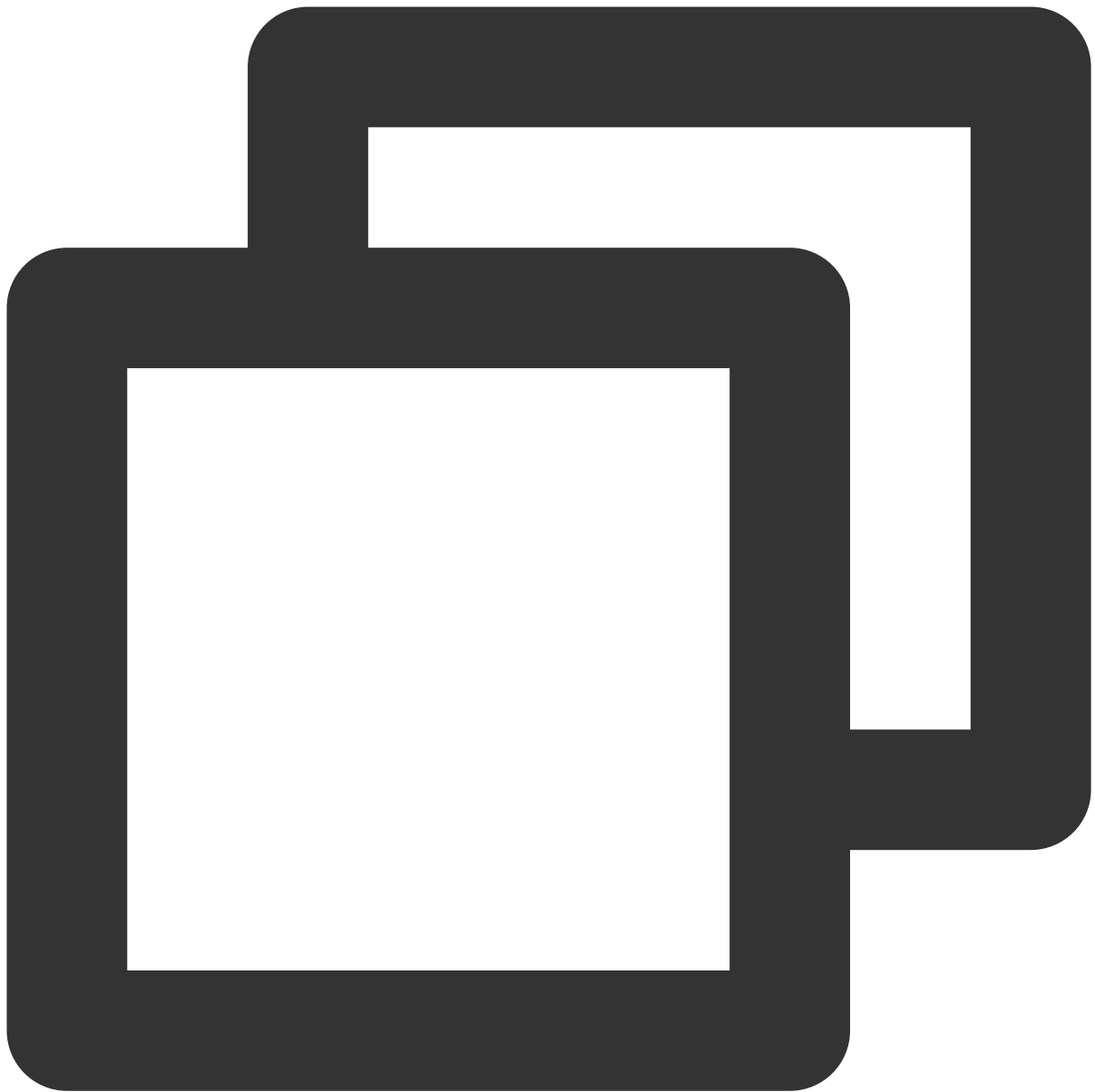
// 设置拉取最大条数, 最多100条
Long limit = 5L;
req.setLimit(limit);
/* 偏移量 注:目前固定设置为0 */
Long offset = 0L;
req.setOffset(offset);
/* 开始时间, yyyyymmddhh 需要拉取的起始时间, 精确到小时 */
String beginTime = "2019071100";
req.setBeginTime(beginTime);
/* 结束时间, yyyyymmddhh 需要拉取的截止时间, 精确到小时
 * 注:EndTime 必须大于 beginTime */
String endTime = "2019071123";
req.setEndTime(endTime);

/* 通过 client 对象调用 SendStatusStatistics 方法发起请求。注意请求方法名与请求
 * 返回的 res 是一个 SendStatusStatisticsResponse 类的实例, 与请求对象对应 */
SendStatusStatisticsResponse res = client.SendStatusStatistics(req);

// 输出json格式的字符串回包
System.out.println(SendStatusStatisticsResponse.toJsonString(res));

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

申请短信模板



```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
//导入可选配置类
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
// 导入 SMS 模块的 client
import com.tencentcloudapi.sms.v20210111.SmsClient;
// 导入要请求接口对应的 request response 类
import com.tencentcloudapi.sms.v20210111.models.AddSmsTemplateRequest;
import com.tencentcloudapi.sms.v20210111.models.AddSmsTemplateResponse;
/**
```

```
* Tencent Cloud Sms AddSmsTemplate
*
*/
public class AddSmsTemplate
{
    public static void main( String[] args )
    {
        try {
            // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId, SecretKey。
            // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中，请参考凭证管理 https://
            // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
            // SecretId、SecretKey 查询：https://consoleintl.cloud.tencent.com/cam/ca
            // Credential cred = new Credential("SecretId", "SecretKey");
            Credential cred = new Credential(System.getenv("TENCENTCLOUD_SECRET_ID"),

            // 实例化一个http选项，可选的，没有特殊需求可以跳过
            HttpProfile httpProfile = new HttpProfile();
            // 从3.0.96版本开始，单独设置 HTTP 代理（无需要直接忽略）
            // httpProfile.setProxyHost("真实代理ip");
            // httpProfile.setProxyPort(真实代理端口);
            httpProfile.setReqMethod("GET"); // get请求(默认为post请求)
            httpProfile.setConnTimeout(30); // 请求连接超时时间，单位为秒(默认60秒)
            httpProfile.setWriteTimeout(30); // 设置写入超时时间，单位为秒(默认0秒)
            httpProfile.setReadTimeout(30); // 设置读取超时时间，单位为秒(默认0秒)

            /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com ，也支持指定地
            httpProfile.setEndpoint("sms.tencentcloudapi.com");

            /* 非必要步骤：
            * 实例化一个客户端配置对象，可以指定超时时间等配置 */
            ClientProfile clientProfile = new ClientProfile();
            /* SDK 默认使用 TC3-HMAC-SHA256 进行签名
            * 非必要请不要修改该字段 */
            clientProfile.setSignMethod("HmacSHA256");
            clientProfile.setHttpProfile(httpProfile);
            /* 实例化 SMS 的 client 对象
            * 第二个参数是地域信息，根据您的选择的国际站地域，如您选择的是新加坡国际站，则应该填入字
            SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile);
            /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
            * 您可以直接查询 SDK 源码确定接口有哪些属性可以设置
            * 属性可能是基本类型，也可能引用了另一个数据结构
            * 推荐使用 IDE 进行开发，可以方便地跳转查阅各个接口和数据结构的文档说明 */
            AddSmsTemplateRequest req = new AddSmsTemplateRequest();
            /* 填充请求参数，这里 request 对象的成员变量即对应接口的入参
            * 您可以通过官网接口文档或跳转到 request 对象的定义处查看请求参数的定义
            * 基本类型的设置：
            * 帮助链接：
```

```
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    * sms helper：https://intl.cloud.tencent.com/document/product/382/3773 */
    /* 模板名称*/
    String templatename = "腾讯云";
    req.setTemplateName(templatename);
    /* 模板内容 */
    String templatecontent = "{1}为您的登录验证码，请于{2}分钟内填写，如非本人操作，请";
    req.setTemplateContent(templatecontent);
    /* 短信类型，1表示营销短信，2表示通知短信，3表示验证码短信 */
    long smstype = 3;
    req.setSmsType(smstype);
    /* 是否国际/港澳台短信：0：表示中国大陆地区短信，1：表示国际/港澳台短信。 */
    long international = 0;
    req.setInternational(international);
    /* 模板备注：例如申请原因，使用场景等 */
    String remark = "xxx";
    req.setRemark(remark);
    /* 通过 client 对象调用 AddSmsTemplate 方法发起请求。注意请求方法名与请求对象是对
    * 返回的 res 是一个 AddSmsTemplateResponse 类的实例，与请求对象对应 */
    AddSmsTemplateResponse res = client.AddSmsTemplate(req);
    // 输出 JSON 格式的字符串回包
    System.out.println(AddSmsTemplateResponse.toJsonString(res));
    // 可以取出单个值，您可以通过官网接口文档或跳转到 response 对象的定义处查看返回字段的
    System.out.println(res.getRequestId());
} catch (TencentCloudSDKException e) {
    e.printStackTrace();
}
}
```

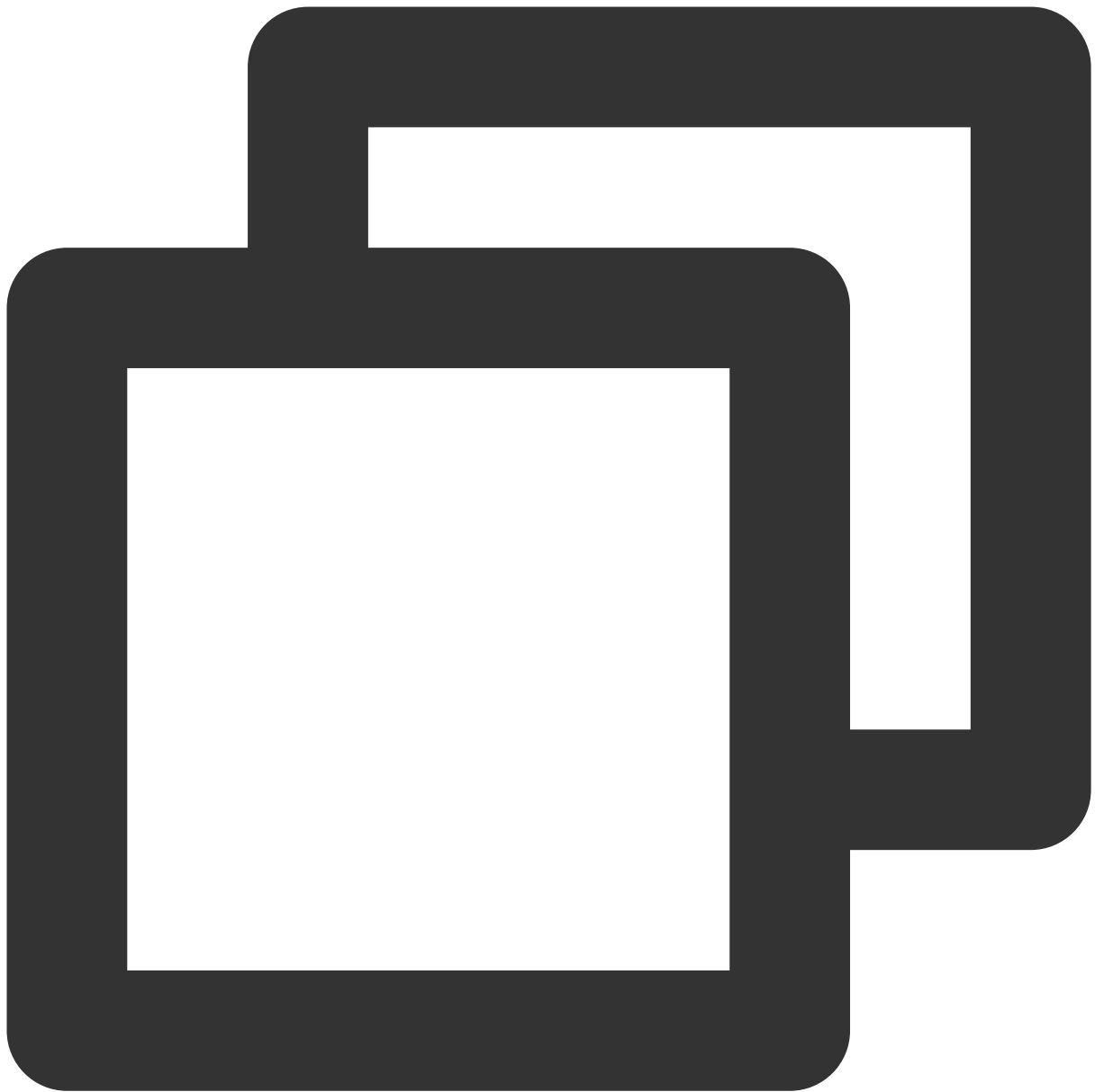
常见问题

java.lang.NoSuchMethodError: xxx.setSkipSign 问题

部分用户可能使用时遇到报错：`java.lang.NoSuchMethodError: xxx.setSkipSign`。这是因为 `tencentcloud-sdk-java-common` 包和 `tencentcloud-sdk-java-sms` 的版本不一致导致的。

该问题可能是 `pom` 中指定的 `common` 包版本有误，也可能是因为引用了其他第三方 `sdk` 而间接引用了不匹配的 `common` 版本导致的。

解决方式是在 `pom.xml` 中显式指定相同版本的 `common` 包版本，如当 `tencentcloud-sdk-java-sms` 的版本是 `3.1.1000`，可如下指定 `common` 包版本：



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

kotlin 问题

部分用户可能使用时遇到报错：`java.lang.NoSuchMethodError:`

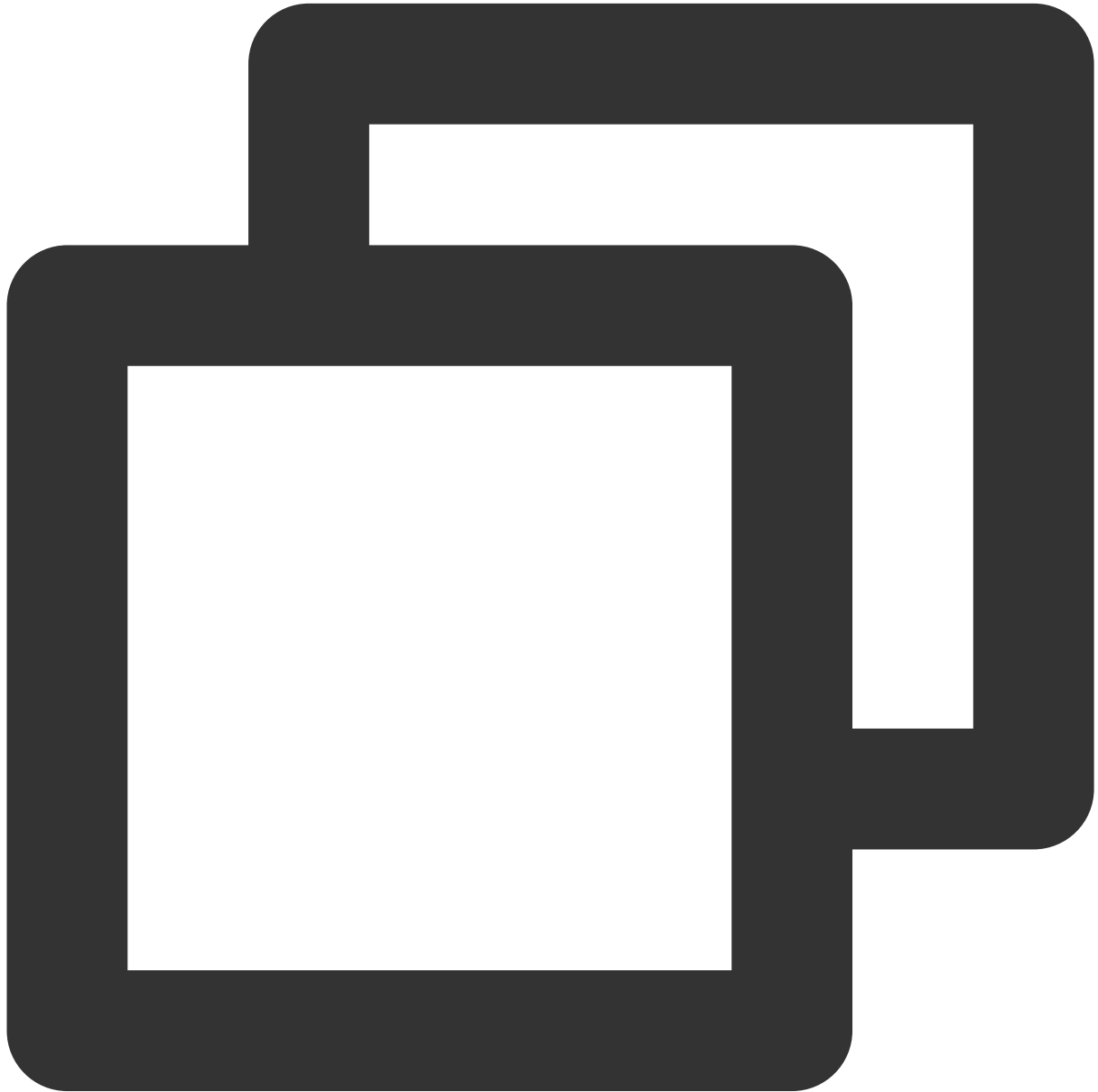
`kotlin.collections.ArraysKt.copyInto`。这是因为 kotlin 运行环境版本较低导致，可尝试升级 kotlin 版本解决。

更新仓库 pom.xml 文件里面的依赖失败

可能是因为本机配置了代理，而工具在更新时未进行代理的配置导致，按照上文在命令端更新依赖，如果还是失败，这时候需要看是否因为网络不通还是防火墙拦截。

运行示例失败

```
[TencentCloudSDKException]message:java.net.ConnectException-Connection timed out:  
connect requestId: 这里需要排查：是否本机配置了代理，而未在代码中加入代理，代理的加入可参考：
```



```
HttpProfile httpProfile = new HttpProfile();  
httpProfile.setProxyHost("真实代理ip");  
httpProfile.setProxyPort(真实代理端口);
```

版本升级

请注意，从3.0.x版本升级到3.1.x版本有兼容性问题，对于 **Integer** 字段的使用修改为了 **Long** 类型，需要重新编译项目。

证书问题

证书问题通常是客户端环境配置错误导致的。SDK 没有对证书进行操作，依赖的是 **Java** 运行环境本身的处理。出现证书问题后，可以使用 `-Djavax.net.debug=ssl` 开启详细日志辅助判断。

有用户报告使用 **IBM JDK 1.8** 出现证书报错：`javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_failure`，使用 **Oracle JDK** 后问题消失。

PHP SDK

最近更新时间：2024-06-27 15:48:01

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本。

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口。

一次群发请求最多支持200个号码。

签名、正文模板相关接口。

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口，请将“个人认证”变更为“企业认证”。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

通过 Composer 安装

Github 仓库参见：[通过 Composer 安装](#)

Gitee 仓库参见：[通过 Composer 安装](#)

