无服务器云函数

SDK文档

产品文档

腾讯云

文档目录

SDK文档
Python

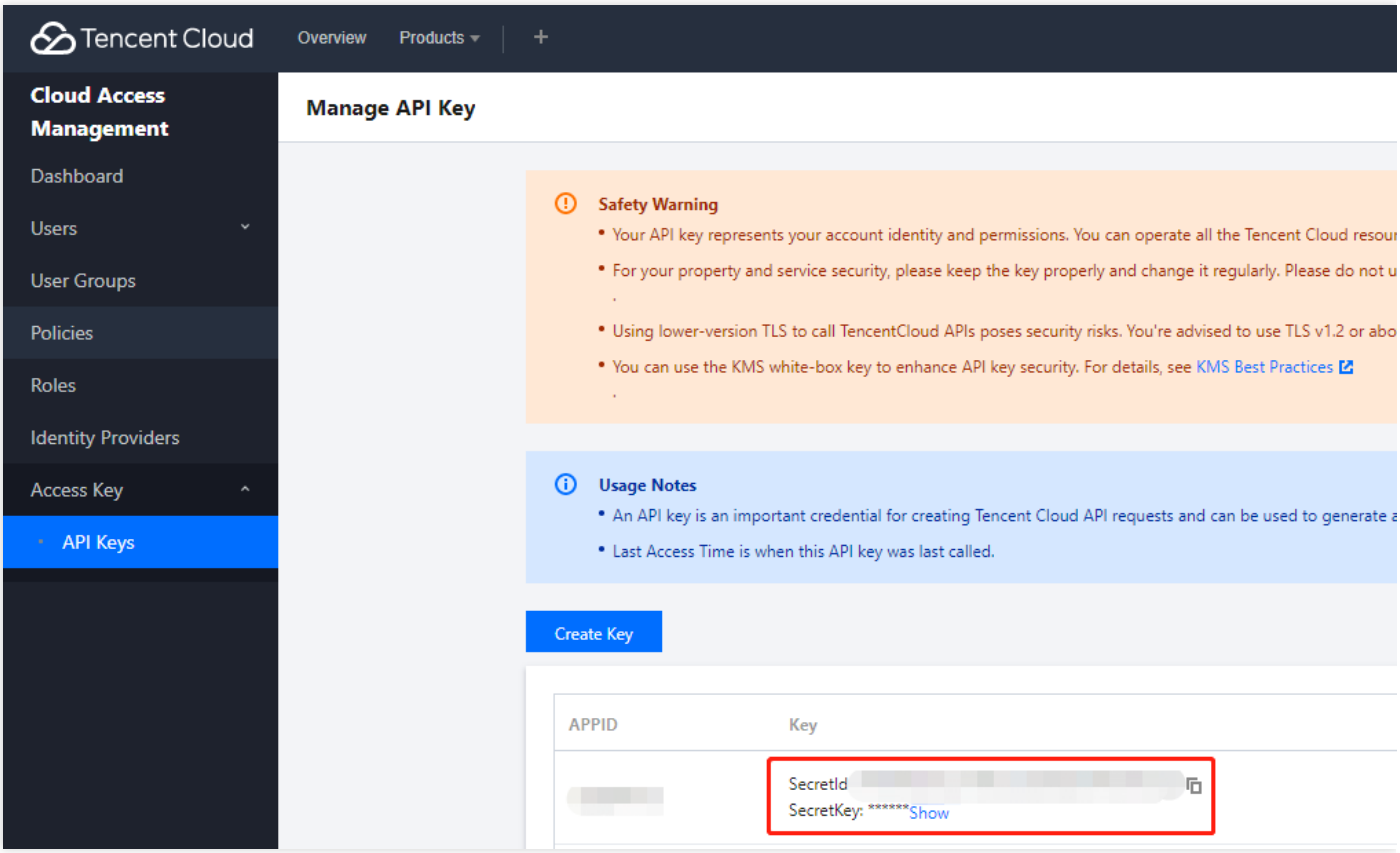
最近更新时间：2021-11-11 10:48:55

简介

- 欢迎使用腾讯云开发者工具套件（SDK）3.0，SDK 3.0 是云 API 3.0 平台的配套工具。SDK 3.0 实现了统一化，各个语言版本的 SDK 具备使用方法相同、接口调用方式相同、错误码和返回包格式相同等优点。
- 本文以 Python SDK 3.0 为例，介绍如何使用、调试并接入腾讯云产品 API。
- 目前已支持云服务器 CVM、私有网络 VPC 、云硬盘 CBS 等腾讯云产品，后续会支持其他云产品接入。