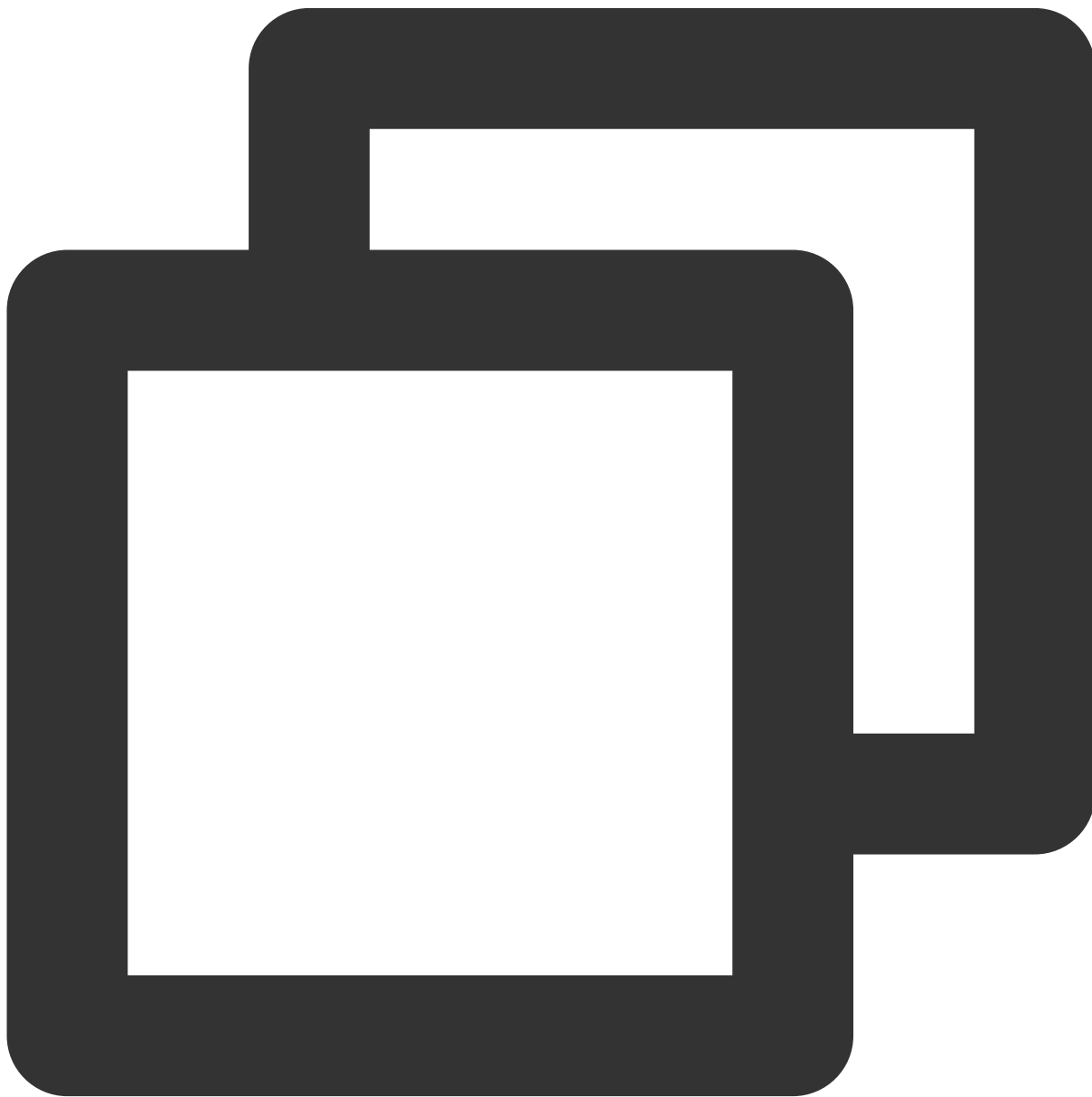
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



```
<?php
require_once '/path/to/vendor/autoload.php';
// 导入对应产品模块的client
use TencentCloud\Sms\V20210111\SmsClient;
// 导入要请求接口对应的Request类
use TencentCloud\Sms\V20210111\Models\SendSmsRequest;
use TencentCloud\Common\Exception\TencentCloudSDKException;
use TencentCloud\Common\Credential;
// 导入可选配置类
use TencentCloud\Common\Profile\ClientProfile;
use TencentCloud\Common\Profile\HttpProfile;

try {
    // 实例化一个证书对象, 入参需要传入腾讯云账户 SecretId, SecretKey
    // 为了保护密钥安全, 建议将密钥设置在环境变量中或者配置文件中。
    // 硬编码密钥到代码中有可能随代码泄露而暴露, 有安全隐患, 并不推荐。
    // SecretId、SecretKey 查询: https://console.intl.cloud.tencent.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项, 可选的, 没有特殊需求可以跳过
    $httpProfile = new HttpProfile();
    // 配置代理
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET"); // get请求(默认为post请求)
    $httpProfile->setReqTimeout(10); // 请求超时时间, 单位为秒(默认60秒)
    $httpProfile->setEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名(默认就近

    // 实例化一个client选项, 可选的, 没有特殊需求可以跳过
    $clientProfile = new ClientProfile();
    $clientProfile->setSignMethod("TC3-HMAC-SHA256"); // 指定签名算法
    $clientProfile->setHttpProfile($httpProfile);

    // 实例化要请求产品(以sms为例)的client对象, clientProfile是可选的
    // 第二个参数是地域信息, 根据您的选择的国际站地域, 如您选择的是新加坡国际站, 则应该填入字符串ap-
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

    // 实例化一个 sms 发送短信请求对象, 每个接口都会对应一个request对象。
    $req = new SendSmsRequest();

    /* 填充请求参数, 这里request对象的成员变量即对应接口的入参
    * 你可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义
    * 基本类型的设置:
    * 帮助链接:
    * 短信控制台: https://console.intl.cloud.tencent.com/smsv2
```

```
* sms helper: https://intl.cloud.tencent.com/document/product/382/3773 */

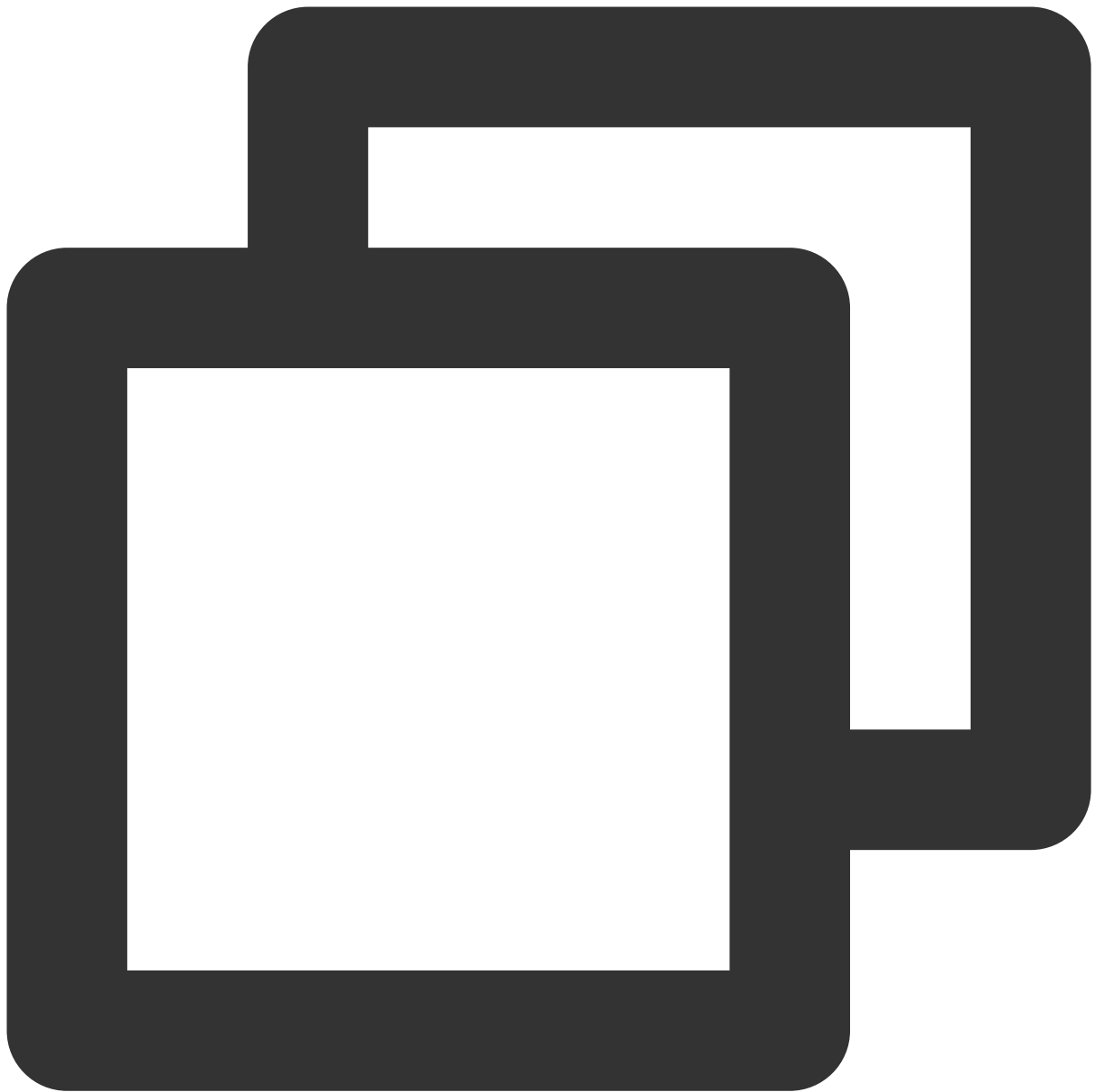
/* 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2400006666 */
$req->SmsSdkAppId = "2400006666";
/* 短信签名内容: 使用 UTF-8 编码, 必须填写已审核通过的签名, 签名信息可登录 [短信控制台] 查看 */
$req->SignName = "xxx";
/* 短信码号扩展号: 默认未开通, 如需开通请联系 [sms helper] */
$req->ExtendCode = "";
/* 下发手机号码, 采用 E.164 标准, +[国家或地区码][手机号]
 * 示例如:+60198890000, 其中前面有一个+号 , 60为国家码, 198890000为手机号, 最多不要超过
 */
$req->PhoneNumberSet = array("+60198890000");
/* 国际/港澳台短信 SenderId: 中国大陆地区短信填空, 默认未开通, 如需开通请联系 [sms helper] */
$req->SenderId = "";
/* 用户的 session 内容: 可以携带用户侧 ID 等上下文信息, server 会原样返回 */
$req->SessionContext = "xxx";
/* 模板 ID: 必须填写已审核通过的模板 ID。模板ID可登录 [短信控制台] 查看 */
$req->TemplateId = "449739";
/* 模板参数: 若无模板参数, 则设置为空*/
$req->TemplateParamSet = array("0");

// 通过client对象调用SendSms方法发起请求。注意请求方法名与请求对象是对应的
// 返回的resp是一个SendSmsResponse类的实例, 与请求对象对应
$res = $client->SendSms($req);

// 输出json格式的字符串回包
print_r($res->toJsonString());

// 也可以取出单个值。
// 您可以通过官网接口文档或跳转到response对象的定义处查看返回字段的定义
print_r($res->TotalCount);
}
catch (TencentCloudSDKException $e) {
    echo $e;
}
```

拉取回执状态



```
<?php
require_once '/path/to/vendor/autoload.php';
// 导入对应产品模块的client
use TencentCloud\Sms\V20210111\SmsClient;
// 导入要请求接口对应的Request类
use TencentCloud\Sms\V20210111\Models\PullSmsSendStatusRequest;
use TencentCloud\Common\Exception\TencentCloudSDKException;
use TencentCloud\Common\Credential;
// 导入可选配置类
use TencentCloud\Common\Profile\ClientProfile;
use TencentCloud\Common\Profile\HttpProfile;
```

```
try {
    // 实例化一个证书对象, 入参需要传入腾讯云账户 SecretId, SecretKey
    // 为了保护密钥安全, 建议将密钥设置在环境变量中或者配置文件中。
    // 硬编码密钥到代码中有可能随代码泄露而暴露, 有安全隐患, 并不推荐。
    // SecretId、SecretKey 查询: https://console.intl.cloud.tencent.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项, 可选的, 没有特殊需求可以跳过
    $httpProfile = new HttpProfile();
    // 配置代理
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET"); // get请求 (默认为post请求)
    $httpProfile->setReqTimeout(30); // 请求超时时间, 单位为秒 (默认60秒)
    $httpProfile->setEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名 (默认就近)

    // 实例化一个client选项, 可选的, 没有特殊需求可以跳过
    $clientProfile = new ClientProfile();
    $clientProfile->setSignMethod("TC3-HMAC-SHA256"); // 指定签名算法
    $clientProfile->setHttpProfile($httpProfile);

    // 实例化要请求产品 (以sms为例) 的client对象, clientProfile是可选的
    // 第二个参数是地域信息, 根据您的选择的国际站地域, 如您选择的是新加坡国际站, 则应该填入字符串ap-
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

    // 实例化一个 sms 发送短信请求对象, 每个接口都会对应一个request对象。
    $req = new PullSmsSendStatusRequest();

    /* 填充请求参数, 这里request对象的成员变量即对应接口的入参
    * 你可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义
    * 基本类型的设置:
    * 帮助链接:
    * 短信控制台: https://console.intl.cloud.tencent.com/smsv2
    * sms helper: https://intl.cloud.tencent.com/document/product/382/3773 */

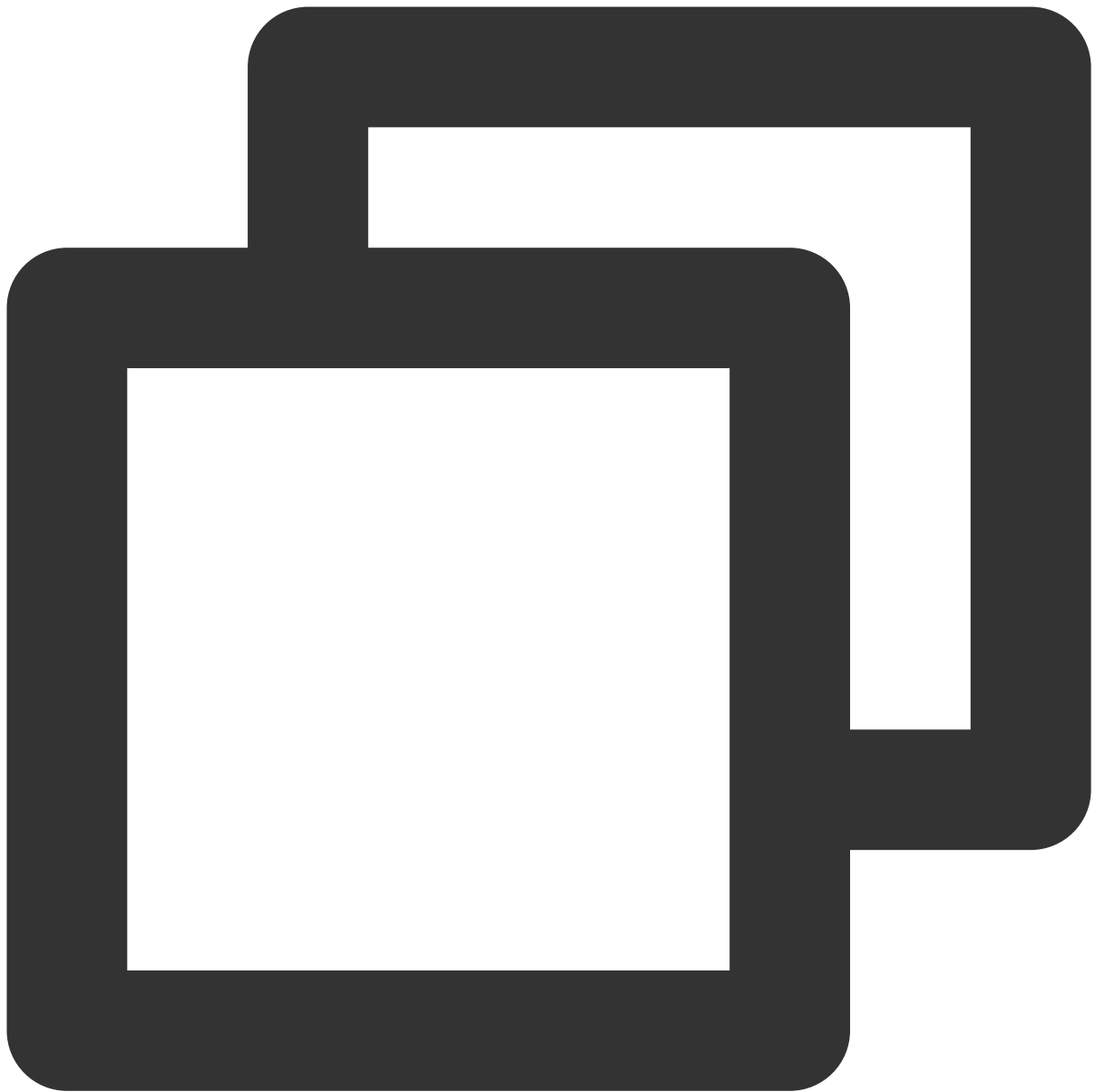
    /* 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2400006666
    $req->SmsSdkAppId = "2400006666";
    /* 拉取最大条数, 最多100条 */
    $req->Limit = 10;

    // 通过client对象调用PullSmsSendStatus方法发起请求。注意请求方法名与请求对象是对应的
    // 返回的resp是一个PullSmsSendStatusResponse类的实例, 与请求对象对应
    $resp = $client->PullSmsSendStatus($req);
```

```
// 输出json格式的字符串回包
print_r($resp->toJsonString());

// 也可以取出单个值。
// 你可以通过官网接口文档或跳转到response对象的定义处查看返回字段的定义
print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

统计短信发送数据



```
<?php
require_once '/path/to/vendor/autoload.php';
// 导入对应产品模块的client
use TencentCloud\Sms\V20210111\SmsClient;
// 导入要请求接口对应的Request类
use TencentCloud\Sms\V20210111\Models\SendStatusStatisticsRequest;
use TencentCloud\Common\Exception\TencentCloudSDKException;
use TencentCloud\Common\Credential;
// 导入可选配置类
use TencentCloud\Common\Profile\ClientProfile;
use TencentCloud\Common\Profile\HttpProfile;
```

```
try {
    // 实例化一个证书对象, 入参需要传入腾讯云账户 SecretId, SecretKey
    // 为了保护密钥安全, 建议将密钥设置在环境变量中或者配置文件中。
    // 硬编码密钥到代码中有可能随代码泄露而暴露, 有安全隐患, 并不推荐。
    // SecretId、SecretKey 查询: https://console.intl.cloud.tencent.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项, 可选的, 没有特殊需求可以跳过
    $httpProfile = new HttpProfile();
    // 配置代理
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET"); // get请求 (默认为post请求)
    $httpProfile->setReqTimeout(30); // 请求超时时间, 单位为秒 (默认60秒)
    $httpProfile->setEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名 (默认就近)

    // 实例化一个client选项, 可选的, 没有特殊需求可以跳过
    $clientProfile = new ClientProfile();
    $clientProfile->setSignMethod("TC3-HMAC-SHA256"); // 指定签名算法
    $clientProfile->setHttpProfile($httpProfile);

    // 实例化要请求产品 (以sms为例) 的client对象, clientProfile是可选的
    // 第二个参数是地域信息, 根据您的选择的国际站地域, 如您选择的是新加坡国际站, 则应该填入字符串ap-
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

    // 实例化一个 sms 发送短信请求对象, 每个接口都会对应一个request对象。
    $req = new SendStatusStatisticsRequest();

    /* 填充请求参数, 这里request对象的成员变量即对应接口的入参
    * 你可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义
    * 基本类型的设置:
    * 帮助链接:
    * 短信控制台: https://console.intl.cloud.tencent.com/smsv2
    * sms helper: https://intl.cloud.tencent.com/document/product/382/3773 */

    /* 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2400006666
    $req->SmsSdkAppId = "2400006666";
    /* 拉取最大条数, 最多100条 */
    $req->Limit = 10;
    /* 偏移量 注: 目前固定设置为0 */
    $req->Offset = 0;
    /* 开始时间, yyyyymmddhh 需要拉取的起始时间, 精确到小时 */
    $req->BeginTime = "2019122500";
    /* 结束时间, yyyyymmddhh 需要拉取的截止时间, 精确到小时
    * 注: EndTime 必须大于 BeginTime */
```

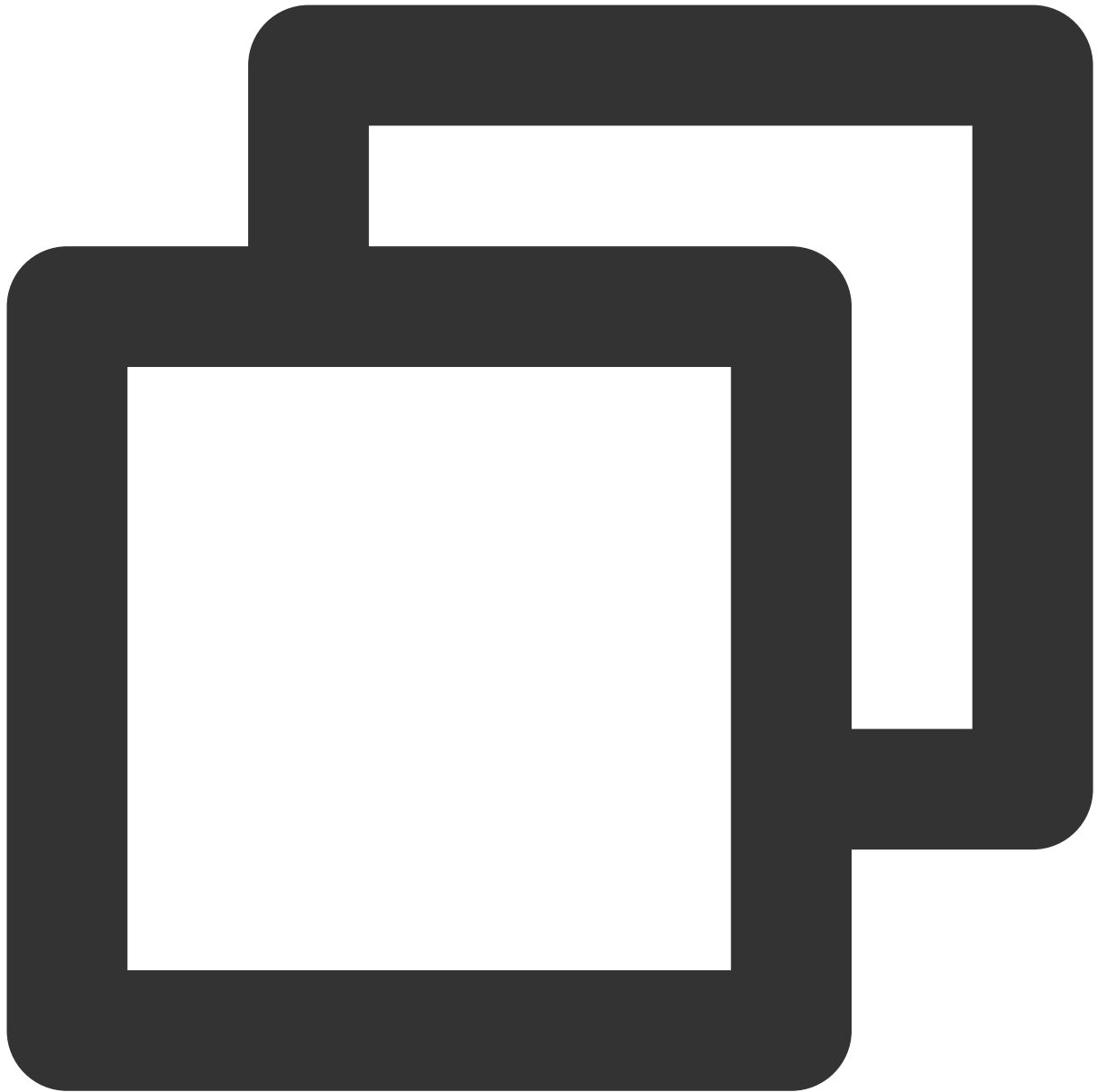
```
$req->EndTime = "2019122523";

// 通过client对象调用SendStatusStatistics方法发起请求。注意请求方法名与请求对象是对应的
// 返回的resp是一个SendStatusStatisticsResponse类的实例，与请求对象对应
$resp = $client->SendStatusStatistics($req);

// 输出json格式的字符串回包
print_r($resp->toJsonString());

// 也可以取出单个值。
// 你可以通过官网接口文档或跳转到response对象的定义处查看返回字段的定义
print_r($resp->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

申请短信模板



```
<?php
require_once '/path/to/vendor/autoload.php';
// 导入 SMS 模块的 client
use TencentCloud\\Sms\\V20210111\\SmsClient;
// 导入要请求接口对应的 Request 类
use TencentCloud\\Sms\\V20210111\\Models\\AddSmsTemplateRequest;
use TencentCloud\\Common\\Exception\\TencentCloudSDKException;
use TencentCloud\\Common\\Credential;
// 导入可选配置类
use TencentCloud\\Common\\Profile\\ClientProfile;
use TencentCloud\\Common\\Profile\\HttpProfile;
```

```
try {
    // 实例化一个证书对象，入参需要传入腾讯云账户 SecretId, SecretKey
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
    // $cred = new Credential("SecretId", "SecretKey");
    $cred = new Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                           getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个 http 选项，可选，无特殊需求时可以跳过
    $httpProfile = new HttpProfile();
    // 配置代理
    // $httpProfile->setProxy("https://ip:port");
    $httpProfile->setReqMethod("GET"); // get请求（默认为 POST 请求）
    $httpProfile->setReqTimeout(30); // 请求超时时间，单位为秒（默认60秒）
    $httpProfile->setEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名（默认就近

    // 实例化一个 client 选项，可选，无特殊需求时可以跳过
    $clientProfile = new ClientProfile();
    $clientProfile->setSignMethod("TC3-HMAC-SHA256"); // 指定签名算法
    $clientProfile->setHttpProfile($httpProfile);

    // 实例化 SMS 的 client 对象，clientProfile 是可选的
    // 第二个参数是地域信息，根据您的选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-
    $client = new SmsClient($cred, "ap-singapore", $clientProfile);

    // 实例化一个 AddSmsTemplateRequest 请求对象，每个接口都会对应一个 request 对象。
    $req = new AddSmsTemplateRequest();

    /* 填充请求参数，这里 request 对象的成员变量即对应接口的入参
     * 您可以通过官网接口文档或跳转到 request 对象的定义处查看请求参数的定义
     * 基本类型的设置：
     * 帮助链接：
     * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
     * sms helper：https://intl.cloud.tencent.com/document/product/382/3773
     */

    /* 模板名称 */
    $req->TemplateName = "腾讯云";
    /* 模板内容 */
    $req->TemplateContent = "{1}为您的登录验证码，请于{2}分钟内填写，如非本人操作，请忽略本短
    /* 短信类型，1表示营销短信，2表示通知短信，3表示验证码短信 */
    $req->SmsType = 3;
    /* 是否国际/港澳台短信：
    0表示中国大陆地区短信
    1表示国际/港澳台短信 */
    $req->International = 0;
```

```
/* 模板备注：例如申请原因，使用场景等 */
$req->Remark = "xxx";
// 通过 client 对象调用 AddSmsTemplate 方法发起请求。注意请求方法名与请求对象是对应的
$res = $client->AddSmsTemplate($req);
// 输出 JSON 格式的字符串回包
print_r($res->toJsonString());

// 可以取出单个值，您可以通过官网接口文档或跳转到 response 对象的定义处查看返回字段的定义
print_r($res->TotalCount);
}
catch(TencentCloudSDKException $e) {
    echo $e;
}
```