依赖环境

- Python 2.7，3.6至3.9版本。
- 获取安全凭证。安全凭证包含 SecretId 及 SecretKey 两部分。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。前往 API 密钥管理 页面，即可进行获取，如下图所示：



> 注意：
> **您的安全凭证代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。**

- 获取调用地址。调用地址（endpoint）一般形式为 `*.tencentcloudapi.com`，产品的调用地址有一定区别，例如，云服务器的调用地址为 `cvm.tencentcloudapi.com`。具体调用地址可参考对应产品的 API 文档。

安装 SDK

方式一、通过 Pip 安装（推荐）

可通过 pip 安装方式将腾讯云 Python SDK 安装至您的项目中。若您的项目环境未安装 pip，请前往 pip 官网 完成安装。
在命令行中执行以下命令，安装 Python SDK。

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

> 注意：
> 若同时具备 python2 及 python3 环境，则需使用 pip3 命令进行安装。

中国大陆地区的用户可以使用国内镜像源提高下载速度，例如：`pip install -i https://mirrors.tencent.com/pypi/simple/ --upgrade tencentcloud-sdk-python`。

> 说明：
> - 如果只想使用某个具体产品的包，例如云服务器 CVM，可以单独安装，但是注意不能和总包同时工作。`pip install --upgrade tencentcloud-sdk-python-common tencentcloud-sdk-python-cvm`

方式二、通过源码包安装

前往 Github 代码托管地址 下载最新代码，解压后：

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

使用 SDK

以查询实例列表接口为例。

- 简化版
- 详细版

```
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentClo
udSDKException
from tencentcloud.cvm.v20170312 import cvm_client, models
try:
cred = credential.Credential("secretId", "secretKey")
client = cvm_client.CvmClient(cred, "ap-shanghai")
req = models.DescribeInstancesRequest()
resp = client.DescribeInstances(req)
print(resp.to_json_string())
except TencentCloudSDKException as err:
print(err)
```

Commont Client 调用方式

从 `3.0.396` 开始，腾讯云 Python SDK 支持使用 `泛用型的API调用方式(Common Client)` 进行请求。您只需安装 tencentcloud-sdk-python-common 包，即可向任何产品发起调用。

> 说明：
> 您必须明确知道您调用的接口所需参数，否则可能会调用失败。

Common Client 请参见 example。

更多示例

您可以在 Github 中 examples 目录下找到更详细的示例。

相关配置

代理

如果是有代理的环境下，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。

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。

Node.js

最近更新时间：2021-07-22 14:25:25

开发准备

安装 Node.js SDK 前，需要先获取安全凭证。在第一次使用云 API 之前，用户首先需要在腾讯云控制台上申请安全凭证，安全凭证包括 SecretId 和 SecretKey， SecretId 是用于标识 API 调用者的身份，SecretKey 是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 必须严格保管，避免泄露。

开发环境

Node.js 8.9 版本

通过 npm 安装

通过 npm 获取安装是使用 NODEJS SDK 的推荐方法，npm 是 Node.js 的包管理工具。关于 npm 详细可参考 npm 官网 。

1. 执行以下安装命令：

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

2. 在您的代码中引用对应模块代码，请参考下面的示例。

通过源码包安装

1. 前往 Github 代码托管地址 下载源码压缩包。
2. 解压源码包到您项目合适的位置。
3. 在您的代码中引用对应模块代码，请参考下面的示例。

接口列表

接口名称	接口功能
CreateFunction	创建函数
DeleteFunction	删除函数
GetFunction	获取函数详细信息
GetFunctionLogs	获取函数运行日志

接口名称	接口功能
Invoke	运行函数
ListFunctions	获取函数列表
UpdateFunctionCode	更新函数代码
UpdateFunctionConfiguration	更新函数配置

示例

```
'use strict';
const tencentcloud = require("/var/user/tencentcloud-sdk-nodejs");
const Credential = tencentcloud.common.Credential;
// 导入对应产品模块的client models。
const ScfClient = tencentcloud.scf.v20180416.Client;
const models = tencentcloud.scf.v20180416.Models;
exports.main_handler = (event, context, callback) => {
console.log("Hello World")
console.log(event)
// console.log(context)
callback(null, event);
// 实例化一个认证对象，入参需要传入腾讯云账户secretId，secretKey
let cred = new Credential("AKIxxxxxxPDpqj3C", "75rxxxxxxxyJSODrMkx");
// 实例化要请求产品的client对象，以及函数所在的地域
let client = new ScfClient(cred, "ap-shanghai");

// 实例化一个请求对象,获取函数列表
console.log("Start ListFunctions")
let req = new models.ListFunctionsRequest();
// 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.ListFunctions(req, 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.to_json_string());
});
// 实例化一个请求对象,调用invoke()
console.log("Start Invoke")
let request = new models.InvokeRequest();
```

```
// 接口参数,输入需要调用的函数名，RequestResponse(同步) 和 Event(异步)
let params = '{"FunctionName":"test_python", "InvocationType":"RequestResponse"}'
request.from_json_string(params);
// 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.Invoke(request, 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.to_json_string());
},"test_python","RequestResponse");
};
```

打包部署

如果需要在云函数控制台中部署函数，并使用 SDK 调用其他函数，则需要把 tencentcloud 的库和函数代码一起打包成 zip 文件。

- 注意在控制台创建函数时的执行方法，需要和 zip 文件里的代码文件和执行函数对应。
- 最终生成的 zip 包如果大于50MB，需要通过 COS 上传。
- 云 API 默认限频为每秒20次，如需提升并发上限，可以 提交工单 申请。

相关信息

您也可以使用腾讯云云函数 SDK（Tencentserverless SDK），该 SDK 集成云函数业务流接口，简化云函数的调用方法，使您无需再进行公有云 API 接口的封装。详情请参见 函数间调用 SDK。