常见问题

证书问题

如果您的 PHP 环境证书有问题，可能会遇到报错，类似于 `cURL error 60: See`

`http://curl.haxx.se/libcurl/c/libcurl-errors.html`，请尝试按以下步骤解决：

1. 到 `https://curl.haxx.se/ca/cacert.pem` 下载证书文件 `cacert.pem`，将其保存到 PHP 安装路径下。
2. 编辑 `php.ini` 文件，删除 `curl.cainfo` 配置项前的分号注释符（;），值设置为保存的证书文件 `cacert.pem` 的绝对路径。
3. 重启依赖 PHP 的服务。

php_curl 扩展

此 SDK 依赖的 `GuzzleHttp` 需要开启 `php_curl` 扩展，查看环境上的 `php.ini` 环境确认是否已启用，例如在 Linux 环境下，PHP 7.1 版本，托管在 `apache` 下的服务，可以打开 `/etc/php/7.1/apache2/php.ini`，查看 `extension=php_curl.dll` 配置项是否已被注释，请删除此项配置前的注释符并重启 `apache`。

Web 访问异常

命令行下执行正常，但是放在 Web 服务器执行则报错：

```
cURL error 0: The cURL request was retried 3 times and did not succeed. The most likely reason for the failure is that cURL was unable to rewind the body of the request and subsequent retries resulted in the same error. Turn on the debug option to see what went wrong. See https://bugs.php.net/bug.php?id=47204 for more information. (see http://curl.haxx.se/libcurl/c/libcurl-errors.html)
```

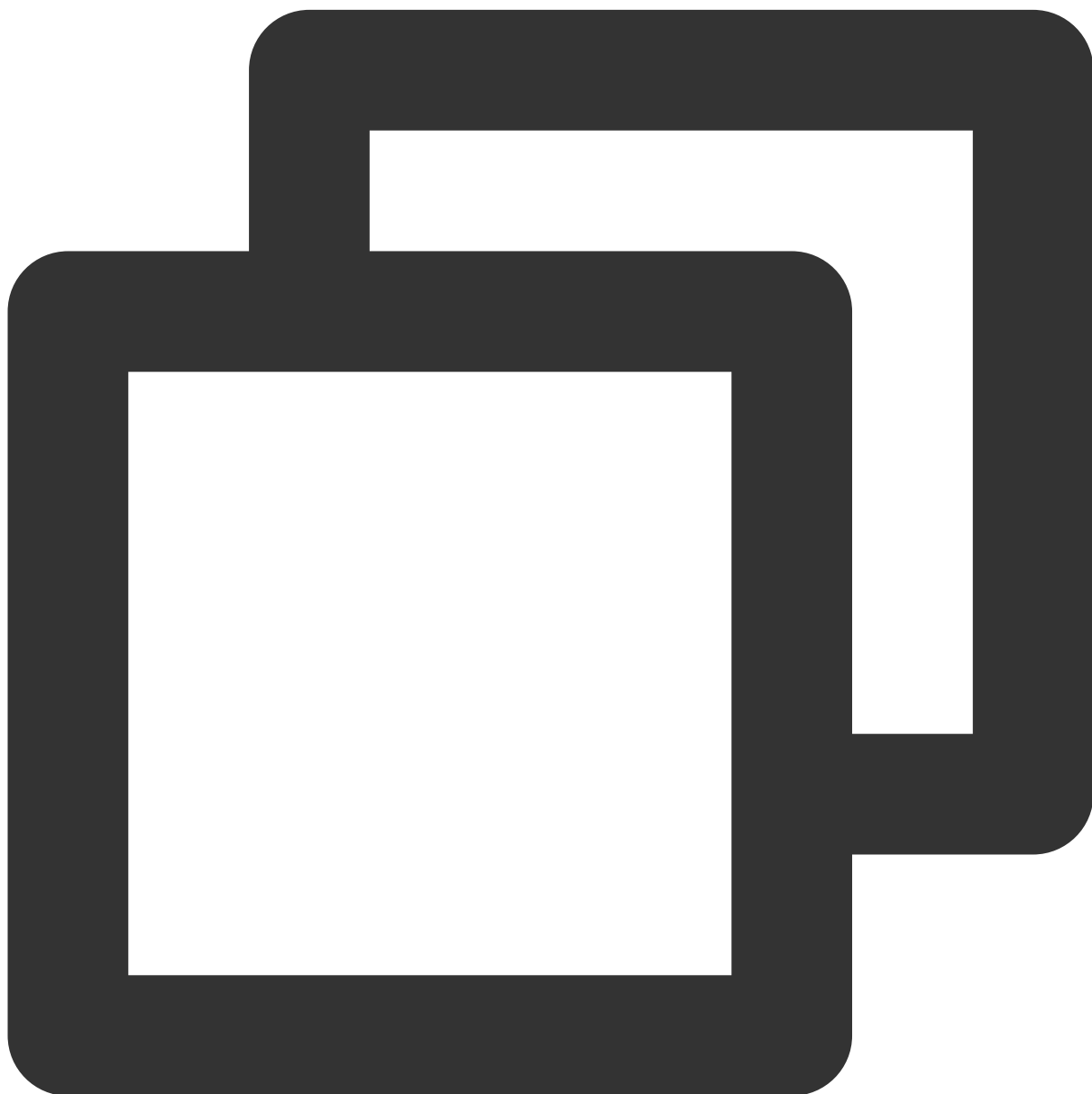
此问题出现情况不一。可以运行 `php -r "echo sys_get_temp_dir();"` ，打印系统默认临时目录绝对路径，然后在 `php.ini` 配置 `sys_temp_dir` 为这个值，尝试是否能解决。

源码安装问题

为了支持部分源码安装的需要，我们将依赖的包文件放在 `vendor` 目录中，又考虑到不能造成对 `composer` 的不兼容，`github` 不得不设置禁止导出 `vendor` 目录，造成必须使用 `git clone` 命令才能拿到 `vendor` 目录的情况，对一些不熟悉 `github` 的用户造成了困扰。从3.0.188版本开始，我们暂时移除了源码安装，必须使用 `composer` 安装 SDK 和依赖的包。

代理设置

在有代理的环境下，需要设置系统环境变量 `https_proxy`，否则可能无法正常调用，抛出连接超时的异常。或使用 `GuzzleHttp` 代理配置：



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

```
$httpProfile = new HttpProfile();  
$httpProfile->setProxy('https://ip:port');  
  
$clientProfile = new ClientProfile();  
$clientProfile->setHttpProfile($httpProfile);
```

Python SDK

最近更新时间：2024-06-27 15:48:00

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本。

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口。

一次群发请求最多支持200个号码。

签名、正文模板相关接口。

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

通过 pip 安装（推荐）

Github 仓库参见：[通过 pip 安装（推荐）](#)

Gitee 仓库参见：[通过 pip 安装（推荐）](#)

通过源码包安装

Github 仓库参见：[通过源码包安装](#)

Gitee 仓库参见：[通过源码包安装](#)

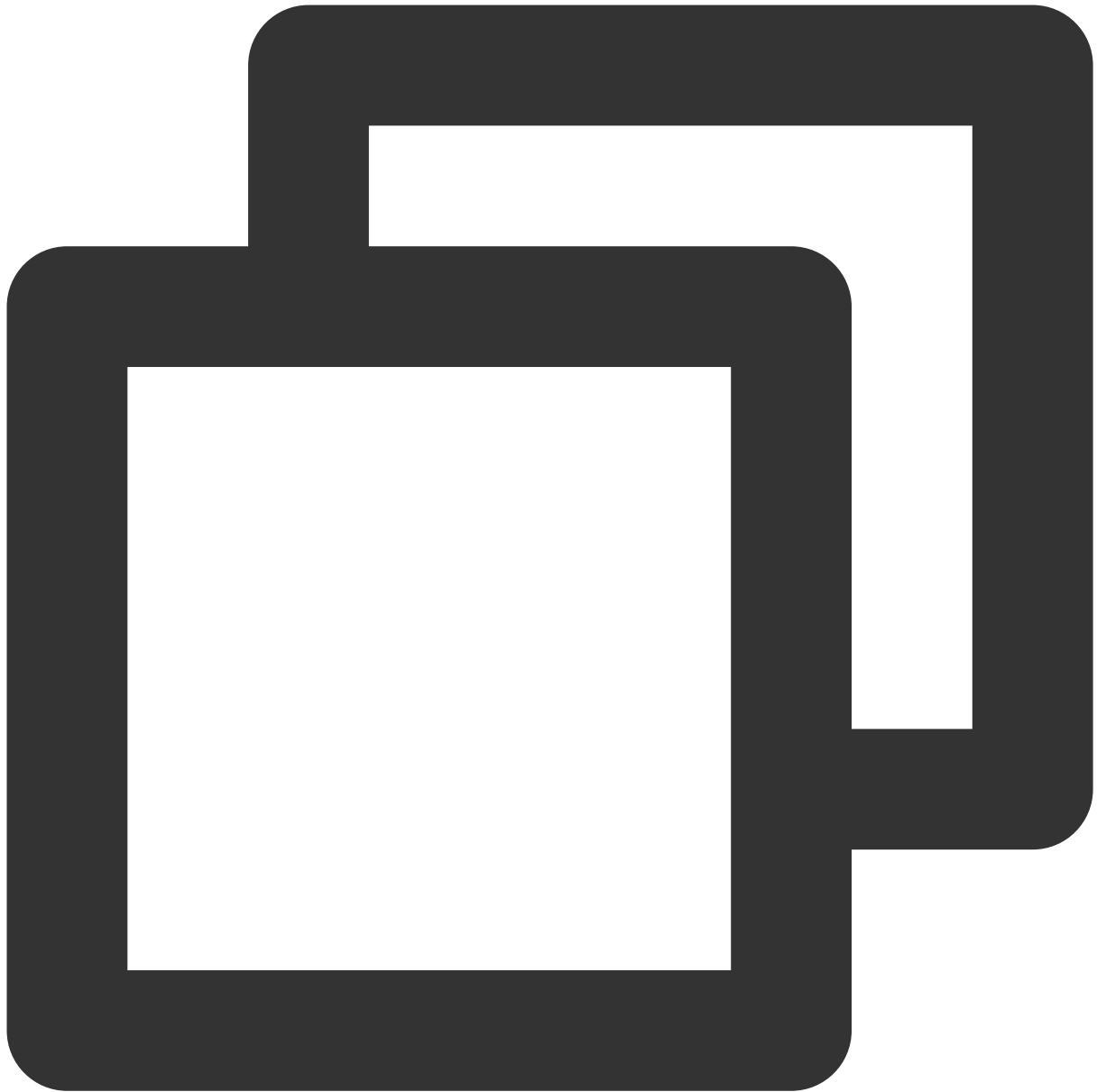
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# 导入对应产品模块的client models。
from tencentcloud.sms.v20210111 import sms_client, models

# 导入可选配置类
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
    # 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey。
```

```
# 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
# 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
# SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
# cred = credential.Credential("secretId", "secretKey")
cred = credential.Credential(
    os.environ.get("TENCENTCLOUD_SECRET_ID"),
    os.environ.get("TENCENTCLOUD_SECRET_KEY"))

# 实例化一个http选项，可选的，没有特殊需求可以跳过。
httpProfile = HttpProfile()
# 如果需要指定proxy访问接口，可以按照如下方式初始化hp
# httpProfile = HttpProfile(proxy="http://用户名:密码@代理IP:代理端口")
httpProfile.reqMethod = "POST" # post请求(默认为post请求)
httpProfile.reqTimeout = 10 # 请求超时时间，单位为秒(默认60秒)
httpProfile.endpoint = "sms.tencentcloudapi.com" # 指定接入地域域名(默认就近接入)

# 非必要步骤：
# 实例化一个客户端配置对象，可以指定超时时间等配置
clientProfile = ClientProfile()
clientProfile.signMethod = "TC3-HMAC-SHA256" # 指定签名算法
clientProfile.language = "en-US"
clientProfile.httpProfile = httpProfile

# 实例化要请求产品(以sms为例)的client对象
# 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-s
client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

# 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
# 你可以直接查询SDK源码确定SendSmsRequest有哪些属性可以设置
# 属性可能是基本类型，也可能引用了另一个数据结构
# 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明
req = models.SendSmsRequest()

# 基本类型的设置：
# SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
# SDK提供对基本类型的指针引用封装函数
# 帮助链接：
# 短信控制台：https://console.intl.cloud.tencent.com/smsv2
# sms helper：https://intl.cloud.tencent.com/document/product/382/3773

# 短信应用ID：短信SdkAppId在[短信控制台]添加应用后生成的实际SdkAppId，示例如2400006666
req.SmsSdkAppId = "2400006666"
# 短信签名内容：使用 UTF-8 编码，必须填写已审核通过的签名，签名信息可登录[短信控制台]查看
req.SignName = "xxx"
# 短信码号扩展号：默认未开通，如需开通请联系[sms helper]
req.ExtendCode = ""
# 用户的 session 内容：可以携带用户侧 ID 等上下文信息，server 会原样返回
```

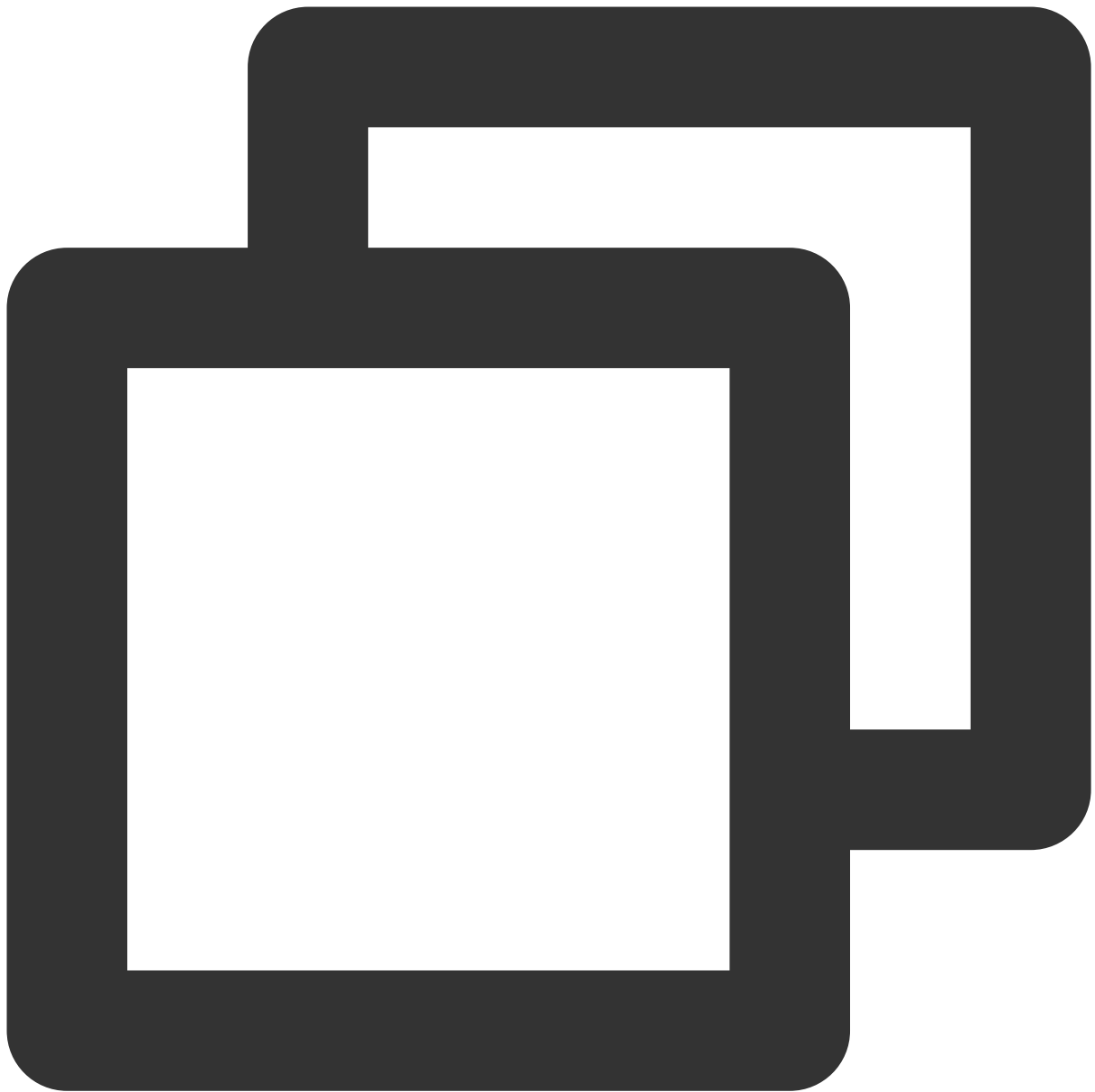
```
req.SessionContext = "xxx"
# 国际/港澳台短信 senderid: 中国大陆地区短信填空, 默认未开通, 如需开通请联系 [sms helper]
req.SenderId = ""
# 下发手机号码, 采用 E.164 标准, +[国家或地区码][手机号]
# 示例如: +60198890000, 其中前面有一个+号, 60为国家码, 198890000为手机号, 最多不要超过2
req.PhoneNumberSet = ["+60198890000"]
# 模板 ID: 必须填写已审核通过的模板 ID。模板ID可登录 [短信控制台] 查看
req.TemplateId = "449739"
# 模板参数: 若无模板参数, 则设置为空
req.TemplateParamSet = ["666"]

# 通过client对象调用DescribeInstances方法发起请求。注意请求方法名与请求对象是对应的。
# 返回的resp是一个DescribeInstancesResponse类的实例, 与请求对象对应。
resp = client.SendSms(req)

# 输出json格式的字符串回包
print(resp.to_json_string(indent=2))

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

拉取短信下发状态



```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# 导入对应产品模块的client models。
from tencentcloud.sms.v20210111 import sms_client, models

# 导入可选配置类
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
    # 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey。
```

```
# 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
# 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
# SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
# cred = credential.Credential("secretId", "secretKey")
cred = credential.Credential(
    os.environ.get("TENCENTCLOUD_SECRET_ID"),
    os.environ.get("TENCENTCLOUD_SECRET_KEY"))

# 实例化一个http选项，可选的，没有特殊需求可以跳过。
httpProfile = HttpProfile()
# 如果需要指定proxy访问接口，可以按照如下方式初始化hp
# httpProfile = HttpProfile(proxy="http://用户名:密码@代理IP:代理端口")
httpProfile.reqMethod = "POST" # post请求(默认为post请求)
httpProfile.reqTimeout = 30 # 请求超时时间，单位为秒(默认60秒)
httpProfile.endpoint = "sms.tencentcloudapi.com" # 指定接入地域域名(默认就近接入)

# 非必要步骤：
# 实例化一个客户端配置对象，可以指定超时时间等配置
clientProfile = ClientProfile()
clientProfile.signMethod = "TC3-HMAC-SHA256" # 指定签名算法
clientProfile.language = "en-US"
clientProfile.httpProfile = httpProfile

# 实例化要请求产品(以sms为例)的client对象
# 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-s
client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

# 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
# 你可以直接查询SDK源码确定SendSmsRequest有哪些属性可以设置
# 属性可能是基本类型，也可能引用了另一个数据结构
# 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明
req = models.PullSmsSendStatusRequest()

# 基本类型的设置：
# SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
# SDK提供对基本类型的指针引用封装函数
# 帮助链接：
# 短信控制台：https://console.intl.cloud.tencent.com/smsv2
# sms helper：https://intl.cloud.tencent.com/document/product/382/3773

# 短信应用ID：短信SdkAppId在[短信控制台]添加应用后生成的实际SdkAppId，示例如2400006666
req.SmsSdkAppId = "2400006666"
# 拉取最大条数，最多100条
req.Limit = 10

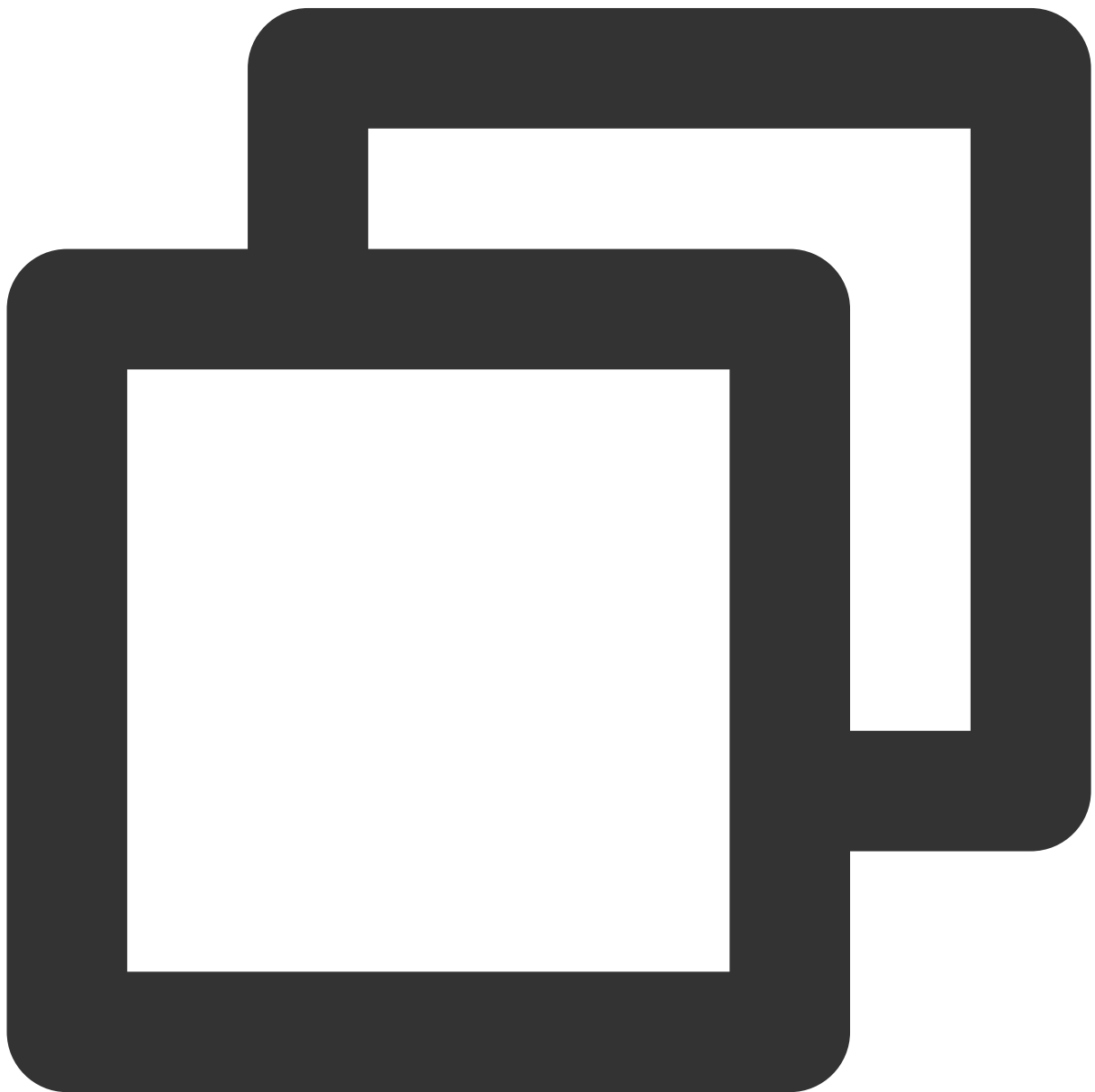
# 通过client对象调用PullSmsSendStatus方法发起请求。注意请求方法名与请求对象是对应的。
# 返回的resp是一个PullSmsSendStatusResponse类的实例，与请求对象对应。
```

```
resp = client.PullSmsSendStatus(req)

# 输出json格式的字符串回包
print(resp.to_json_string(indent=2))

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

统计短信发送数据



```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# 导入对应产品模块的client models。
from tencentcloud.sms.v20210111 import sms_client, models