PHP

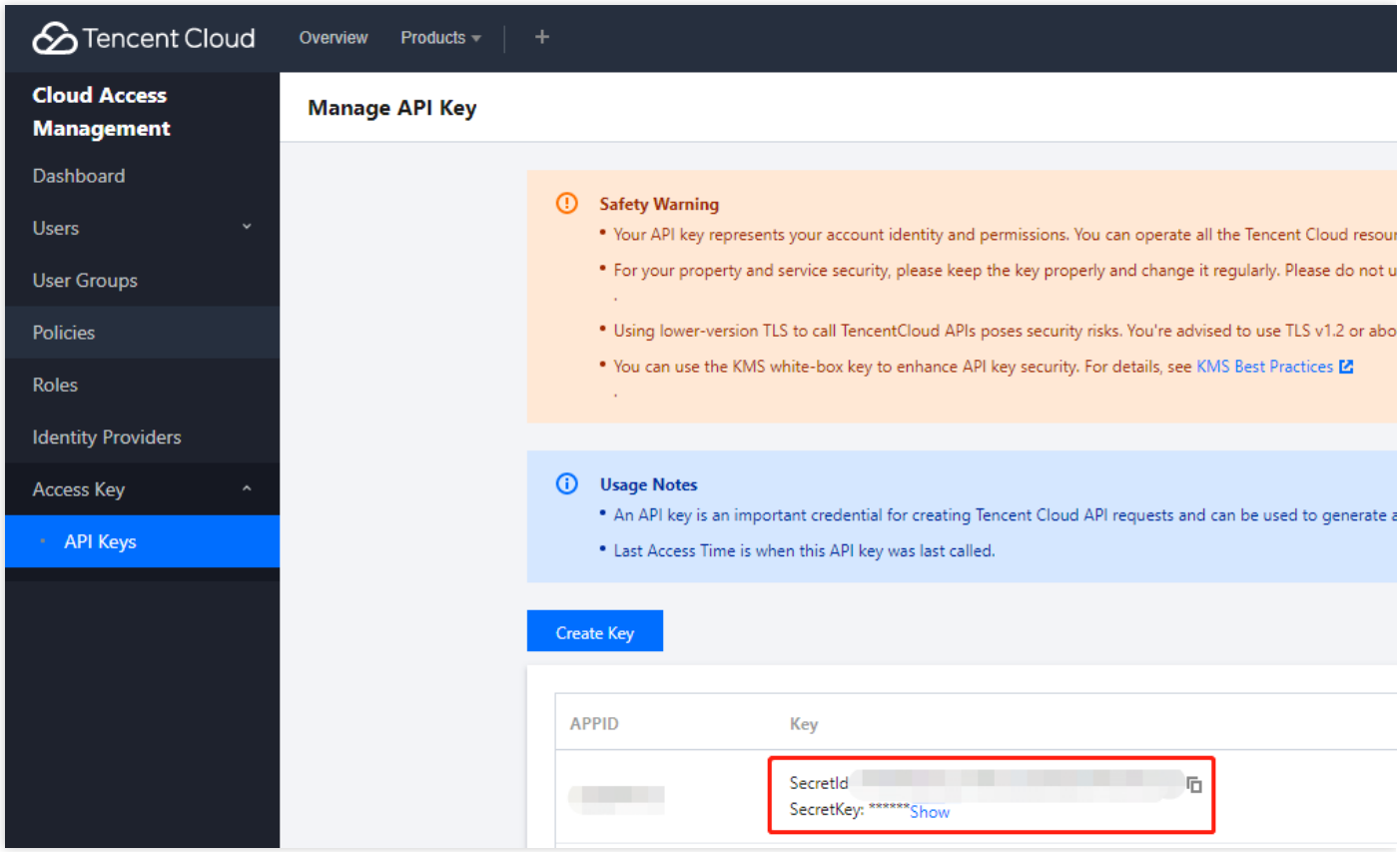
最近更新时间：2021-08-10 16:14:49

简介

- 欢迎使用腾讯云开发者工具套件（SDK）3.0，SDK 3.0 是云 API 3.0 平台的配套工具。SDK 3.0 实现了统一化，各个语言版本的 SDK 具备使用方法相同、接口调用方式相同、错误码和返回包格式。
- 本文以 PHP SDK 3.0 为例，介绍如何使用、调试并接入腾讯云产品 API。
- 目前已支持云服务器 CVM、私有网络 VPC 、云硬盘 CBS 等 腾讯云产品，后续会支持其他云产品接入。

依赖环境

- PHP 5.6.33 版本及以上。
- 获取安全凭证。安全凭证包含 SecretId 及 SecretKey 两部分。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。前往 API 密钥管理 页面，即可进行获取，如下图所示：



> 注意：
> **您的安全凭证代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。**

- 获取调用地址。调用地址（endpoint）一般形式为 `*.tencentcloudapi.com`，产品的调用地址有一定区别，例如，云服务器的调用地址为 `cvm.tencentcloudapi.com`。具体调用地址可参考对应产品的 API 文档。

安装 PHP SDK 3.0

通过 Composer 获取安装是使用 PHP SDK 推荐方法，Composer 是 PHP 的依赖管理工具。关于 Composer 详细可参见 Composer 官网。

> 说明：
> Composer 需要 PHP 5.3.2+ 以上版本，且需要开启 openssl。

步骤1：安装 Composer

- Windows 环境请访问 Composer 官网下载安装包安装。
- Unix 环境在命令行中执行以下命令：

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

```
sudo mv composer.phar /usr/local/bin/composer
```