# 导入可选配置类
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
    # 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey。
    # 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    # 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    # SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
    # cred = credential.Credential("secretId", "secretKey")
    cred = credential.Credential(
        os.environ.get("TENCENTCLOUD_SECRET_ID"),
        os.environ.get("TENCENTCLOUD_SECRET_KEY"))

    # 实例化一个http选项，可选的，没有特殊需求可以跳过。
    httpProfile = HttpProfile()
    # 如果需要指定proxy访问接口，可以按照如下方式初始化hp
    # httpProfile = HttpProfile(proxy="http://用户名:密码@代理IP:代理端口")
    httpProfile.reqMethod = "POST" # post请求(默认为post请求)
    httpProfile.reqTimeout = 30 # 请求超时时间，单位为秒(默认60秒)
    httpProfile.endpoint = "sms.tencentcloudapi.com" # 指定接入地域域名(默认就近接入)

    # 非必要步骤：
    # 实例化一个客户端配置对象，可以指定超时时间等配置
    clientProfile = ClientProfile()
    clientProfile.signMethod = "TC3-HMAC-SHA256" # 指定签名算法
    clientProfile.language = "en-US"
    clientProfile.httpProfile = httpProfile

    # 实例化要请求产品(以sms为例)的client对象
    # 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-s
    client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

    # 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
    # 你可以直接查询SDK源码确定SendSmsRequest有哪些属性可以设置
    # 属性可能是基本类型，也可能引用了另一个数据结构
    # 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明
    req = models.SendStatusStatisticsRequest()

    # 基本类型的设置：
    # SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
    # SDK提供对基本类型的指针引用封装函数
```

```
# 帮助链接：
# 短信控制台：https://consoleintl.cloud.tencent.com/smsv2
# sms helper：https://intl.cloud.tencent.com/document/product/382/3773

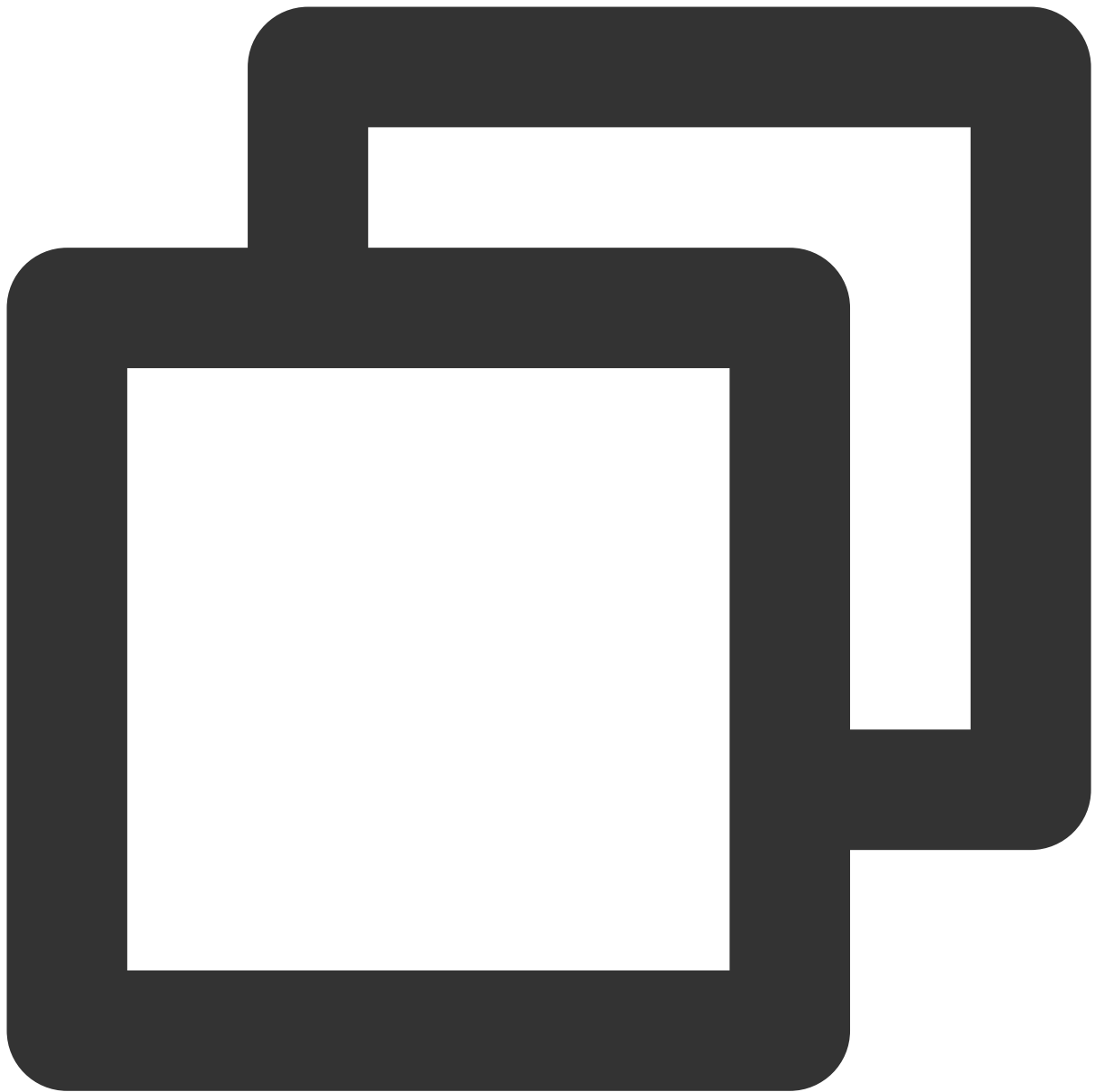
# 短信应用ID：短信SmsSdkAppId在 [短信控制台] 添加应用后生成的实际SmsSdkAppId，示例如2400
req.SmsSdkAppId = "2400006666"
# 拉取最大条数，最多100条
req.Limit = 10
# 偏移量 注：目前固定设置为0
req.Offset = 0
# 开始时间，yyyymmddhh 需要拉取的起始时间，精确到小时
req.BeginTime = "2019122400"
# 结束时间，yyyymmddhh 需要拉取的截止时间，精确到小时
# 注：EndTime 必须大于 BeginTime
req.EndTime = "2019122523"

# 通过client对象调用SendStatusStatistics方法发起请求。注意请求方法名与请求对象是对应的。
# 返回的resp是一个SendStatusStatisticsResponse类的实例，与请求对象对应。
resp = client.SendStatusStatistics(req)

# 输出json格式的字符串回包
print(resp.to_json_string(indent=2))

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

申请短信模板



```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# 导入对应产品模块的client models。
from tencentcloud.sms.v20210111 import sms_client, models

# 导入可选配置类
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
try:
    # 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey。
```

```
# 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
# 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
# SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
# cred = credential.Credential("secretId", "secretKey")
cred = credential.Credential(
    os.environ.get("TENCENTCLOUD_SECRET_ID"),
    os.environ.get("TENCENTCLOUD_SECRET_KEY"))

# 实例化一个 http 选项，可选，无特殊需求时可以跳过
httpProfile = HttpProfile()
# 如果需要指定proxy访问接口，可以按照如下方式初始化hp
# httpProfile = HttpProfile(proxy="http://用户名:密码@代理IP:代理端口")
httpProfile.reqMethod = "POST" # POST 请求（默认为 POST 请求）
httpProfile.reqTimeout = 30 # 请求超时时间，单位为秒（默认60秒）
httpProfile.endpoint = "sms.tencentcloudapi.com" # 指定接入地域域名（默认就近接入）

# 非必要步骤：
# 实例化一个客户端配置对象，可以指定超时时间等配置
clientProfile = ClientProfile()
clientProfile.signMethod = "TC3-HMAC-SHA256" # 指定签名算法
clientProfile.language = "en-US"
clientProfile.httpProfile = httpProfile

# 实例化 SMS 的 client 对象
# 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-s
client = sms_client.SmsClient(cred, "ap-singapore", clientProfile)

# 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
# 您可以直接查询 SDK 源码确定 AddSmsTemplateRequest 有哪些属性可以设置
# 属性可能是基本类型，也可能引用了另一个数据结构
# 推荐使用 IDE 进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明
req = models.AddSmsTemplateRequest()

# 基本类型的设置：
# SDK 采用的是指针风格指定参数，即使对于基本类型也需要用指针来对参数赋值
# SDK 提供对基本类型的指针引用封装函数
# 帮助链接：
# 短信控制台：https://console.intl.cloud.tencent.com/smsv2
# sms helper：https://intl.cloud.tencent.com/document/product/382/3773

# 模板名称
req.TemplateName = "腾讯云"
# 模板内容
req.TemplateContent = "{1}为您的登录验证码，请于{2}分钟内填写，如非本人操作，请忽略本短信"
# 短信类型，1表示营销短信，2表示通知短信，3表示验证码短信
req.SmsType = 3
# 是否国际/港澳台短信：
```

```
# 0：表示中国大陆地区短信
# 1：表示国际/港澳台短信
req.International = 0
# 模板备注：例如申请原因，使用场景等
req.Remark = "xxx"

# 通过 client 对象调用 AddSmsTemplate 方法发起请求。注意请求方法名与请求对象是对应的
# 返回的resp是一个 AddSmsTemplateResponse 类的实例，与请求对象对应。
resp = client.AddSmsTemplate(req)

# 输出 JSON 格式的字符串回包
print(resp.to_json_string(indent=2))

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

常见问题

证书问题

在 Mac 操作系统安装 Python 3.6 或以上版本时，可能会遇到证书错误：`Error: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1056)`。

这是因为在 Mac 操作系统下，Python 不再使用系统默认的证书，且本身也不提供证书。在进行 HTTPS 请求时，需要使用 `certifi` 库提供的证书，但 SDK 不支持指定，所以只能使用 `sudo "/Applications/Python 3.6/Install Certificates.command"` 命令安装证书才能解决此问题。

虽然 Python 2 版本不应该有上述问题，但在个别用户环境上可能也会存在类似的情况，同样可以通过 `sudo "/Applications/Python 2.7/Install Certificates.command"` 解决。

代理设置

如果有代理的环境下，可通过以下两种方式设置代理：

在初始化 `HttpProfile` 时指定 `proxy`，参考 [example](#)。

需要设置系统环境变量 `https_proxy`。

否则可能无法正常调用，抛出连接超时的异常。

Node.js SDK

最近更新时间：2024-06-27 15:48:00

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本。

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口。

一次群发请求最多支持200个号码。

签名、正文模板相关接口。

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口，请将“个人认证”变更为“企业认证”。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

通过 npm 安装（推荐）

Github 仓库参见：[通过 npm 安装](#)

Gitee 仓库参见：[通过 npm 安装](#)

通过源码包安装

Github 仓库参见：[通过源码包安装](#)

Gitee 仓库参见：[通过源码包安装](#)

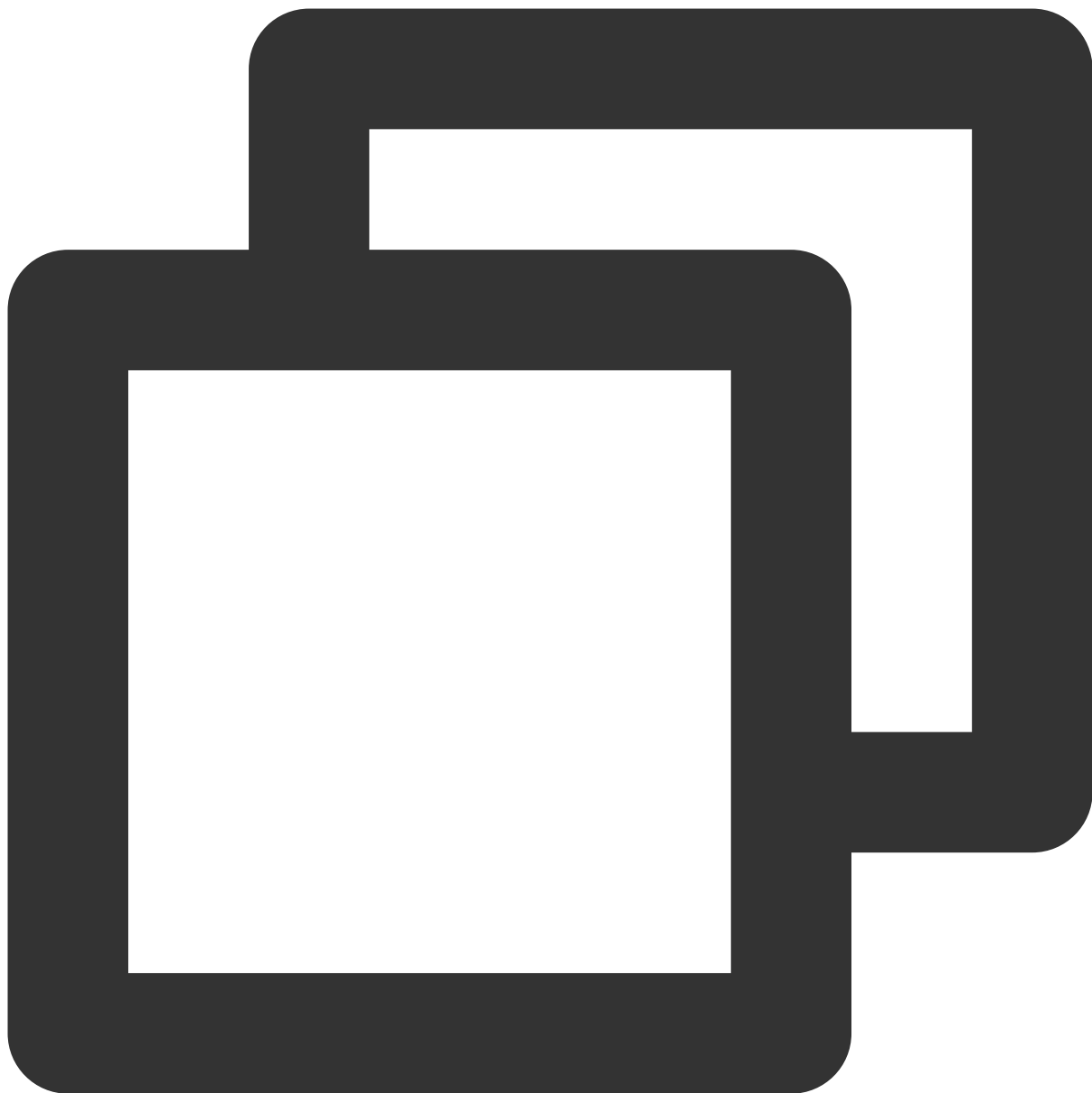
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



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

// 导入对应产品模块的client models。
const smsClient = tencentcloud.sms.v20210111.Client

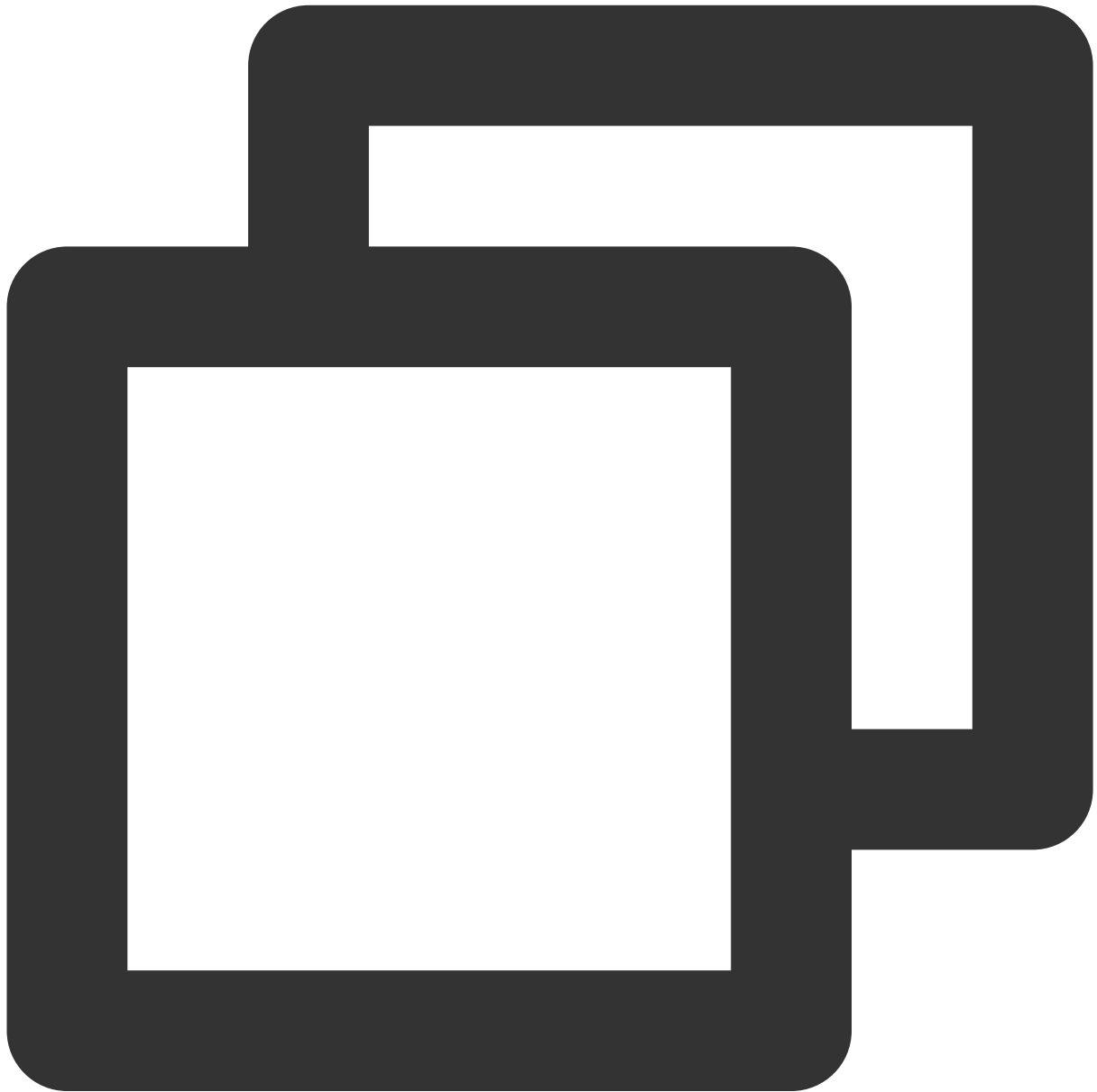
/* 实例化要请求产品(以sms为例)的client对象 */
const client = new smsClient({
  credential: {
    /* 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    * 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    * CAM密匙查询：https://console.intl.cloud.tencent.com/cam/capi */
```

```
secretId: process.env.TENCENTCLOUD_SECRET_ID,
secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
},
/* 必填：地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-singapore
region: "ap-singapore",
/* 非必填：
* 客户端配置对象，可以指定超时时间等配置 */
profile: {
  /* SDK默认用TC3-HMAC-SHA256进行签名，非必要请不要修改这个字段 */
  signMethod: "HmacSHA256",
  httpProfile: {
    reqMethod: "POST", // 请求方法
    reqTimeout: 10, // 请求超时时间，默认60s
    endpoint: "sms.tencentcloudapi.com"
  },
},
},
})

/* 请求参数，根据调用的接口和实际情况，可以进一步设置请求参数
* 属性可能是基本类型，也可能引用了另一个数据结构
* 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
const params = {
  /* 短信应用ID：短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId，示例如2400006666
  SmsSdkAppId: "2400006666",
  /* 短信签名内容：使用 UTF-8 编码，必须填写已审核通过的签名，签名信息可登录 [短信控制台] 查看 */
  SignName: "xxx",
  /* 短信码号扩展号：默认未开通，如需开通请联系 [sms helper] */
  ExtendCode: "",
  /* 国际/港澳台短信 senderid: 中国大陆地区短信填空，默认未开通，如需开通请联系 [sms helper] */
  SenderId: "",
  /* 用户的 session 内容：可以携带用户侧 ID 等上下文信息，server 会原样返回 */
  SessionContext: "",
  /* 下发手机号码，采用 e.164 标准，+[国家或地区码][手机号]
  * 示例如：+60198890000， 其中前面有一个+号，60为国家码，198890000为手机号，最多不要超过20
  PhoneNumberSet: ["+60198890000"],
  /* 模板 ID：必须填写已审核通过的模板 ID。模板ID可登录 [短信控制台] 查看 */
  TemplateId: "449739",
  /* 模板参数：若无模板参数，则设置为空*/
  TemplateParamSet: ["666"],
}
// 通过client对象调用想要访问的接口，需要传入请求对象以及响应回调函数
client.SendSms(params, function (err, response) {
  // 请求异常返回，打印异常信息
  if (err) {
    console.log(err)
    return
  }
})
```

```
// 请求正常返回，打印response对象  
console.log(response)  
})
```

拉取回执状态

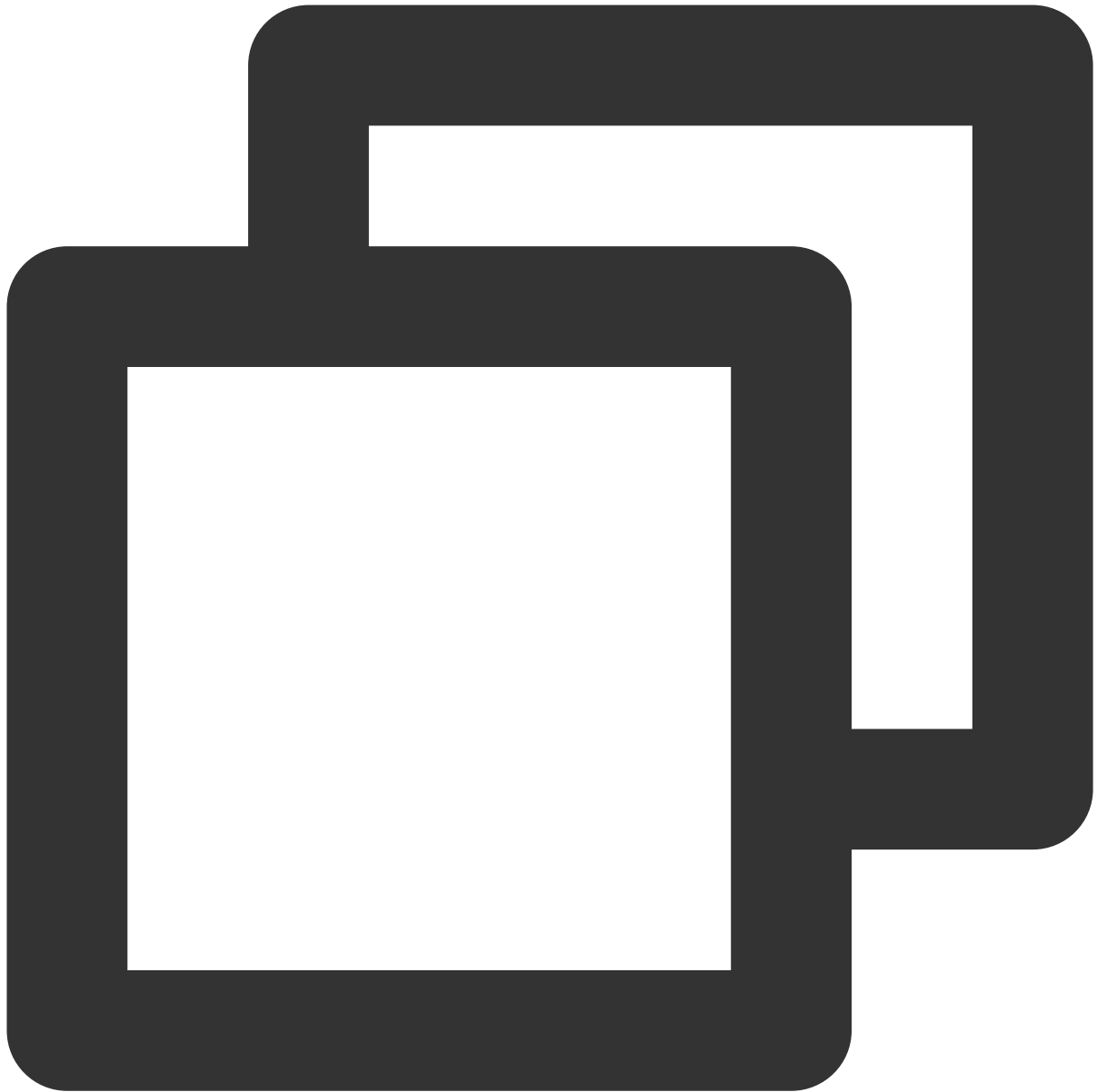


```
const tencentcloud = require("tencentcloud-sdk-nodejs")  
  
// 导入对应产品模块的client models。  
const smsClient = tencentcloud.sms.v20210111.Client
```

```
/* 实例化要请求产品(以sms为例)的client对象 */
const client = new smsClient({
  credential: {
    /* 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    * 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    * CAM密钥查询: https://console.intl.cloud.tencent.com/cam/capi */
    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  /* 必填：地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-singapore
  region: "ap-singapore",
  /* 非必填：
  * 客户端配置对象，可以指定超时时间等配置 */
  profile: {
    /* SDK默认用TC3-HMAC-SHA256进行签名，非必要请不要修改这个字段 */
    signMethod: "HmacSHA256",
    httpProfile: {
      reqMethod: "POST", // 请求方法
      reqTimeout: 30, // 请求超时时间，默认60s
      endpoint: "sms.tencentcloudapi.com"
    },
  },
},
})

/* 请求参数，根据调用的接口和实际情况，可以进一步设置请求参数
* 属性可能是基本类型，也可能引用了另一个数据结构
* 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
const params = {
  // 短信应用ID: 短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId, 示例如2400006666
  SdkAppId: "2400006666",
  // 拉取最大条数，最多100条
  Limit: 10,
}
// 通过client对象调用想要访问的接口，需要传入请求对象以及响应回调函数
client.PullSmsSendStatus(params, function (err, response) {
  // 请求异常返回，打印异常信息
  if (err) {
    console.log(err)
    return
  }
  // 请求正常返回，打印response对象
  console.log(response)
})
```