步骤2：添加镜像源

中国大陆地区的用户可以使用腾讯云镜像源提高下载速度，在打开的命令窗口执行以下命令：

```
composer config -g repos.packagist composer https://mirrors.tencent.com/composer/
```

步骤3：添加依赖

在打开的命令窗口执行命令安装 SDK（安装到指定位置），例如安装到 `C:\Users\···>` 目录下，则在指定的位置打开命令窗口，并执行以下命令：

```
composer require tencentcloud/tencentcloud-sdk-php
```

步骤4：添加引用

在代码中添加以下引用代码。注意：如下仅为示例，Composer 会在项目根目录下生成 vendor 目录，/path/to/ 为项目根目录的实际绝对路径（如果是在当前目录执行，可以省略绝对路径）。

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

说明：

- 如果只想安装某个产品的，可以使用 `composer require tencentcloud/` 产品名，例如 `composer require tencentcloud/cvm` 。

使用 SDK

- 推荐使用 API 3.0 Explorer，提供在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 3.0 和 SDK 的难度。
- 还可以参考 SDK 仓库中 examples 目录中的示例，展示了更多的用法。

下面以查询实例接口 DescribeInstances 为例:

- 简化版
- 详细版

```
<?php
require_once '/path/to/vendor/autoload.php';
use TencentCloud\Cvm\V20170312\Models\DescribeInstancesRequest;
use TencentCloud\Common\Exception\TencentCloudSDKException;
use TencentCloud\Common\Credential;
try {
$cred = new Credential("secretId", "secretKey");
$client = new CvmClient($cred, "ap-guangzhou");
$req = new DescribeInstancesRequest();
$resp = $client->DescribeInstances($req);
print_r($resp->toJsonString());
}
catch(TencentCloudSDKException $e) {
echo $e;
}
```

更多示例

您可以在 Github Examples 目录下找到更详细的示例。

相关配置

代理

如果是有代理的环境下，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。或者使用 GuzzleHttp 代理配置:

```
$cred = new Credential("secretId", "secretKey");
$httpProfile = new HttpProfile();
$httpProfile->setProxy('https://ip:port');
$clientProfile = new ClientProfile();
$clientProfile->setHttpProfile($httpProfile);
$client = new OcrClient($cred, 'ap-beijing', $this->clientProfile);
```

常见问题

展开全部

证书问题

展开&收起

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

展开&收起

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

展开&收起

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

展开&收起

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。

Java

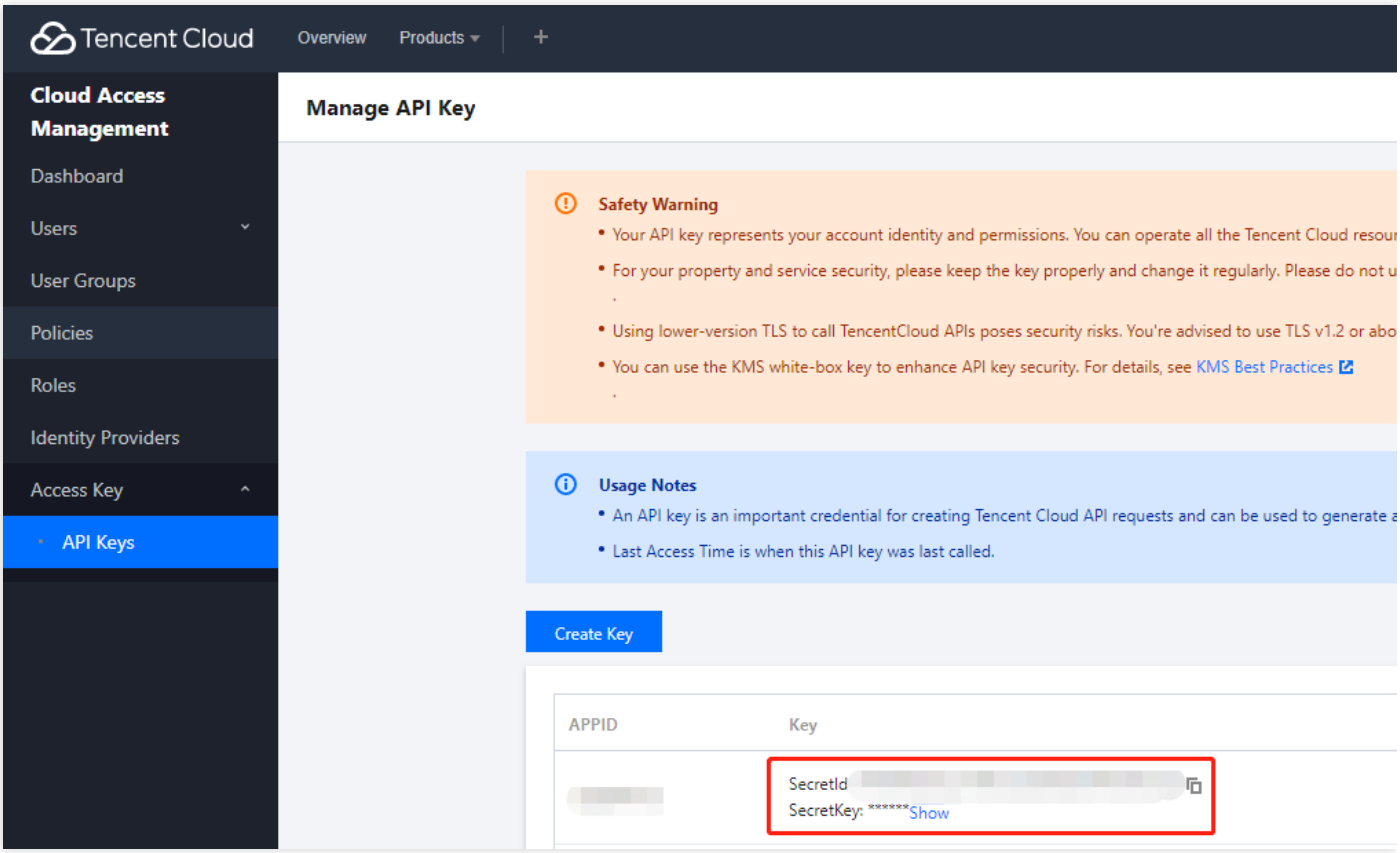
最近更新时间：2021-08-10 16:15:23

简介

- 欢迎使用腾讯云开发者工具套件（SDK）3.0，SDK 3.0 是云 API 3.0 平台的配套工具。SDK 3.0 实现了统一化，各个语言版本的 SDK 具备使用方法相同、接口调用方式相同、错误码和返回包格式相同等优点。
- 本文以 Java SDK 3.0 为例，介绍如何使用、调试并接入腾讯云产品 API。
- 目前已支持云服务器 CVM、私有网络 VPC 、云硬盘 CBS 等 腾讯云产品，后续会支持其他云产品接入。

依赖环境

- JDK 7版本及以上。
- 获取安全凭证。安全凭证包含 SecretId 及 SecretKey 两部分。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。前往 API 密钥管理 页面，即可进行获取，如下图所示：



> 注意：
> **您的安全凭证代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。**

- 获取调用地址。调用地址（endpoint）一般形式为 `*.tencentcloudapi.com`，产品的调用地址有一定区别，例如，云服务器的调用地址为 `cvm.tencentcloudapi.com`。具体调用地址可参考对应产品的 API 文档。

安装 SDK

方式一、通过 Maven 安装（推荐）

Maven 是 JAVA 的依赖管理工具，支持您项目所需的依赖项，并将其安装到项目中。

1. 请访问 Maven官网 下载对应系统 Maven 安装包进行安装，关于 Maven 详细可参考 Maven 官网。

2. 为您的项目添加 Maven 依赖项，只需在 pom.xml 中找到 `<dependencies>` 标签，在里面添加以下依赖项即可。

```
<dependency>
<groupId>com.tencentcloudapi</groupId>
<artifactId>tencentcloud-sdk-java</artifactId>
<!-- go to https://search.maven.org/search?q=tencentcloud-sdk-java and get the latest version. -->
<!-- 请到https://search.maven.org/search?q=tencentcloud-sdk-java查询所有版本，最新版本如下 -->
<version>3.1.217</version>
</dependency>
```