统计短信发送数据



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

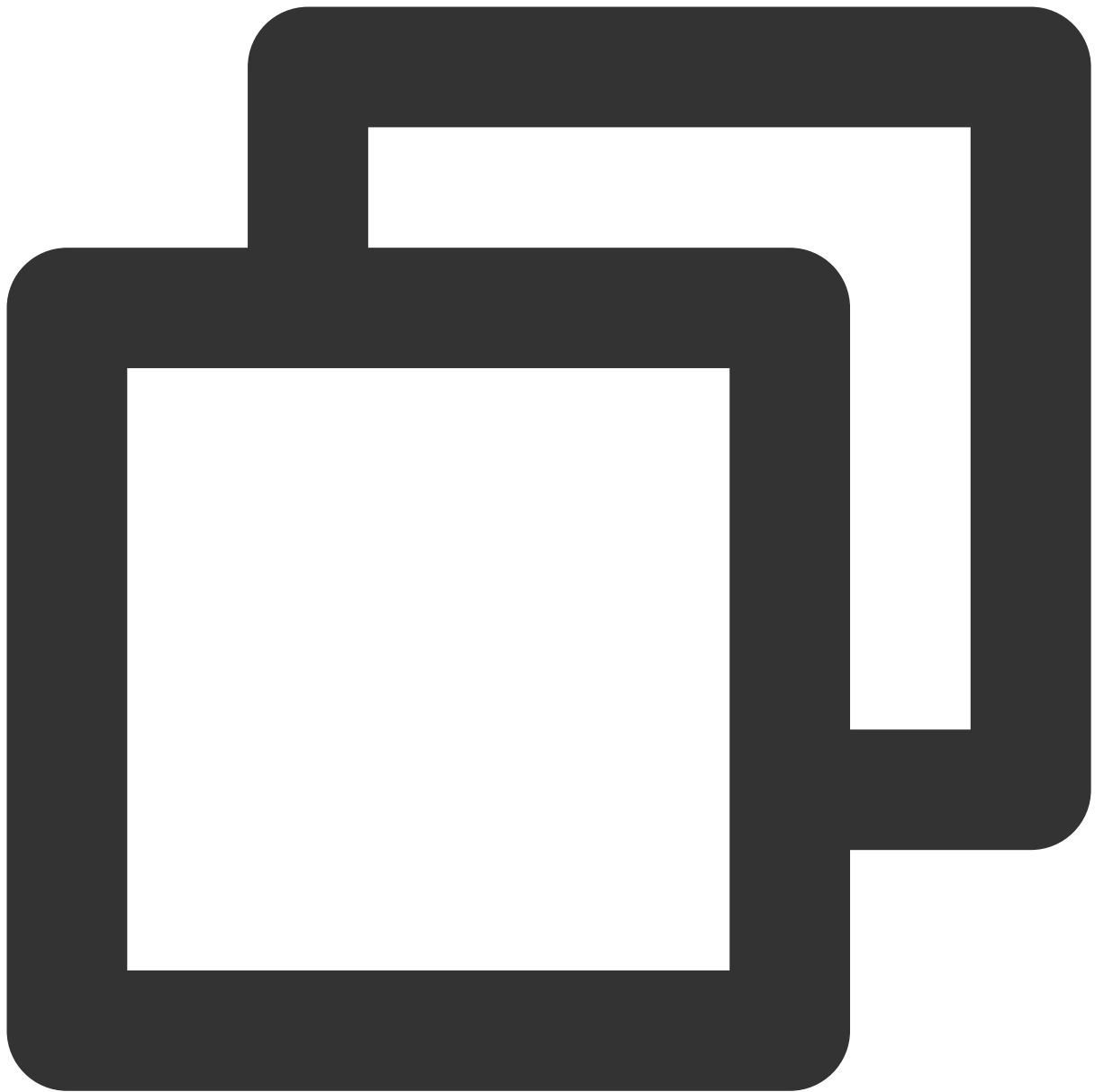
// 导入对应产品模块的client models。
const smsClient = tencentcloud.sms.v20210111.Client

/* 实例化要请求产品(以sms为例)的client对象 */
const client = new smsClient({
  credential: {
    /* 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    * 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    * CAM密匙查询：https://console.intl.cloud.tencent.com/cam/capi */
```

```
secretId: process.env.TENCENTCLOUD_SECRET_ID,
secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
},
/* 必填：地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-singapore
region: "ap-singapore",
/* 非必填：
* 客户端配置对象，可以指定超时时间等配置 */
profile: {
  /* SDK默认用TC3-HMAC-SHA256进行签名，非必要请不要修改这个字段 */
  signMethod: "HmacSHA256",
  httpProfile: {
    reqMethod: "POST", // 请求方法
    reqTimeout: 30, // 请求超时时间，默认60s
    endpoint: "sms.tencentcloudapi.com"
  },
},
})

/* 请求参数，根据调用的接口和实际情况，可以进一步设置请求参数
* 属性可能是基本类型，也可能引用了另一个数据结构
* 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
const params = {
  // 短信应用ID：短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId，示例如2400006666
  SmsSdkAppId: "2400006666",
  // 拉取最大条数，最多100条
  Limit: 10,
  // 偏移量 注：目前固定设置为0
  Offset: 0,
  // 开始时间，yyyymmddhh 需要拉取的起始时间，精确到小时
  BeginTime: "2019122400",
  // 结束时间，yyyymmddhh 需要拉取的截止时间，精确到小时
  // 注：EndTime 必须大于 BeginTime
  EndTime: "2019122523",
}
// 通过client对象调用想要访问的接口，需要传入请求对象以及响应回调函数
client.SendStatusStatistics(params, function (err, response) {
  // 请求异常返回，打印异常信息
  if (err) {
    console.log(err)
    return
  }
  // 请求正常返回，打印response对象
  console.log(response)
})
```

申请短信模板



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

// 导入对应产品模块的client models。
const smsClient = tencentcloud.sms.v20210111.Client

/* 实例化要请求产品(以sms为例)的client对象 */
const client = new smsClient({
  credential: {
    /* 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    * 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    * CAM密匙查询：https://console.intl.cloud.tencent.com/cam/capi */
```

```

    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  /* 必填：地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-singapore
  region: "ap-singapore",
  /* 非必填：
  * 客户端配置对象，可以指定超时时间等配置 */
  profile: {
    /* SDK默认用TC3-HMAC-SHA256进行签名，非必要请不要修改这个字段 */
    signMethod: "HmacSHA256",
    httpProfile: {
      reqMethod: "POST", // 请求方法
      reqTimeout: 30, // 请求超时时间，默认60s
      endpoint: "sms.tencentcloudapi.com"
    },
  },
},
}))

/* 请求参数，根据调用的接口和实际情况，可以进一步设置请求参数
* 属性可能是基本类型，也可能引用了另一个数据结构
* 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
const params = {
  /* 模板名称 */
  TemplateName: "腾讯云",
  /* 模板内容 */
  TemplateContent: "{1}为您的登录验证码，请于{2}分钟内填写，如非本人操作，请忽略本短信。",
  /* 短信类型，1表示营销短信，2表示通知短信，3表示验证码短信 */
  SmsType: 3,
  /* 是否国际/港澳台短信：0：表示中国大陆地区短信；1：表示国际/港澳台短信 */
  International: 0,
  /* 模板备注：例如申请原因，使用场景等 */
  Remark: "xxx",
}
// 通过client对象调用想要访问的接口，需要传入请求对象以及响应回调函数
client.AddSmsTemplate(params, function (err, response) {
  // 请求异常返回，打印异常信息
  if (err) {
    console.log(err)
    return
  }
  // 请求正常返回，打印response对象
  console.log(response)
})

```

常见问题

代理设置

如有代理的环境下，需要设置系统环境变量 `https_proxy`，否则可能无法正常调用，抛出连接超时的异常现象。

C# SDK

最近更新时间：2024-06-27 15:48:00

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本。

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口。

一次群发请求最多支持200个号码。

签名、正文模板相关接口。

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口，请将“个人认证”变更为“企业认证”。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

通过 nuget 安装（推荐）

Github 仓库参见：[通过 nuget 安装（推荐）](#)

Gitee 仓库参见：[通过 nuget 安装（推荐）](#)

通过源码安装

Github 仓库参见：[通过源码安装](#)

Gitee 仓库参见：[通过源码安装](#)

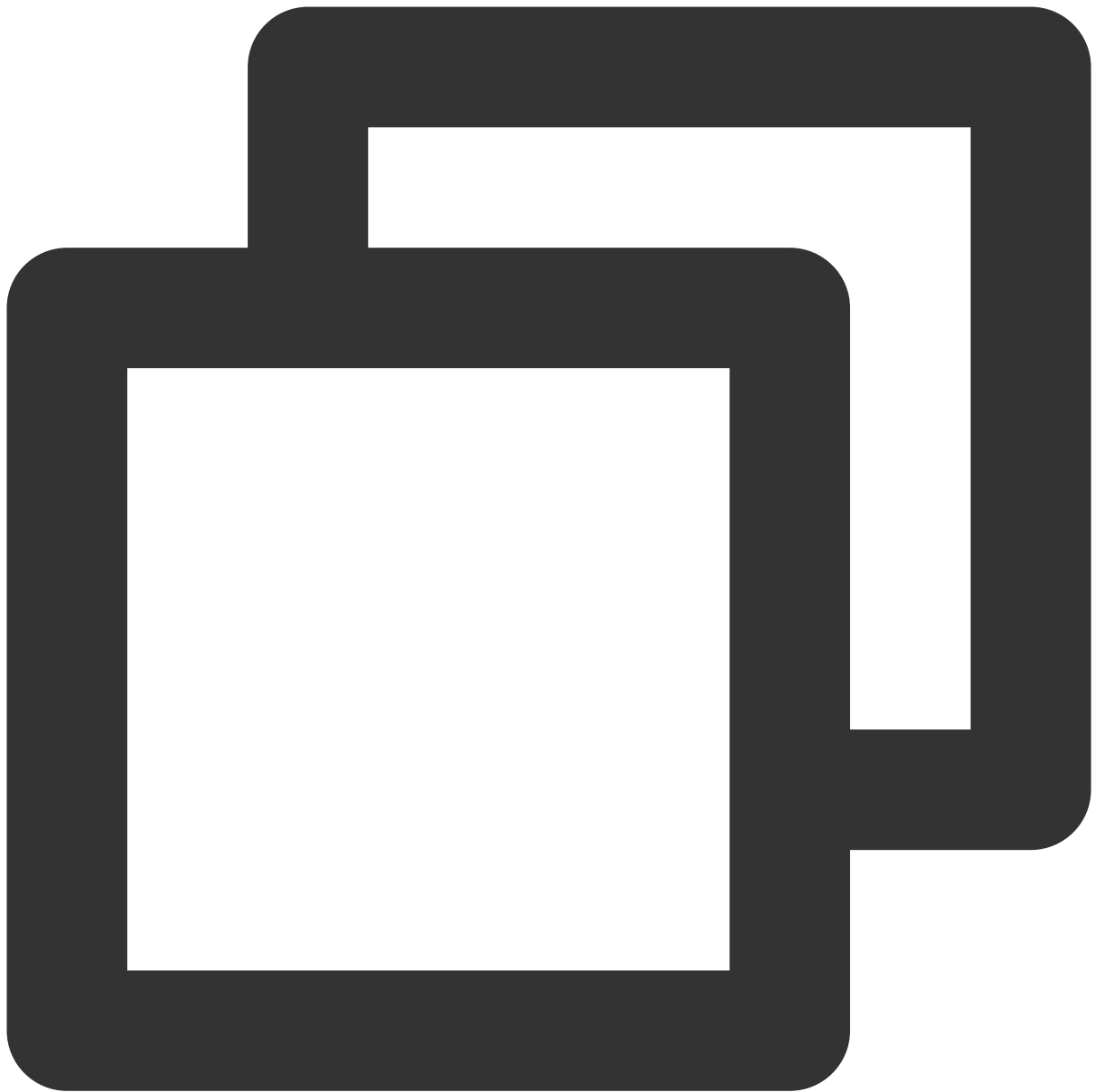
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



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

namespace TencentCloudExamples
{
    class SendSms
    {

```

```
static void Main(string[] args)
{
    try
    {
        // 必要步骤：
        // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
        // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
        // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
        // 这里采用的是从环境变量读取的方式，需要在环境变量中先设置这两个值。
        // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/credentials
        Credential cred = new Credential {
            SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID"),
            SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");
        };

        /* 非必要步骤：
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        ClientProfile clientProfile = new ClientProfile();
        /* SDK默认用TC3-HMAC-SHA256进行签名
        * 非必要请不要修改这个字段 */
        clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
        /* 非必要步骤
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        HttpProfile httpProfile = new HttpProfile();
        /* SDK默认使用POST方法。
        * 如果你一定要使用GET方法，可以在这里设置。GET方法无法处理一些较大的请求 */
        httpProfile.ReqMethod = "GET";
        httpProfile.Timeout = 10; // 请求连接超时时间，单位为秒(默认60秒)
        /* SDK会自动指定域名。通常是不需要特地指定域名的，但是如果你访问的是金融区的服务
        * 则必须手动指定域名，例如sms的上海金融区域名： sms.ap-shanghai-fsi.tencentcloudapi.com
        httpProfile.Endpoint = "sms.tencentcloudapi.com";
        // 代理服务器，当你的环境下有代理服务器时设定
        // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_PROXY");

        clientProfile.HttpProfile = httpProfile;
        /* 实例化要请求产品(以sms为例)的client对象
        * 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该传入ap-singapore
        SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile);

        /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
        * 你可以直接查询SDK源码确定SendSmsRequest有哪些属性可以设置
        * 属性可能是基本类型，也可能引用了另一个数据结构
        * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
        SendSmsRequest req = new SendSmsRequest();

        /* 基本类型的设置：
        * SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
```

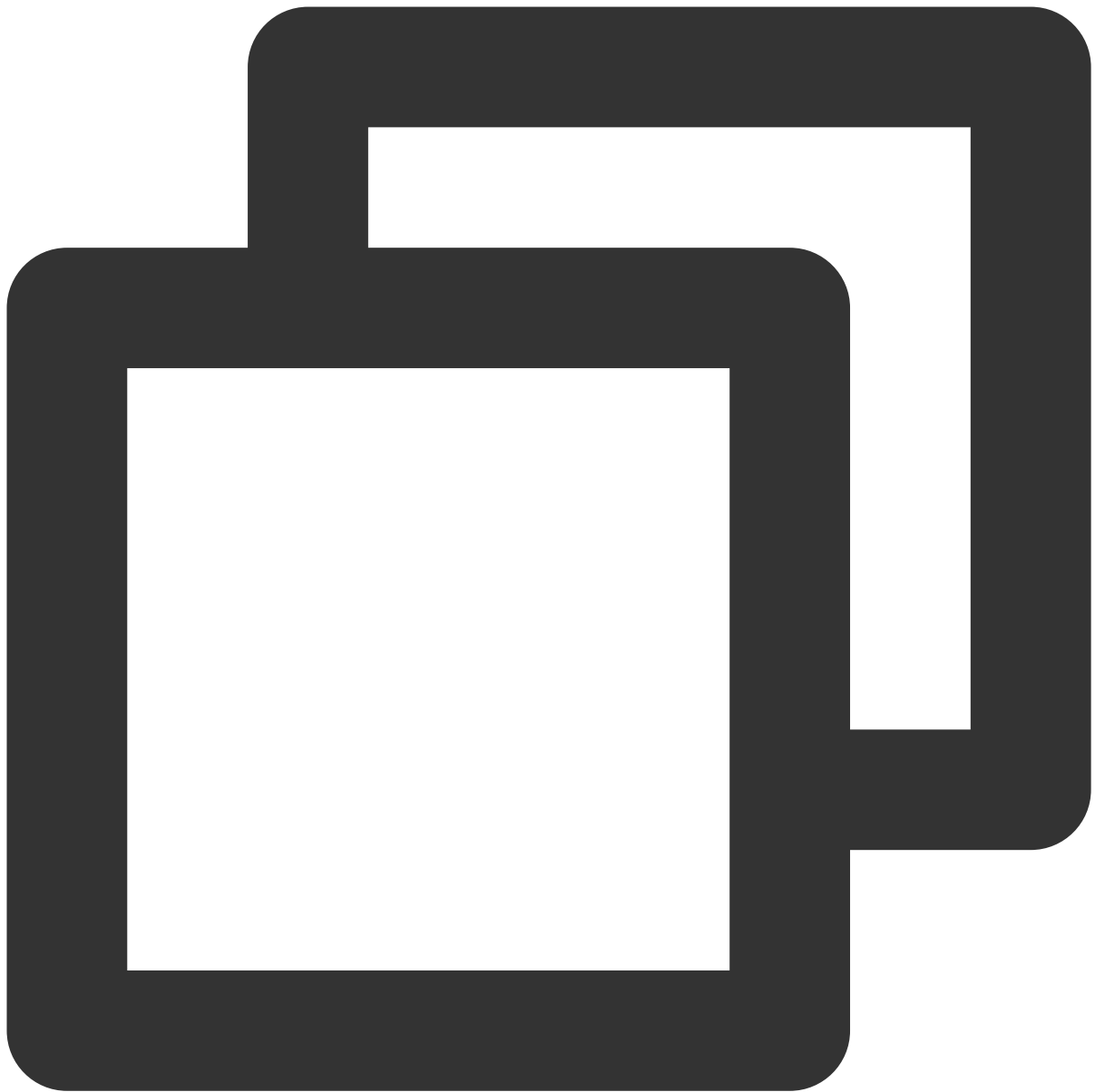
```
* SDK提供对基本类型的指针引用封装函数
* 帮助链接：
* 短信控制台：https://console.intl.cloud.tencent.com/smsv2
* sms helper：https://intl.cloud.tencent.com/document/product/382/

req.SmsSdkAppId = "2400006666";
/* 短信签名内容：使用 UTF-8 编码，必须填写已审核通过的签名，签名信息可登录 [短
req.SignName = "xxx";
/* 短信码号扩展号：默认未开通，如需开通请联系 [sms helper] */
req.ExtendCode = "";
/* 国际/港澳台短信 senderid：中国大陆地区短信填空，默认未开通，如需开通请联系
req.SenderId = "";
/* 用户的 session 内容：可以携带用户侧 ID 等上下文信息，server 会原样返回 *
req.SessionContext = "";
/* 下发手机号码，采用 E.164 标准，+[国家或地区码][手机号]
* 示例如：+60198890000，其中前面有一个+号，60为国家码，198890000为手机号
req.PhoneNumberSet = new String[] { "+60198890000" };
/* 模板 ID：必须填写已审核通过的模板 ID。模板ID可登录 [短信控制台] 查看 */
req.TemplateId = "449739";
/* 模板参数：若无模板参数，则设置为空*/
req.TemplateParamSet = new String[] { "666" };

// 通过client对象调用DescribeInstances方法发起请求。注意请求方法名与请求对象
// 返回的resp是一个DescribeInstancesResponse类的实例，与请求对象对应
SendSmsResponse resp = client.SendSmsSync(req);

// 输出json格式的字符串回包
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
Console.Read();
}
}
```

拉取回执状态



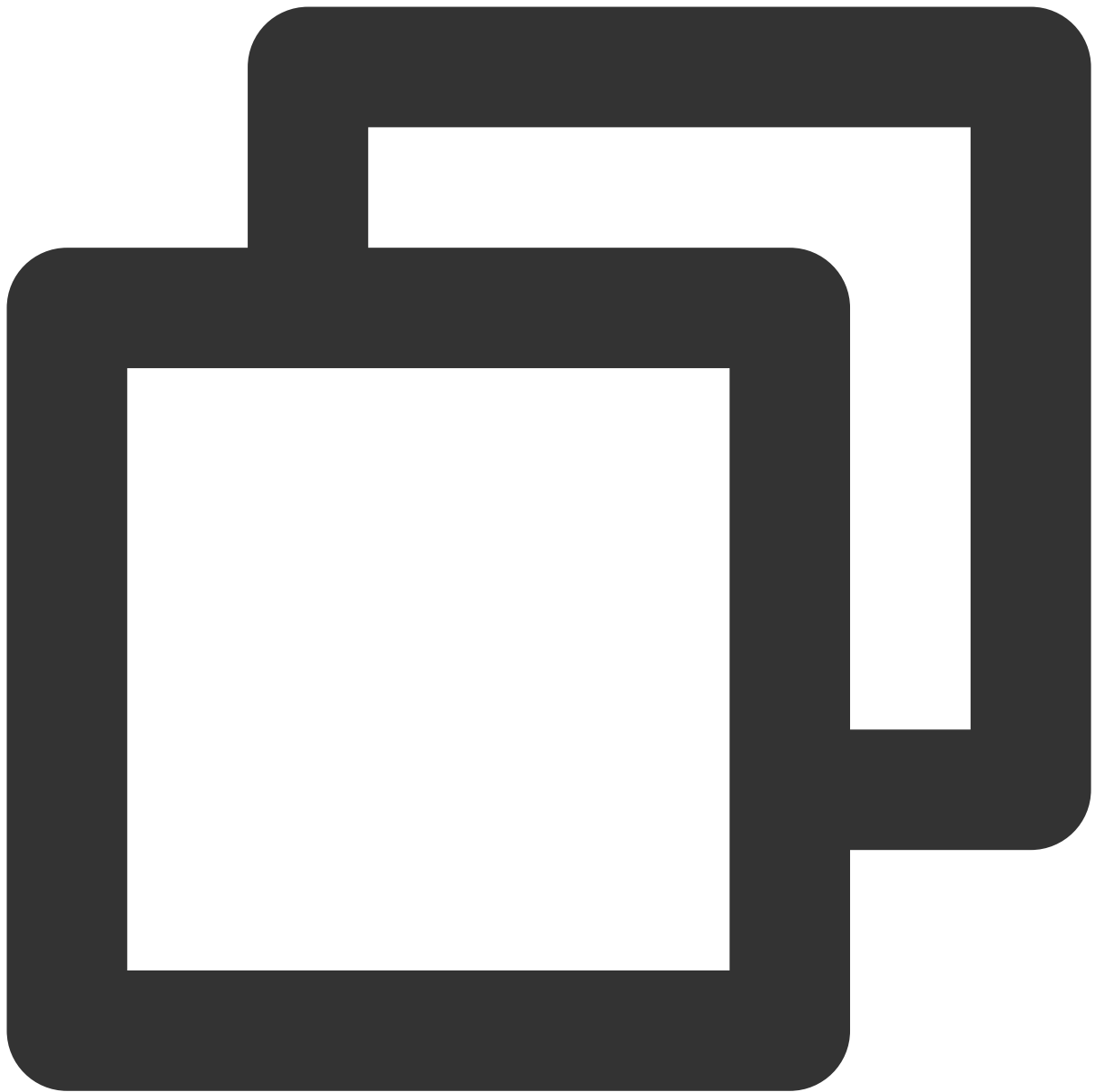
```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20210111;
using TencentCloud.Sms.V20210111.Models;
namespace TencentCloudExamples
{
    class PullSmsSendStatus
    {
        static void Main(string[] args)
```