> 注意：
> - 这里的版本号只是举例，您可以在 Maven 仓库 上找到最新的版本。
> - Maven 仓库 中显示的4.0.11是废弃版本，由于 Maven 索引更新问题尚未完全删除。
> - 若上面的引用方式会将腾讯云所有产品 SDK 下载到本地，可以将 artifactId 换成 tencentcloud-sdk-java-cvm/cbs/vpc 等，即可引用特定产品的 SDK，代码中使用方式和大包相同，可参考示例。最新版本也可在 Maven仓库 查询，可大 大节省存储空间。

3. 设置镜像源以加快下载速度，编辑 maven 的 settings.xml 配置文件，在 mirrors 段落增加镜像配置：

```
<repositories>
<repository>
<id>nexus-tencentyun</id>
<name>Nexus tencentyun</name>
<url>https://mirrors.tencent.com/nexus/repository/maven-public/</url>
</repository>
</repositories>
```

方式二、通过源码包安装

1. 前往 Github 代码托管地址 下载源码压缩包。
2. 解压源码包到您项目合适的位置。
3. 需要将 vendor 目录下的 jar 包放在 java 可找到的路径中。
4. 引用方法可参考示例。

使用 SDK

- 简化版
- 详细版

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
import com.tencentcloudapi.cvm.v20170312.CvmClient;
import com.tencentcloudapi.cvm.v20170312.models.DescribeInstancesRequest;
import com.tencentcloudapi.cvm.v20170312.models.DescribeInstancesResponse;
public class DescribeInstances {
public static void main(String[] args) {
try {
Credential cred = new Credential("secretId", "secretKey");
CvmClient client = new CvmClient(cred, "ap-shanghai");
DescribeInstancesRequest req = new DescribeInstancesRequest();
DescribeInstancesResponse resp = client.DescribeInstances(req);
System.out.println(DescribeInstancesResponse.toJsonString(resp));
} catch (TencentCloudSDKException e) {
System.out.println(e.toString());
}
}
}
```

更多示例

您可以在 github 中的 `examples` 目录下获取更多详细的示例。

相关配置

代理配置

从3.0.96版本开始，可以单独设置 HTTP 代理：

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

语言

从3.1.16版本开始，我们添加了对公共参数 Language 的支持，以满足部分产品国际化的诉求。和以前一样，Language 默认不传，通常是中文的，但也有默认英文的。目前可选值为中文（zh-CN）或者英文（en-US），通过如下方法设置：

```
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.Language;
...
ClientProfile clientProfile = new ClientProfile();
clientProfile.setLanguage(Language.ZH_CN);
```

支持 http

SDK 支持 http 协议和 https 协议，通过设置 HttpProfile 的 setProtocol() 方法可以实现协议间的切换：

```
HttpProfile httpProfile = new HttpProfile();
httpProfile.setProtocol("http://"); //http 协议
httpProfile.setProtocol("https://"); //https 协议
```

支持打印日志

自3.1.80版本开始，SDK 支持打印日志。

首先，在创建 CLientProfile 对象时，设置 debug 模式为真,会打印sdk异常信息和流量信息。

```
ClientProfile clientProfile = new ClientProfile();
clientProfile.setDebug(true);
```

腾讯云 java sdk 使用 commons.logging 类进行打印日志，如下所示。