```
{
    try
    {
        // 必要步骤：
        // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
        // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
        // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
        // 这里采用的是从环境变量读取的方式，需要在环境变量中先设置这两个值。
        // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cap
        Credential cred = new Credential {
            SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRETID"),
            SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRETKEY");
        };
        /* 非必要步骤：
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        ClientProfile clientProfile = new ClientProfile();
        /* SDK 默认用 TC3-HMAC-SHA256 进行签名
        * 非必要请不要修改该字段 */
        clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
        /* 非必要步骤
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        HttpProfile httpProfile = new HttpProfile();
        /* SDK 默认使用 POST 方法
        * 如需使用 GET 方法，可以在此处设置，但 GET 方法无法处理较大的请求 */
        httpProfile ReqMethod = "POST";
        httpProfile.Timeout = 30; // 请求连接超时时间，单位为秒(默认60秒)
        /* SDK 会自动指定域名，通常无需指定域名，但访问金融区的服务时必须手动指定域名
        * 例如 SMS 的上海金融区域名为 sms.ap-shanghai-fsi.tencentcloudapi.com */
        httpProfile.Endpoint = "sms.tencentcloudapi.com";
        // 代理服务器，当您的环境下有代理服务器时设定
        // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_PROXY");
        clientProfile.HttpProfile = httpProfile;
        /* 实例化 SMS 的 client 对象
        * 第二个参数是地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填
        SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile)
        /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
        * 您可以直接查询 SDK 源码确定 SendSmsRequest 有哪些属性可以设置
        * 属性可能是基本类型，也可能引用了另一个数据结构
        * 推荐使用 IDE 进行开发，可以方便地跳转查阅各个接口和数据结构的文档说明 */
        PullSmsSendStatusRequest req = new PullSmsSendStatusRequest();

        /* 基本类型的设置：
        * SDK 采用的是指针风格指定参数，即使对于基本类型也需要用指针来对参数赋值
        * SDK 提供对基本类型的指针引用封装函数
        * 帮助链接：
        * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
        * sms helper：https://intl.cloud.tencent.com/document/product/382/37
```

```
// 设置拉取最大条数，最多100条
req.Limit = 100;
/* 短信应用 ID：在 [短信控制台] 添加应用后生成的实际 SdkAppId，例如2400006666
req.SmsSdkAppId = "2400006666";
// 通过 client 对象调用 PullSmsSendStatus 方法发起请求，注意请求方法名与请求对象
// 返回的 resp 是一个 PullSmsSendStatusResponse 类的实例，与请求对象对应
PullSmsSendStatusResponse resp = client.PullSmsSendStatusSync(req);
// 输出 JSON 格式的字符串回包
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
Console.Read();
}
}
```

统计短信发送数据



```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20210111;
using TencentCloud.Sms.V20210111.Models;
namespace TencentCloudExamples
{
    class SendStatusStatistics
    {
        static void Main(string[] args)
```

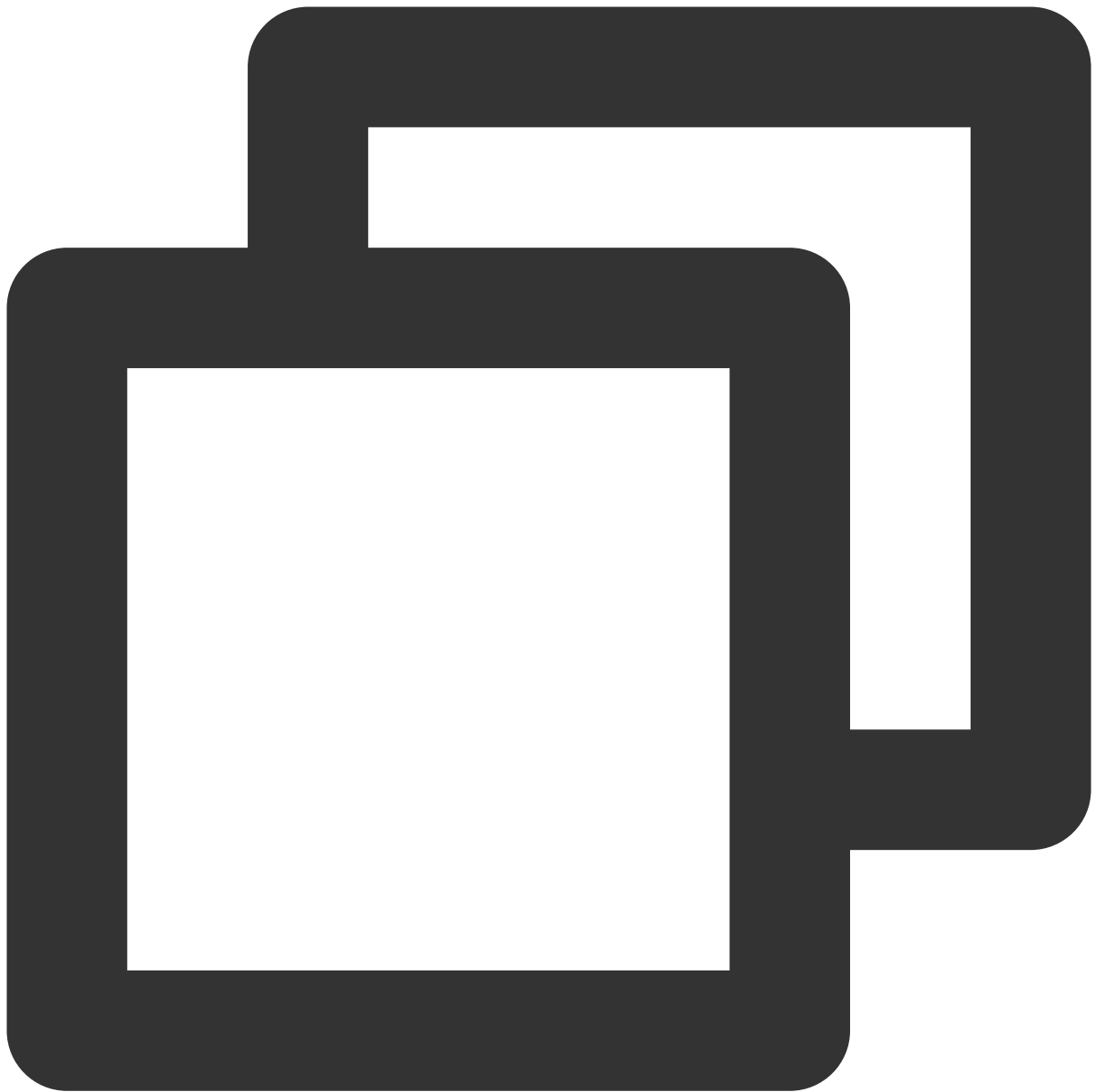
```
{
    try
    {
        // 必要步骤：
        // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
        // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
        // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
        // 这里采用的是从环境变量读取的方式，需要在环境变量中先设置这两个值。
        // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/c
        Credential cred = new Credential {
            SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECR
            SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SEC
        };
        /* 非必要步骤：
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        ClientProfile clientProfile = new ClientProfile();
        /* SDK 默认用 TC3-HMAC-SHA256 进行签名
        * 非必要请不要修改该字段 */
        clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
        /* 非必要步骤
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        HttpProfile httpProfile = new HttpProfile();
        /* SDK 默认使用 POST 方法
        * 如需使用 GET 方法，可以在此处设置，但 GET 方法无法处理较大的请求 */
        httpProfile ReqMethod = "POST";
        httpProfile.Timeout = 30; // 请求连接超时时间，单位为秒(默认60秒)
        /* SDK 会自动指定域名，通常无需指定域名，但访问金融区的服务时必须手动指定域名
        * 例如 SMS 的上海金融区域名为 sms.ap-shanghai-fsi.tencentcloudapi.com */
        httpProfile.Endpoint = "sms.tencentcloudapi.com";
        // 代理服务器，当您的环境下有代理服务器时设定
        // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_
        clientProfile.HttpProfile = httpProfile;
        /* 实例化 SMS 的 client 对象
        * 第二个参数是地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该
        SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile
        /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
        * 您可以直接查询 SDK 源码确定 SendSmsRequest 有哪些属性可以设置
        * 属性可能是基本类型，也可能引用了另一个数据结构
        * 推荐使用 IDE 进行开发，可以方便地跳转查阅各个接口和数据结构的文档说明 */
        SendStatusStatisticsRequest req = new SendStatusStatisticsRequest();

        /* 基本类型的设置：
        * SDK 采用的是指针风格指定参数，即使对于基本类型也需要用指针来对参数赋值
        * SDK 提供对基本类型的指针引用封装函数
        * 帮助链接：
        * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
        * sms helper：https://intl.cloud.tencent.com/document/product/382/3
```

```
*/

/* 短信应用 ID：在 [短信控制台] 添加应用后生成的实际 sdkAppId，例如2400006666
req.SmsSdkAppId = "2400006666";
// 设置拉取最大条数，最多100条
req.Limit = 5L;
/* 偏移量，目前固定设置为0 */
req.Offset = 0L;
/* 开始时间，yyyymmddhh 需要拉取的起始时间，精确到小时 */
req.BeginTime = "2019071100";
/* 结束时间，yyyymmddhh 需要拉取的截止时间，精确到小时
 * 注：EndTime 必须大于 BeginTime */
req.EndTime = "2019071123";
// 通过 client 对象调用 SendStatusStatistics 方法发起请求，注意请求方法名与
// 返回的 resp 是一个 SendStatusStatisticsResponse 类的实例，与请求对象对应
SendStatusStatisticsResponse resp = client.SendStatusStatisticsSync(
// 输出 JSON 格式的字符串回包
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
Console.Read();
}
}
```

申请短信模板



```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
using TencentCloud.Sms.V20210111;
using TencentCloud.Sms.V20210111.Models;
namespace TencentCloudExamples
{
    class AddSmsTemplate
    {
        static void Main(string[] args)
```

```
{
    try
    {
        // 必要步骤：
        // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
        // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
        // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
        // 这里采用的是从环境变量读取的方式，需要在环境变量中先设置这两个值。
        // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cap
        Credential cred = new Credential {
            SecretId = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRE
            SecretKey = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECR
        };
        /* 非必要步骤：
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        ClientProfile clientProfile = new ClientProfile();
        /* SDK 默认用 TC3-HMAC-SHA256 进行签名
        * 非必要请不要修改该字段 */
        clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
        /* 非必要步骤
        * 实例化一个客户端配置对象，可以指定超时时间等配置 */
        HttpProfile httpProfile = new HttpProfile();
        /* SDK 默认使用 POST 方法
        * 如需使用 GET 方法，可以在此处设置，但 GET 方法无法处理较大的请求 */
        httpProfile ReqMethod = "GET";
        httpProfile.Timeout = 30; // 请求连接超时时间，单位为秒(默认60秒)
        /* SDK 会自动指定域名，通常无需指定域名，但访问金融区的服务时必须手动指定域名
        * 例如 SMS 的上海金融区域名为 sms.ap-shanghai-fsi.tencentcloudapi.com */
        httpProfile.Endpoint = "sms.tencentcloudapi.com";
        // 代理服务器，当您的环境下有代理服务器时设定
        // httpProfile.WebProxy = Environment.GetEnvironmentVariable("HTTPS_P
        clientProfile.HttpProfile = httpProfile;
        /* 实例化 SMS 的 client 对象
        * 第二个参数是地域信息，根据您的国际站地域，如您选择的是新加坡国际站，则应该填
        SmsClient client = new SmsClient(cred, "ap-singapore", clientProfile)
        /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
        * 您可以直接查询 SDK 源码确定 SendSmsRequest 有哪些属性可以设置
        * 属性可能是基本类型，也可能引用了另一个数据结构
        * 推荐使用 IDE 进行开发，可以方便地跳转查阅各个接口和数据结构的文档说明 */
        AddSmsTemplateRequest req = new AddSmsTemplateRequest();

        /* 基本类型的设置：
        * SDK 采用的是指针风格指定参数，即使对于基本类型也需要用指针来对参数赋值
        * SDK 提供对基本类型的指针引用封装函数
        * 帮助链接：
        * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
        * sms helper：https://intl.cloud.tencent.com/document/product/382/37
```

```
*/

/* 模板名称 */
req.TemplateName = "腾讯云";
/* 模板内容 */
req.TemplateContent = "{1}为您的登录验证码, 请于{2}分钟内填写, 如非本人操作,
/* 短信类型, 1表示营销短信, 2表示通知短信, 3表示验证码短信 */
req.SmsType = 3;
/* 是否国际/港澳台短信:
 * 0: 表示中国大陆地区短信
 * 1: 表示国际/港澳台短信 */
req.International = 0;
/* 模板备注: 例如申请原因, 使用场景等 */
req.Remark = "xxx";

// 通过 client 对象调用 AddSmsTemplate 方法发起请求, 注意请求方法名与请求对象
// 返回的 resp 是一个 AddSmsTemplateResponse 类的实例, 与请求对象对应
AddSmsTemplateResponse resp = client.AddSmsTemplateSync(req);
// 输出 JSON 格式的字符串回包
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
Console.Read();
}
}
```

常见问题

代理设置

若在代理的环境下使用 SDK 进行接口调用, 则需设置系统环境变量 `https_proxy` (已在示例代码中体现), 否则可能出现无法正常调用、抛出连接超时异常的现象。

同步调用与异步调用

新版本 SDK 中同时提供了异步接口和同步接口, 同步接口统一在异步接口之后添加了 `Sync` 后缀, 在上述代码中已有样例。

注意:

在示例中由于是控制台应用程序, 因此可以使用同步方式调用异步接口, 即 `ConfigureAwait(false).GetAwaiter().GetResult()`。在开发 ASP 应用程序, 或者 Windows Forms 应用程序时, UI 控件的响应方法中, 不能使用同步方式调用异步接口, 否则会造成界面停止响应。

解决办法：将 UI 控件的响应方法改为异步，同时要注意同步上下文。另外，由于异步调用立即返回控制权给用户，很容易造成用户多次点击，或者用户进行了一些不期望的操作，程序中应注意此类问题。源码可以参考项目中的 WindowsFormsDemo 项目。

依赖版本

说明：

SDK 依赖的 FluentClient 使用的是3.2版本，但这个包目前发布了4.0版本且不兼容低版本，在 nuget 中升级此包到 4.0版本会导致无法调用或调用失败等问题。

Go SDK

最近更新时间：2024-06-27 15:48:01

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本。

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口。

一次群发请求最多支持200个号码。

签名、正文模板相关接口。

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口，请将“个人认证”变更为“企业认证”。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

通过 go get 安装（推荐）

Github 仓库参见：[通过 go get 安装（推荐）](#)

Gitee 仓库参见：[通过 go get 安装（推荐）](#)

通过源码安装

Github 仓库参见：[通过源码安装](#)

Gitee 仓库参见：[通过源码安装](#)

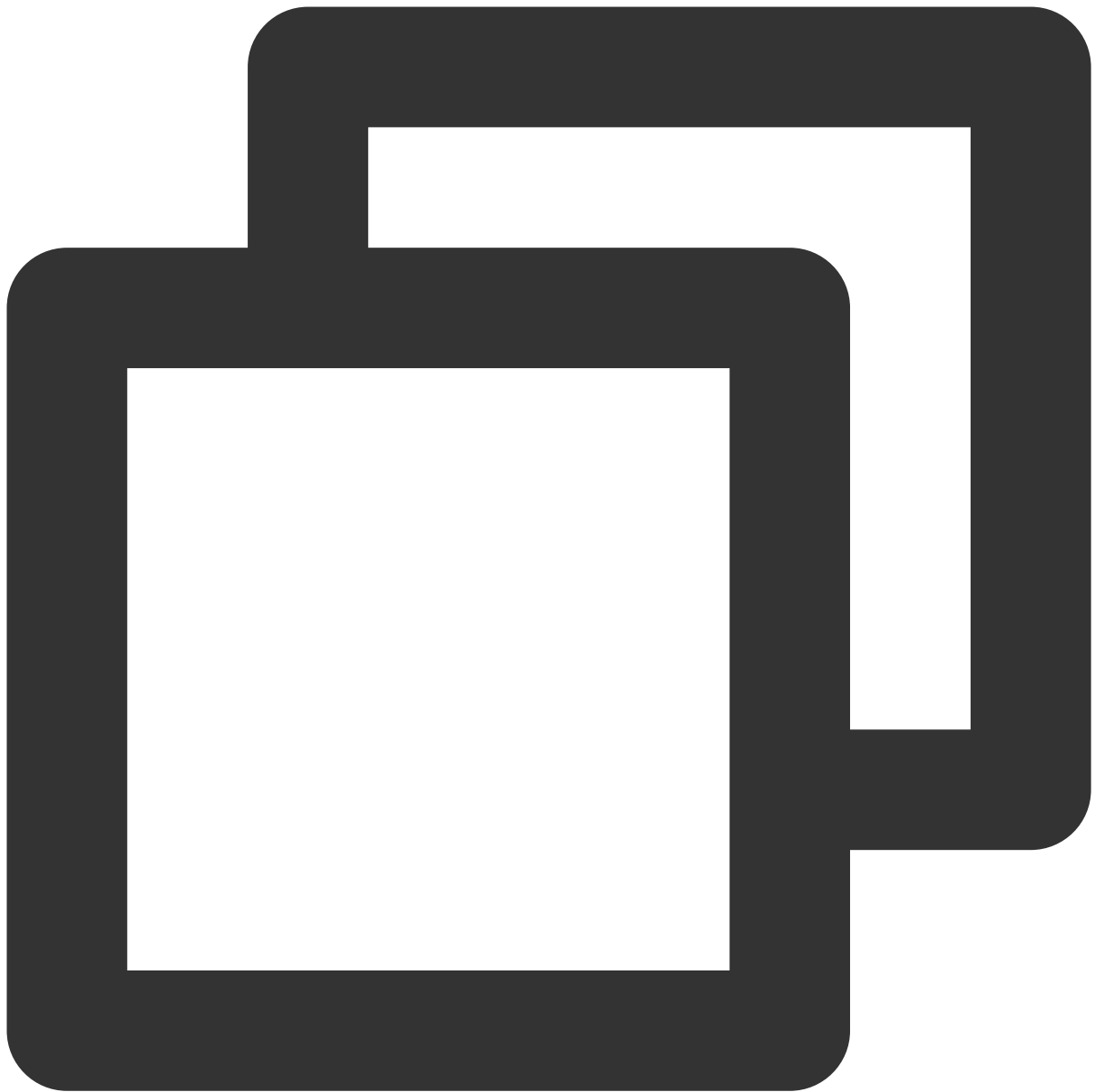
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



```
package main

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

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    sms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20210111" //
```

```
)

func main() {
    // 必要步骤：
    // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* 非必要步骤：
    * 实例化一个客户端配置对象，可以指定超时时间等配置 */
    cpf := profile.NewClientProfile()

    /* SDK默认使用POST方法。
    * 如果你一定要使用GET方法，可以在这里设置。GET方法无法处理一些较大的请求 */
    cpf.HttpProfile.ReqMethod = "POST"
    cpf.HttpProfile.ReqTimeout = 10 // 请求超时时间，单位为秒(默认60秒)
    /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com，也支持指定地域域名访问 */
    cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

    /* SDK默认用TC3-HMAC-SHA256进行签名，非必要请不要修改这个字段 */
    cpf.SignMethod = "HmacSHA1"

    /* 实例化要请求产品(以sms为例)的client对象
    * 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
    * 你可以直接查询SDK源码确定接口有哪些属性可以设置
    * 属性可能是基本类型，也可能引用了另一个数据结构
    * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
    request := sms.NewSendSmsRequest()

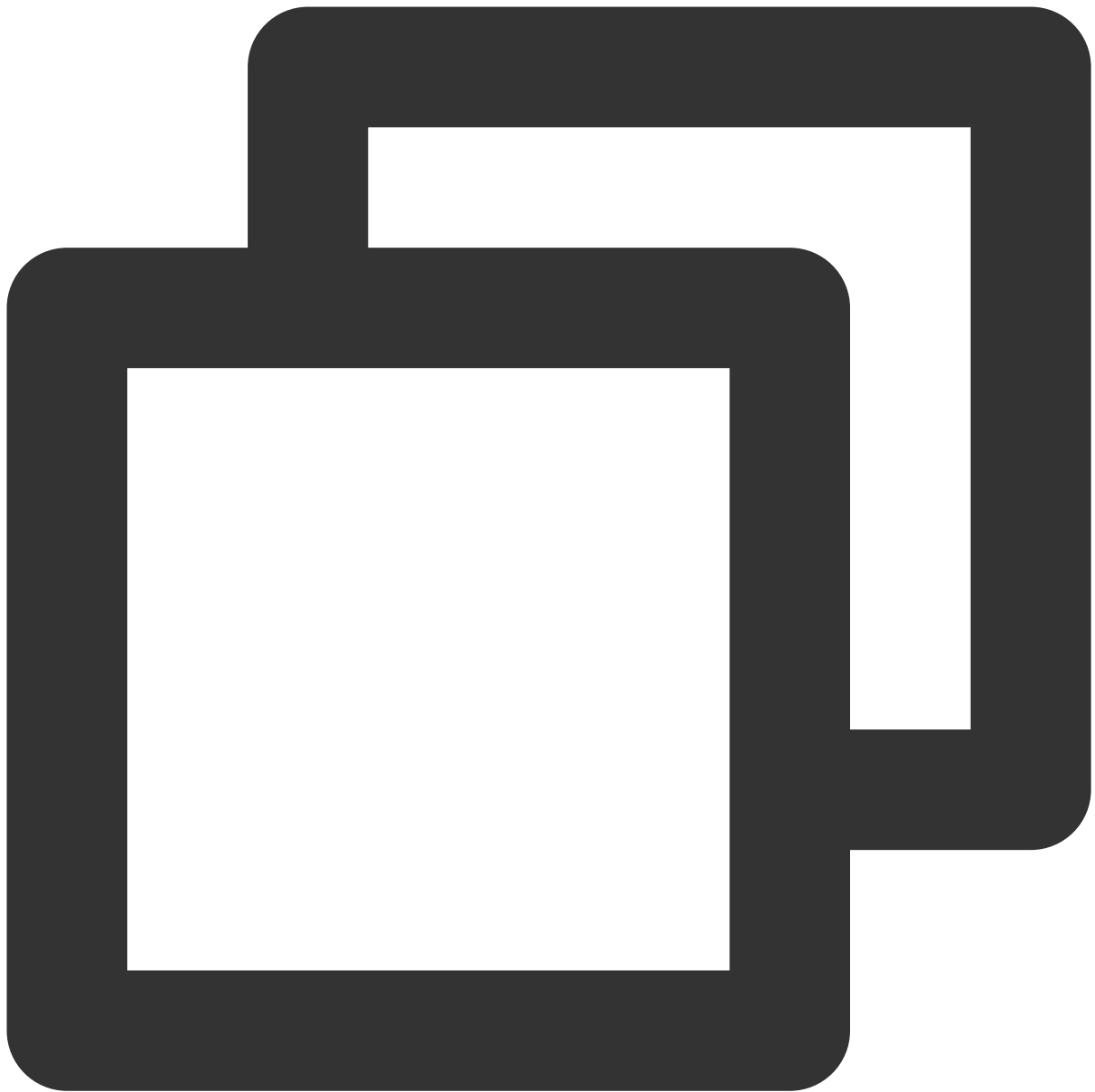
    /* 基本类型的设置：
    * SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
    * SDK提供对基本类型的指针引用封装函数
    * 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    * sms helper: https://intl.cloud.tencent.com/document/product/382/3773 */

    /* 短信应用ID：短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId，示例如2400006666 */
    request.SmsSdkAppId = common.StringPtr("2400006666")
    /* 短信签名内容：使用 UTF-8 编码，必须填写已审核通过的签名，签名信息可登录 [短信控制台] 查看
```

```
request.SignName = common.StringPtr("xxx")
/* 国际/港澳台短信 SenderId: 中国大陆地区短信填空, 默认未开通, 如需开通请联系 [sms helper] */
request.SenderId = common.StringPtr("")
/* 用户的 session 内容: 可以携带用户侧 ID 等上下文信息, server 会原样返回 */
request.SessionContext = common.StringPtr("xxx")
/* 短信码号扩展号: 默认未开通, 如需开通请联系 [sms helper] */
request.ExtendCode = common.StringPtr("")
/* 模板参数: 若无模板参数, 则设置为空 */
request.TemplateParamSet = common.StringPtrs([]string{"0"})
/* 模板 ID: 必须填写已审核通过的模板 ID。模板ID可登录 [短信控制台] 查看 */
request.TemplateId = common.StringPtr("449739")
/* 下发手机号码, 采用 E.164 标准, +[国家或地区码][手机号]
 * 示例如: +60198890000, 其中前面有一个+号, 60为国家码, 198890000为手机号, 最多不要超过
 */
request.PhoneNumberSet = common.StringPtrs([]string{"+60198890000"})

// 通过client对象调用想要访问的接口, 需要传入请求对象
response, err := client.SendSms(request)
// 处理异常
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    fmt.Printf("An API error has returned: %s", err)
    return
}
// 非SDK异常, 直接失败。实际代码中可以加入其他的处理。
if err != nil {
    panic(err)
}
b, _ := json.Marshal(response.Response)
// 打印返回的json字符串
fmt.Printf("%s", b)
}
```

拉取回执状态



```
package main

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

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    sms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20210111" //
```

```
)

func main() {
    // 必要步骤：
    // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* 非必要步骤：
    * 实例化一个客户端配置对象，可以指定超时时间等配置 */
    cpf := profile.NewClientProfile()

    /* SDK默认使用POST方法。
    * 如果你一定要使用GET方法，可以在这里设置。GET方法无法处理一些较大的请求 */
    cpf.HttpProfile.ReqMethod = "POST"
    cpf.HttpProfile.ReqTimeout = 30 // 请求超时时间，单位为秒(默认60秒)
    /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com ，也支持指定地域域名访问
    cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

    /* SDK默认用TC3-HMAC-SHA256进行签名
    * 非必要请不要修改这个字段 */
    cpf.SignMethod = "HmacSHA1"

    /* 实例化要请求产品(以sms为例)的client对象
    * 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
    * 你可以直接查询SDK源码确定接口有哪些属性可以设置
    * 属性可能是基本类型，也可能引用了另一个数据结构
    * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
    request := sms.NewPullSmsSendStatusRequest()

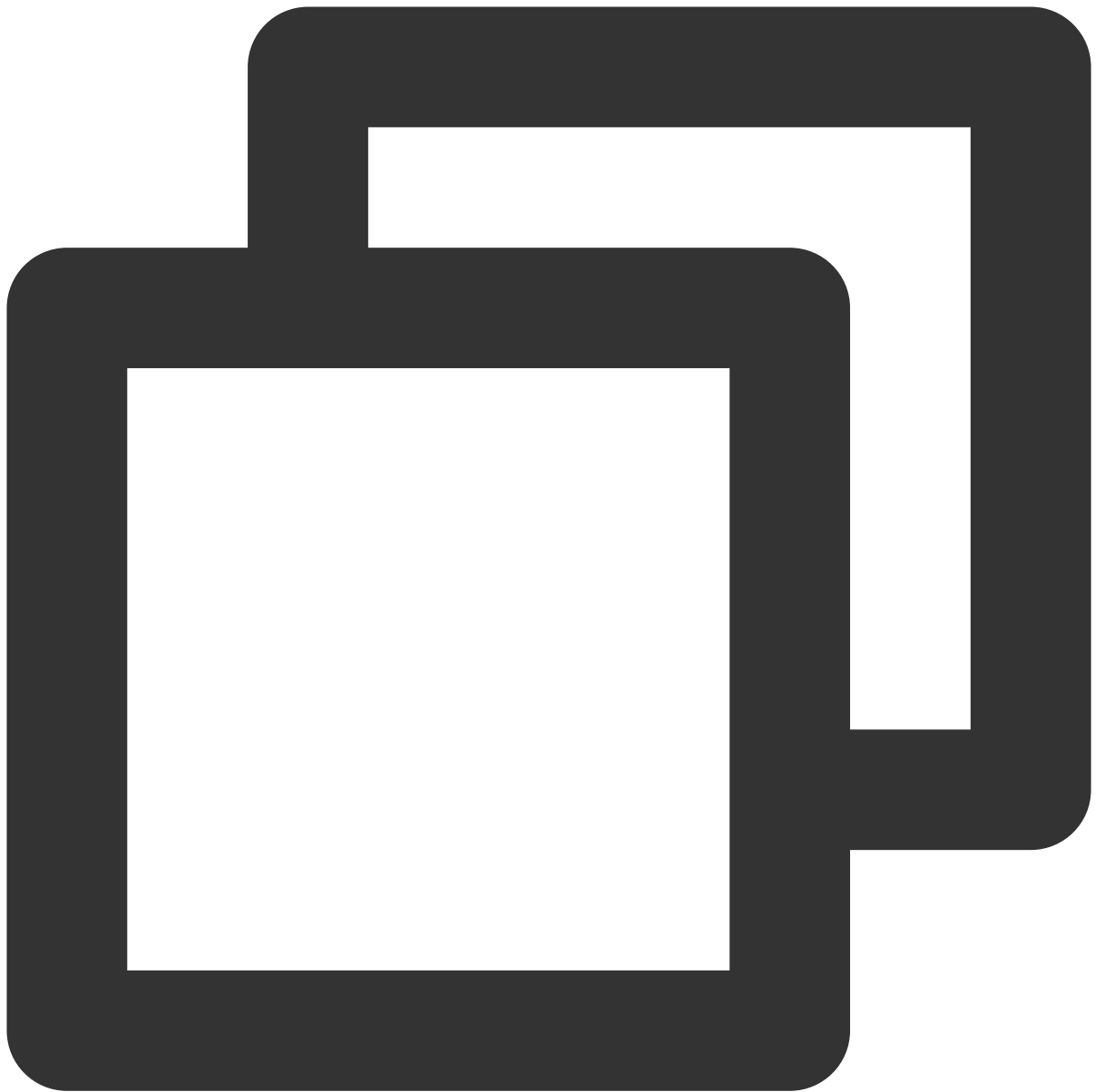
    /* 基本类型的设置：
    * SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
    * SDK提供对基本类型的指针引用封装函数
    * 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    * sms helper：https://intl.cloud.tencent.com/document/product/382/3773 */

    /* 短信应用ID：短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId，示例如2400006666
    request.SmsSdkAppId = common.StringPtr("2400006666")
```

```
/* 拉取最大条数, 最多100条 */
request.Limit = common.Uint64Ptr(10)

// 通过client对象调用想要访问的接口, 需要传入请求对象
response, err := client.PullSmsSendStatus(request)
// 处理异常
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    fmt.Printf("An API error has returned: %s", err)
    return
}
// 非SDK异常, 直接失败。实际代码中可以加入其他的处理。
if err != nil {
    panic(err)
}
b, _ := json.Marshal(response.Response)
// 打印返回的json字符串
fmt.Printf("%s", b)
}
```

统计短信发送数据



```
package main

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

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    sms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20210111" //
```

```
)

func main() {
    // 必要步骤：
    // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* 非必要步骤：
    * 实例化一个客户端配置对象，可以指定超时时间等配置 */
    cpf := profile.NewClientProfile()

    /* SDK默认使用POST方法。
    * 如果你一定要使用GET方法，可以在这里设置。GET方法无法处理一些较大的请求 */
    cpf.HttpProfile.ReqMethod = "POST"
    cpf.HttpProfile.ReqTimeout = 30 // 请求超时时间，单位为秒(默认60秒)
    /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com ，也支持指定地域域名访问 */
    cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

    /* SDK默认用TC3-HMAC-SHA256进行签名
    * 非必要请不要修改这个字段 */
    cpf.SignMethod = "HmacSHA1"

    /* 实例化要请求产品(以sms为例)的client对象
    * 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
    * 你可以直接查询SDK源码确定接口有哪些属性可以设置
    * 属性可能是基本类型，也可能引用了另一个数据结构
    * 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明 */
    request := sms.NewSendStatusStatisticsRequest()

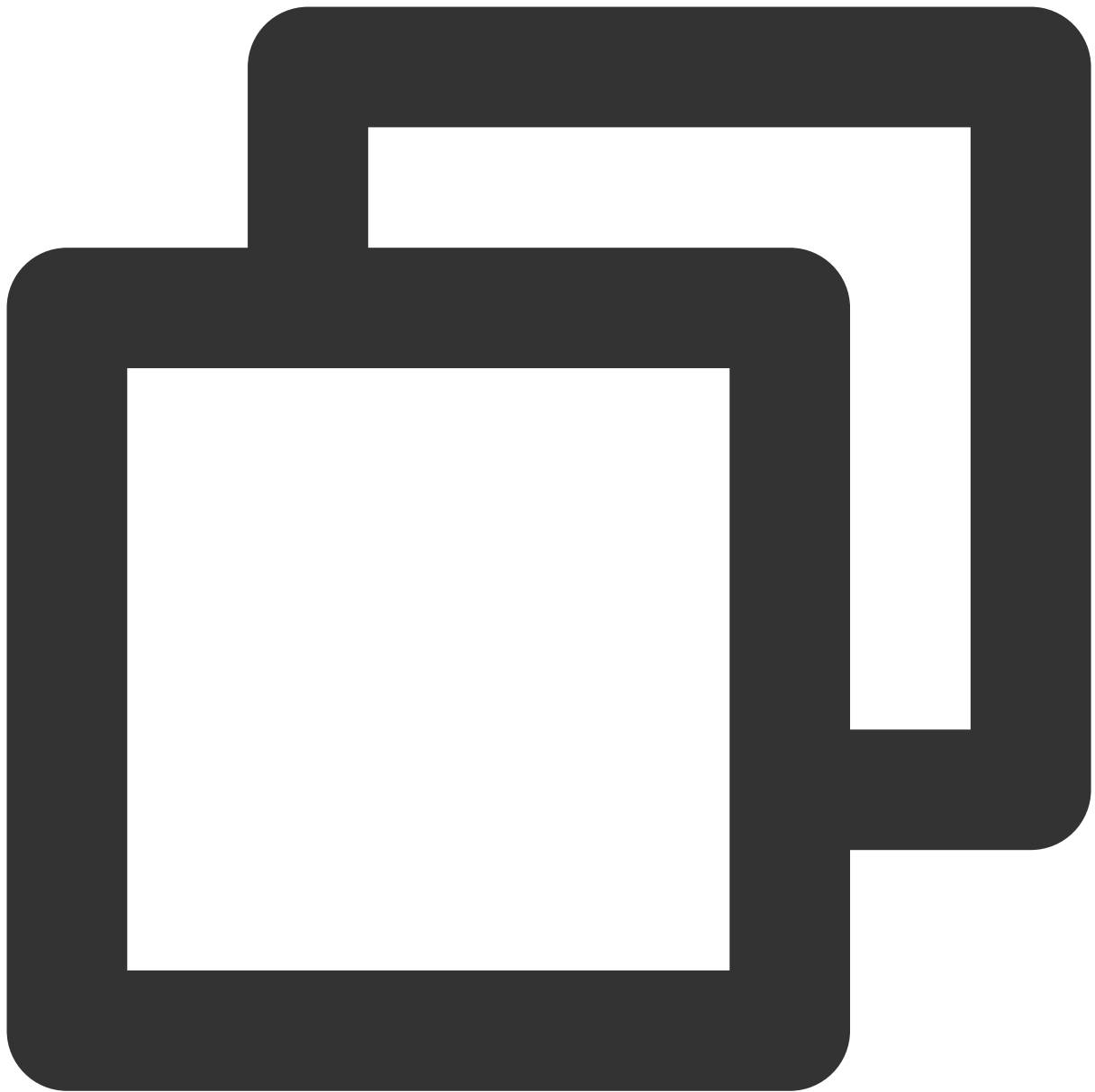
    /* 基本类型的设置：
    * SDK采用的是指针风格指定参数，即使对于基本类型你也需要用指针来对参数赋值。
    * SDK提供对基本类型的指针引用封装函数
    * 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    * sms helper：https://intl.cloud.tencent.com/document/product/382/3773 */

    /* 短信应用ID：短信SdkAppId在 [短信控制台] 添加应用后生成的实际SdkAppId，示例如2400006666 */
    request.SmsSdkAppId = common.StringPtr("2400006666")
```

```
/* 拉取最大条数，最多100条 */
request.Limit = common.Uint64Ptr(10)
/* 偏移量 注：目前固定设置为0 */
request.Offset = common.Uint64Ptr(0)
/* 开始时间，yyyymmddhh 需要拉取的起始时间，精确到小时 */
request.BeginTime = common.StringPtr("2019122400")
/* 结束时间，yyyymmddhh 需要拉取的截止时间，精确到小时
 * 注：EndTime 必须大于 BeginTime */
request.EndTime = common.StringPtr("2019122523")

// 通过client对象调用想要访问的接口，需要传入请求对象
response, err := client.SendStatusStatistics(request)
// 处理异常
if _, ok := err.(*errors.TencentCloudSDKError); ok {
    fmt.Printf("An API error has returned: %s", err)
    return
}
// 非SDK异常，直接失败。实际代码中可以加入其他的处理。
if err != nil {
    panic(err)
}
b, _ := json.Marshal(response.Response)
// 打印返回的json字符串
fmt.Printf("%s", b)
}
```

申请短信模板



```
package main

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

    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    sms "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/sms/v20210111" //
```

```
)

func main() {
    // 必要步骤：
    // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 SecretId, SecretKey。
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // SecretId、SecretKey 查询：https://console.intl.cloud.tencent.com/cam/capi
    // credential := common.NewCredential("SecretId", "SecretKey")
    credential := common.NewCredential(
        os.Getenv("TENCENTCLOUD_SECRET_ID"),
        os.Getenv("TENCENTCLOUD_SECRET_KEY"),
    )
    /* 非必要步骤：
    * 实例化一个客户端配置对象，可以指定超时时间等配置 */

    cpf := profile.NewClientProfile()

    /* SDK默认使用POST方法。
    * 如果你一定要使用GET方法，可以在这里设置。GET方法无法处理一些较大的请求 */
    cpf.HttpProfile.ReqMethod = "POST"
    cpf.HttpProfile.ReqTimeout = 30 // 请求超时时间，单位为秒(默认60秒)
    /* 指定接入地域域名，默认就近地域接入域名为 sms.tencentcloudapi.com，也支持指定地域域名访问 */
    cpf.HttpProfile.Endpoint = "sms.tencentcloudapi.com"

    /* SDK 默认用 TC3-HMAC-SHA256 进行签名，非必要请不要修改该字段 */
    cpf.SignMethod = "HmacSHA1"

    /* 实例化 SMS 的 client 对象
    * 第二个参数是地域信息，根据您选择的国际站地域，如您选择的是新加坡国际站，则应该填入字符串ap-
    client, _ := sms.NewClient(credential, "ap-singapore", cpf)

    /* 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
    * 您可以直接查询 SDK 源码确定接口有哪些属性可以设置
    * 属性可能是基本类型，也可能引用了另一个数据结构
    * 推荐使用 IDE 进行开发，可以方便地跳转查阅各个接口和数据结构的文档说明 */
    request := sms.NewAddSmsTemplateRequest()
    /* 基本类型的设置：
    * SDK 采用的是指针风格指定参数，即使对于基本类型也需要用指针来对参数赋值。
    * SDK 提供对基本类型的指针引用封装函数
    * 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    * sms helper：https://intl.cloud.tencent.com/document/product/382/3773
    */
    /* 模板名称 */
    request.TemplateName = common.StringPtr("腾讯云")
    /* 模板内容 */
```

```
request.TemplateContent = common.StringPtr("{1}为您的登录验证码, 请于{2}分钟内填写, 如  
/* 短信类型, 1表示营销短信, 2表示通知短信, 3表示验证码短信 */  
request.SmsType = common.Uint64Ptr(3)  
/* 是否国际/港澳台短信:  
    0: 表示中国大陆地区短信  
    1: 表示国际/港澳台短信 */  
request.International = common.Uint64Ptr(0)  
/* 模板备注: 例如申请原因, 使用场景等 */  
request.Remark = common.StringPtr("xxx")  
// 通过 client 对象调用想要访问的接口, 需要传入请求对象  
response, err := client.AddSmsTemplate(request)  
// 处理异常  
if _, ok := err.(*errors.TencentCloudSDKError); ok {  
    fmt.Printf("An API error has returned: %s", err)  
    return  
}  
// 非 SDK 异常, 直接失败。实际代码中可以加入其他的处理  
if err != nil {  
    panic(err)  
}  
b, _ := json.Marshal(response.Response)  
// 打印返回的 JSON 字符串  
fmt.Printf("%s", b)  
}
```

常见问题

代理设置

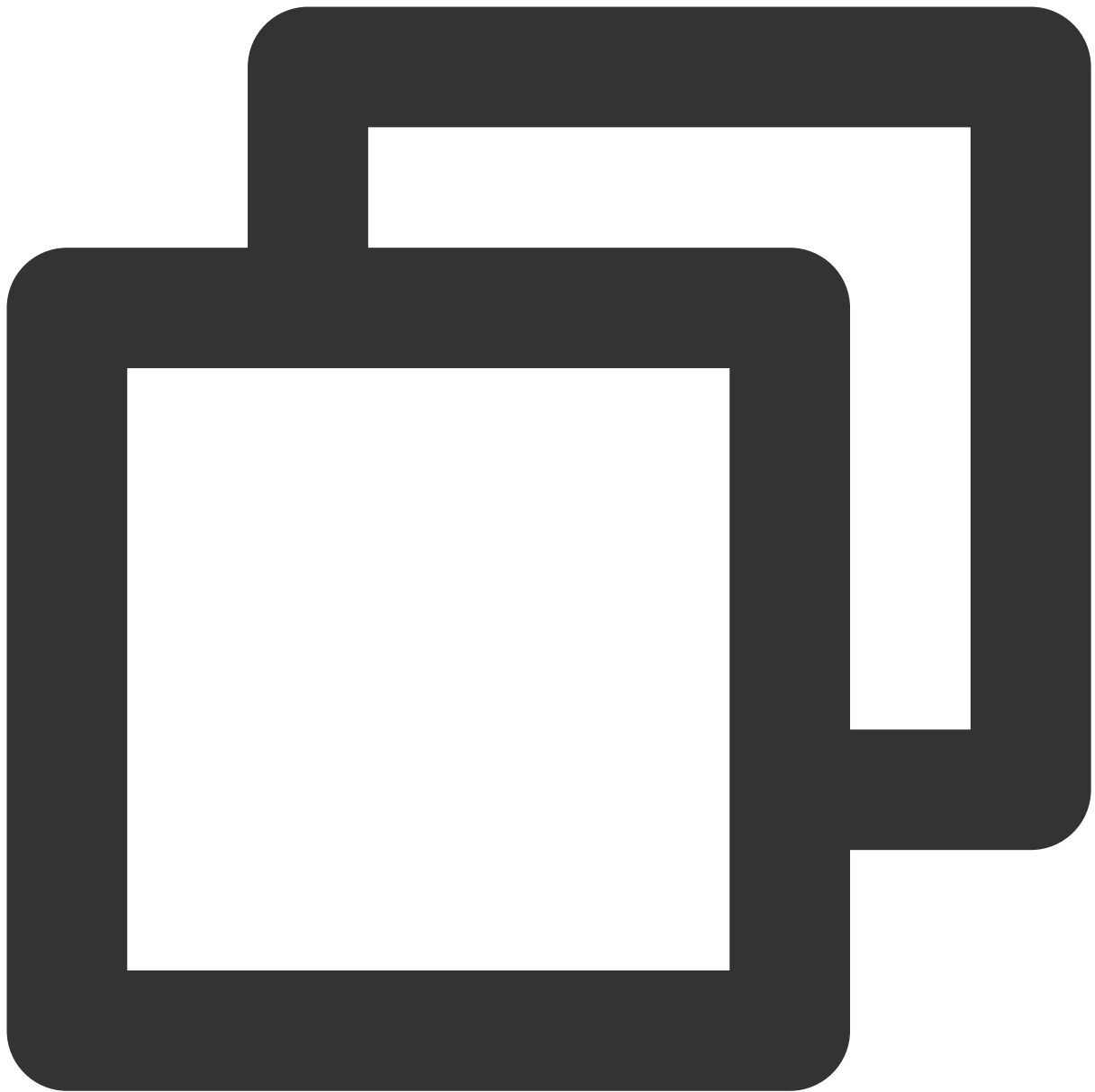
如有代理的环境下, 需要设置系统环境变量 `https_proxy`, 否则可能无法正常调用, 抛出连接超时的异常现象。

开启 DNS 缓存

当前 GO SDK 总是会去请求 DNS 服务器, 而没有使用到 `nscd` 的缓存, 可以通过导出环境变量 `GODEBUG=netdns=cgo`, 或者 `go build` 编译时指定参数 `-tags 'netcgo'` 控制读取 `nscd` 缓存。

忽略服务器证书校验

虽然使用 SDK 调用公有云服务时, 必须校验服务器证书, 以识别他人伪装的服务器, 确保请求的安全。但某些极端情况下, 例如测试时, 您可能会需要忽略自签名的服务器证书。以下是其中一种可能的方法:



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

注意：

除非您知道自己在进行何种操作，并明白由此带来的风险，否则不要尝试关闭服务器证书校验。

import 导包失败

例如报错：`imported and not used: "os"`，说明“os”这个包并未在代码中使用到，去掉即可。

C++ SDK

最近更新时间：2024-06-27 15:48:01

SDK 3.0是云 API 3.0平台的配套工具，您可以通过 SDK 使用所有 [短信 API](#)。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，错误码相同以及返回包格式相同等优点。

注意：

接入国际站要求的接口版本：

接入国际站需要使用短信 API 2021-01-11 版本，可参考示例代码。

发送短信相关接口：

一次群发请求最多支持200个号码。

签名、正文模板相关接口：

个人认证用户不支持使用签名、正文模板相关接口，只能通过短信控制台 [管理短信签名](#) 和 [管理短信正文模板](#)。如需使用该类接口，请将“个人认证”变更为“企业认证”，具体操作请参见 [实名认证变更指引](#)。

前提条件

了解 [地域](#) 概念，选择需求的地域接入。

已开通短信服务，创建签名和模板并通过审核，具体操作请参见 [国际/港澳台短信快速入门](#)。

已在访问管理控制台 > [API 密钥管理](#) 页面获取 SecretID 和 SecretKey。

SecretID 用于标识 API 调用者的身份。

SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥，**SecretKey 需妥善保管，避免泄露**。

短信的调用地址为 `sms.tencentcloudapi.com`。

相关资料

各个接口及其参数的详细介绍请参见 [API 文档](#)。

下载 SDK 源码请访问 [Github 仓库](#) 或者 [Gitee 仓库](#)。

安装 SDK

环境依赖

Github 仓库参见：[环境依赖](#)

Gitee 仓库参见：[环境依赖](#)

从源代码构建 SDK

Github 仓库参见：[从源代码构建 SDK](#)

Gitee 仓库参见：[从源代码构建 SDK](#)

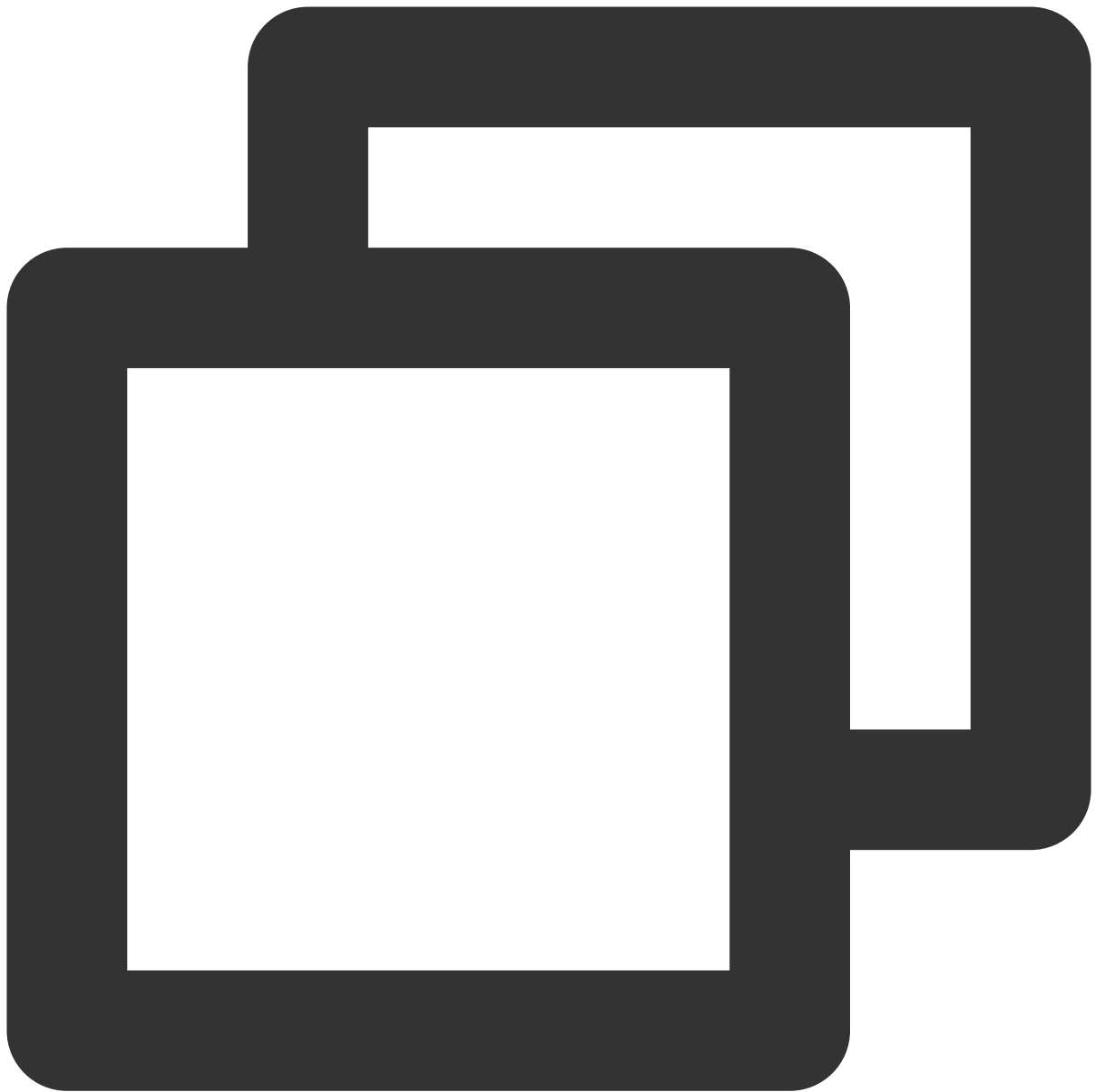
示例代码

说明：

所有示例代码仅作参考，无法直接编译和运行，需根据实际情况进行修改，您也可以根据实际需求使用 [API 3.0 Explorer](#) 自动化生成 Demo 代码。

每个接口都有一个对应的 Request 结构和一个 Response 结构。本文仅列举几个常用功能的示例代码，如下所示。

发送短信



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

#include <cstdlib>
```

```
#include <iostream>
#include <string>

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

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

    // use the sdk
    // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，切勿将密钥泄露给他人
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // 前往 https://console.intl.cloud.tencent.com/cam/capi 获取 API 密钥 SecretId Se
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项，可选的，没有特殊需求可以跳过。
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true); // 状态保持，默认是False
    httpProfile.SetEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名(默认就近接
    httpProfile.SetReqTimeout(10); // 请求超时时间，单位为秒(默认60秒)
    httpProfile.SetConnectTimeout(10); // 响应超时时间，单位是秒(默认是60秒)

    ClientProfile clientProfile = ClientProfile(httpProfile);

    SendSmsRequest req = SendSmsRequest();

    /* 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    */
    /* 短信应用 ID：在 [短信控制台] 添加应用后生成的实际 SdkAppId，例如2400006666 */
    req.SetSmsSdkAppId("2400006666");

    /* 短信签名内容：使用 UTF-8 编码，必须填写已审核通过的签名 */
    req.SetSignName("Tencent");

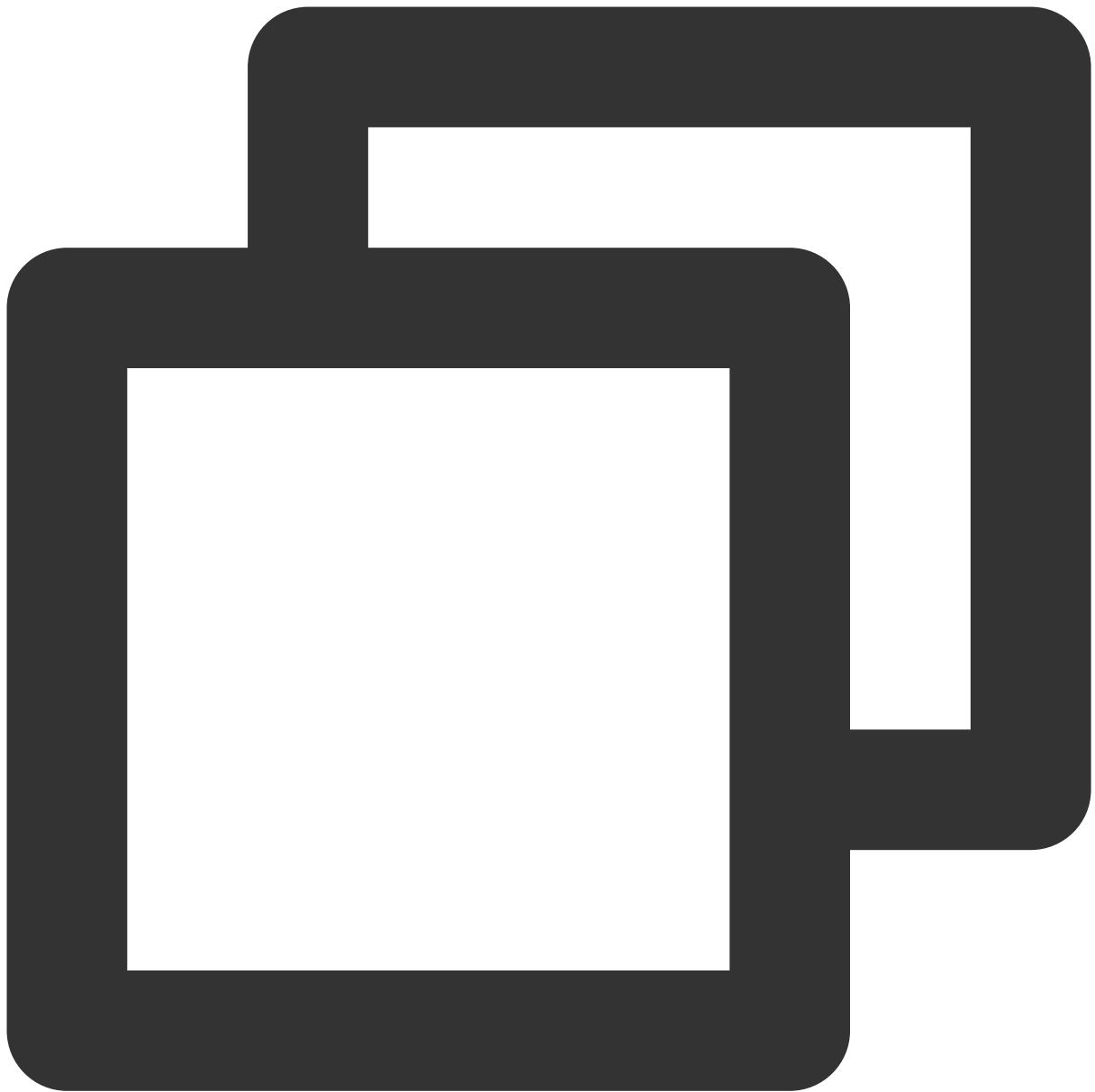
    /* 模板 ID：必须填写已审核通过的模板 ID */
    req.SetTemplateId("449739");

    /* 模板参数：模板参数的个数需要与 TemplateId 对应模板的变量个数保持一致，若无模板参数，则设
    req.SetTemplateParamSet(std::vector<std::string>{"1234"});

    /* 下发手机号码，采用 E.164 标准，+[国家或地区码][手机号]
```

```
return 0;
```

拉取回执状态



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

#include <cstdlib>
```

```
#include <iostream>
#include <string>

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

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

    // use the sdk
    // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，切勿将密钥泄露给他人
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // 前往 https://console.intl.cloud.tencent.com/cam/capi 获取 API 密钥 SecretId Se
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项，可选的，没有特殊需求可以跳过。
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true); // 状态保持，默认是False
    httpProfile.SetEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名(默认就近接
    httpProfile.SetReqTimeout(30); // 请求超时时间，单位为秒(默认60秒)
    httpProfile.SetConnectTimeout(30); // 响应超时时间，单位是秒(默认是60秒)

    ClientProfile clientProfile = ClientProfile(httpProfile);

    PullSmsSendStatusRequest req = PullSmsSendStatusRequest();

    /* 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    */
    /* 短信应用 ID：在 [短信控制台] 添加应用后生成的实际 SdkAppId，例如2400006666 */
    req.SetSmsSdkAppId("2400006666");
    // 设置拉取最大条数，最多100条
    req.SetLimit(100);

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

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

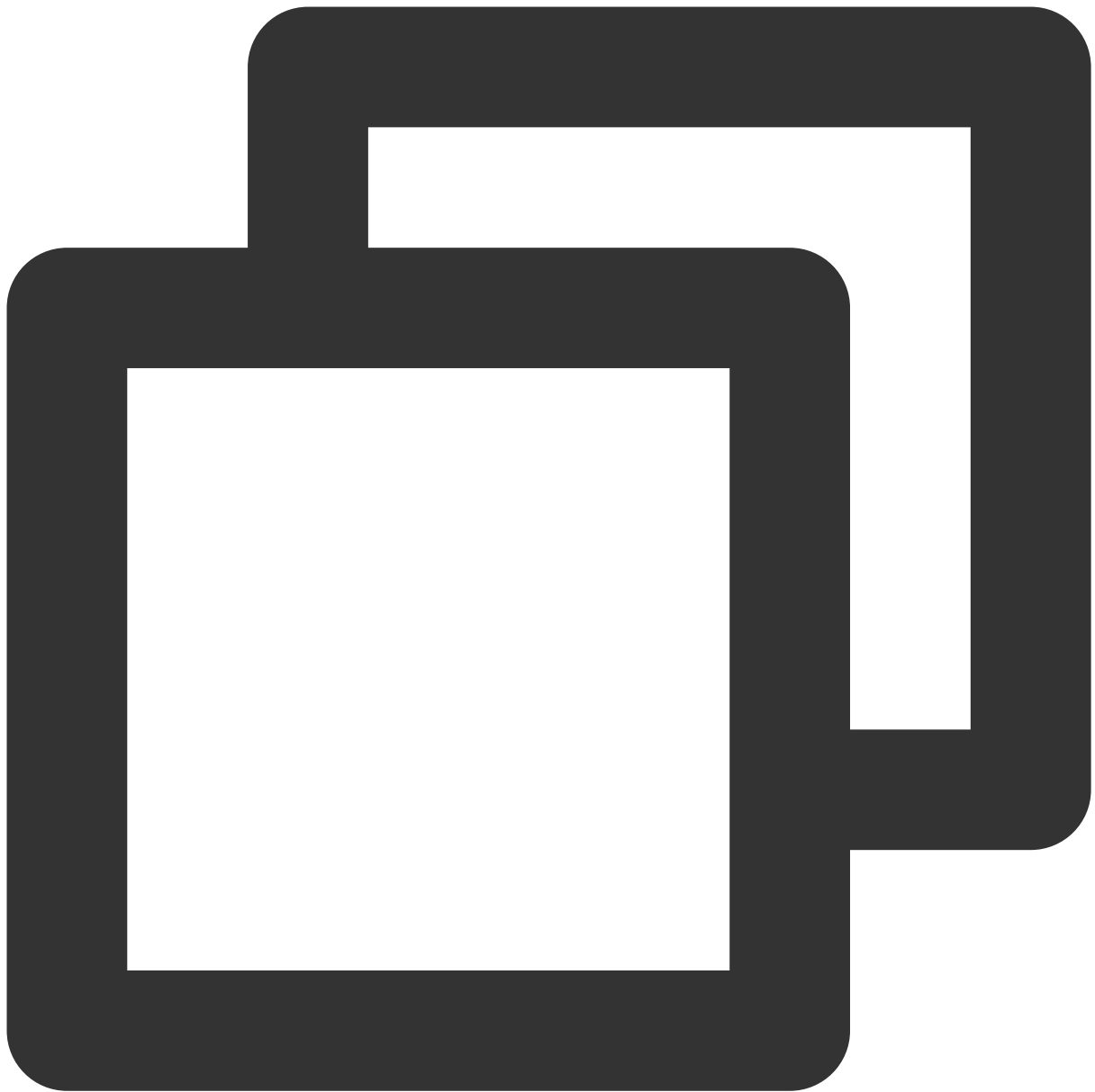
    auto outcome = sms_client.PullSmsSendStatus(req);
    if (!outcome.IsSuccess())
```

```
{
    cout << outcome.GetError().PrintAll() << endl;
    TencentCloud::ShutdownAPI();
    return -1;
}
PullSmsSendStatusResponse rsp = outcome.GetResult();
cout<<"RequestId="<<rsp.GetRequestId()<<endl;
cout<<"PullSmsSendStatusResponse="<<rsp.ToJsonString()<<endl;

TencentCloud::ShutdownAPI();

return 0;
}
```

统计短信发送数据



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

#include <cstdlib>
```

```
#include <iostream>
#include <string>

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

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

    // use the sdk
    // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，切勿将密钥泄露给他人
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // 前往 https://console.intl.cloud.tencent.com/cam/capi 获取 API 密钥 SecretId Se
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项，可选的，没有特殊需求可以跳过。
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true); // 状态保持，默认是False
    httpProfile.SetEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名(默认就近接
    httpProfile.SetReqTimeout(30); // 请求超时时间，单位为秒(默认60秒)
    httpProfile.SetConnectTimeout(30); // 响应超时时间，单位是秒(默认是60秒)

    ClientProfile clientProfile = ClientProfile(httpProfile);

    SendStatusStatisticsRequest req = SendStatusStatisticsRequest();

    /* 帮助链接：
     * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
     */
    /* 短信应用 ID：在 [短信控制台] 添加应用后生成的实际 SdkAppId，例如2400006666 */
    req.SetSmsSdkAppId("2400006666");
    // 最大上限，目前固定设置为0
    req.SetLimit(0);
    /* 偏移量，目前固定设置为0 */
    req.SetOffset(0);
    /* 设置起始时间，格式为yyyymmddhh，精确到小时，例如2021050113，表示2021年5月1号13时。 */
    req.SetBeginTime("2019071100");
    /* 设置结束时间，格式为yyyymmddhh，精确到小时，例如2021050118，表示2021年5月1号18时。
       注：EndTime 必须大于 BeginTime。 */
    req.SetEndTime("2019071123");

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

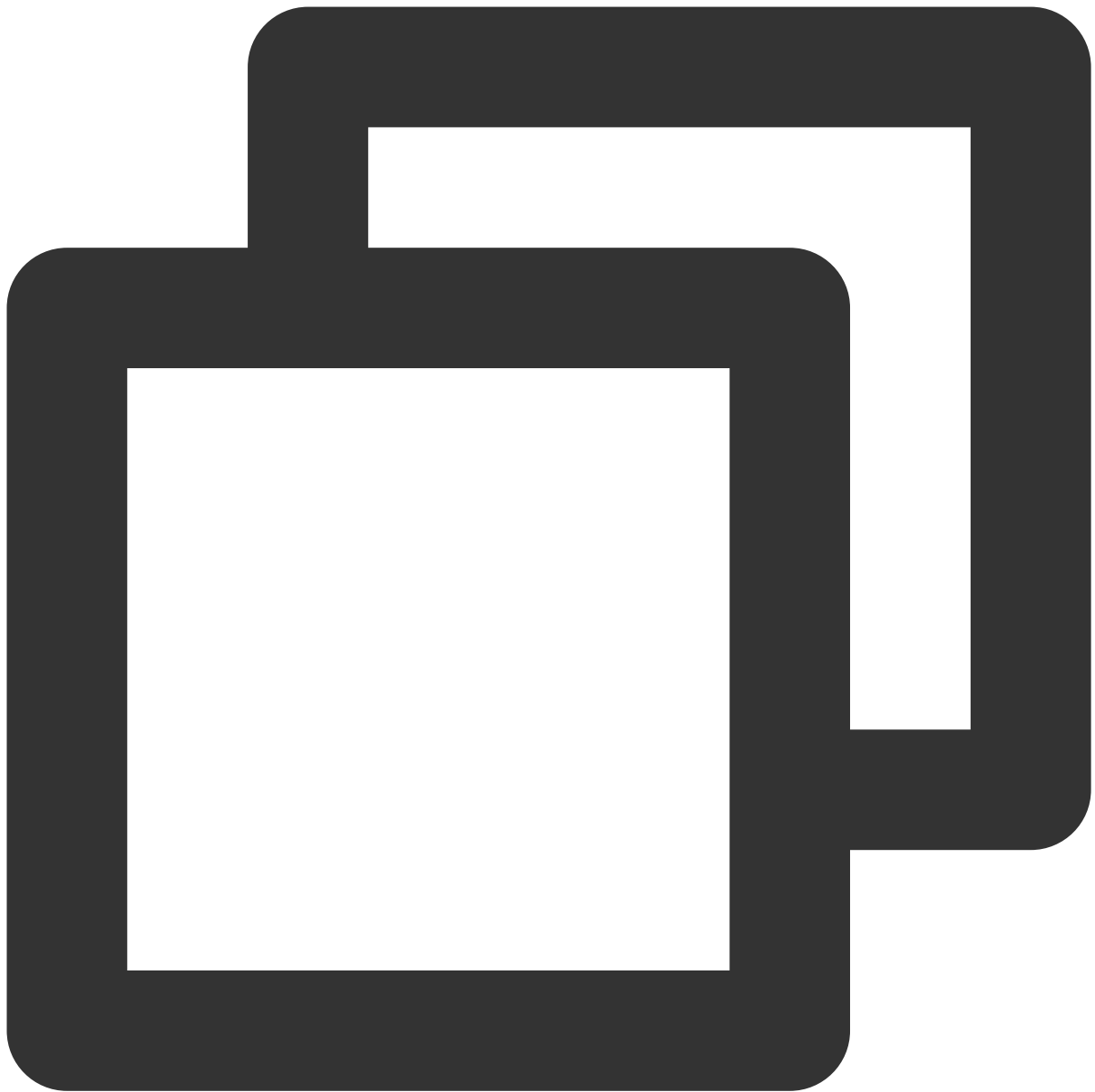
```
// set proxy
// NetworkProxy proxy = NetworkProxy(NetworkProxy::Type::HTTP, "localhost.proxy
// cvm_client.SetNetworkProxy(proxy);

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

TencentCloud::ShutdownAPI();

return 0;
}
```

申请短信模板



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

#include <cstdlib>
```

```
#include <iostream>
#include <string>

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

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

    // use the sdk
    // 实例化一个认证对象，入参需要传入腾讯云账户 SecretId 和 SecretKey，切勿将密钥泄露给他人
    // 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    // 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
    // 前往 https://console.intl.cloud.tencent.com/cam/capi 获取 API 密钥 SecretId Se
    Credential cred = Credential(getenv("TENCENTCLOUD_SECRET_ID"),
                                getenv("TENCENTCLOUD_SECRET_KEY"));

    // 实例化一个http选项，可选的，没有特殊需求可以跳过。
    HttpProfile httpProfile = HttpProfile();
    httpProfile.SetKeepAlive(true); // 状态保持，默认是False
    httpProfile.SetEndpoint("sms.tencentcloudapi.com"); // 指定接入地域域名(默认就近接
    httpProfile.SetReqTimeout(30); // 请求超时时间，单位为秒(默认60秒)
    httpProfile.SetConnectTimeout(30); // 响应超时时间，单位是秒(默认是60秒)

    ClientProfile clientProfile = ClientProfile(httpProfile);

    AddSmsTemplateRequest req = AddSmsTemplateRequest();

    /* 帮助链接：
    * 短信控制台：https://console.intl.cloud.tencent.com/smsv2
    */
    /* 模板名称 */
    req.SetTemplateName("腾讯云");
    /* 模板内容 */
    req.SetTemplateContent("{1}为您的登录验证码，请于{2}分钟内填写，如非本人操作，请忽略本短信");
    /* 短信类型，1表示营销短信，2表示通知短信，3表示验证码短信 */
    req.SetSmsType(3);
    /* 是否国际/港澳台短信：
    * 0：表示中国大陆地区短信
    * 1：表示国际/港澳台短信 */
    req.SetInternational(0);
    /* 模板备注：例如申请原因，使用场景等 */
    req.SetRemark("xxx");
```

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

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

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

TencentCloud::ShutdownAPI();

return 0;
}
```