```
九月 10, 2020 5:14:30 下午 com.tencentcloudapi.cvm.v20170312.CvmClient info
信息: send request, request url: https://cvm.ap-shanghai.tencentcloudapi.com/?Non
ce=367595572&Action=DescribeInstances&Filters.0.Values.1=ap-shanghai-2&Version=20
17-03-12&Filters.0.Values.0=ap-shanghai-1&SecretId=AKIDziAMHwiVO0LPCizu61e1iCQP7Y
iaOX7Q&Filters.0.Name=zone&RequestClient=SDK_JAVA_3.1.129&Region=ap-shanghai&Sign
atureMethod=HmacSHA256&Timestamp=1599729270&Signature=DcGRPdquMZZRPj1NFXP5bsOGnRl
aT2KXy7aegNhZa00%3D. request headers information:
九月 10, 2020 5:14:32 下午 com.tencentcloudapi.cvm.v20170312.CvmClient info
信息: recieve response, response url: https://cvm.ap-shanghai.tencentcloudapi.co
m/?Nonce=367595572&Action=DescribeInstances&Filters.0.Values.1=ap-shanghai-2&Vers
ion=2017-03-12&Filters.0.Values.0=ap-shanghai-1&SecretId=AKIDziAMHwiVO0LPCizu61e1
iCQP7YiaOX7Q&Filters.0.Name=zone&RequestClient=SDK_JAVA_3.1.129&Region=ap-shangha
i&SignatureMethod=HmacSHA256&Timestamp=1599729270&Signature=DcGRPdquMZZRPj1NFXP5b
sOGnRlaT2KXy7aegNhZa00%3D, response headers: Server: nginx;Date: Thu, 10 Sep 2020
09:14:32 GMT;Content-Type: application/json;Content-Length: 103;Connection: keep-
alive;OkHttp-Selected-Protocol: http/1.1;OkHttp-Sent-Millis: 1599729271230;OkHttp
-Received-Millis: 1599729272020;,response body information: com.squareup.okhttp.i
nternal.http.RealResponseBody@8646db9
```

用户可以根据自己的需要配置日志打印类，如 log4j，配置方法如下:

- 配置 pom 文件，设置 log4j 版本。

```
<dependency>
<groupId>log4j</groupId>
<artifactId>log4j</artifactId>
<version>1.2.17</version>
</dependency>
```

- 设置环境变量为 log4j，并创建 log 类。

```
System.setProperty("org.apache.commons.logging.Log", "org.apache.commons.loggin
g.impl.Log4JLogger");
Log logger = LogFactory.getLog("TestLog");
logger.info("hello world");
```

旧版 SDK

我们推荐您使用新版 SDK，如果需要旧版 SDK，请在您的 Maven pom.xml 添加以下依赖项即可：

```
<dependency>
<groupId>com.qcloud</groupId>
```

```
<artifactId>qcloud-java-sdk</artifactId>
<version>2.0.6</version>
</dependency>
```

常见问题

展开全部

更新仓库 pom.xml 文件里面的依赖失败

展开&收起

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

展开&收起

`[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，代理的加入可参考上文的代理配置。

版本升级

展开&收起

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。

依赖冲突

展开&收起

目前，SDK 依赖 okhttp 2.5.0，如果和其他依赖 okhttp3 的包混用时，有可能会报错：如 `Exception in thread "main" java.lang.NoSuchMethodError: okio.BufferedSource.rangeEquals(JLokio/ByteString;)Z` 。

原因是 okhttp3 依赖 okio 1.12.0，而 okhttp 依赖 okio 1.6.0，maven 在解析依赖时的规则是路径最短优先和顺序优先，所以如果 SDK 在 pom.xml 依赖中先被声明，则 okio 1.6.0 会被使用，从而报错。

在 SDK 没有升级到 okhttp3 前的解决办法：

1. 在 pom.xml 中明确指定依赖 okio 1.12.0 版本（注意可能有其他包需要用到更高的版本，变通下取最高版本即可）。
2. 将 SDK 放在依赖的最后（注意如果此前已经编译过，则需要先删除掉 maven 缓存的 okhttp 包），以同时使用依赖 okhttp3 的 CMQ SDK 为例，形如（注意变通版本号）：

```
<dependency>
<groupId>com.qcloud</groupId>
<artifactId>cmq-http-client</artifactId>
<version>1.0.7</version>
</dependency>
<dependency>
<groupId>com.tencentcloudapi</groupId>
<artifactId>tencentcloud-sdk-java</artifactId>
<version>3.1.59</version>
</dependency>
```

Go

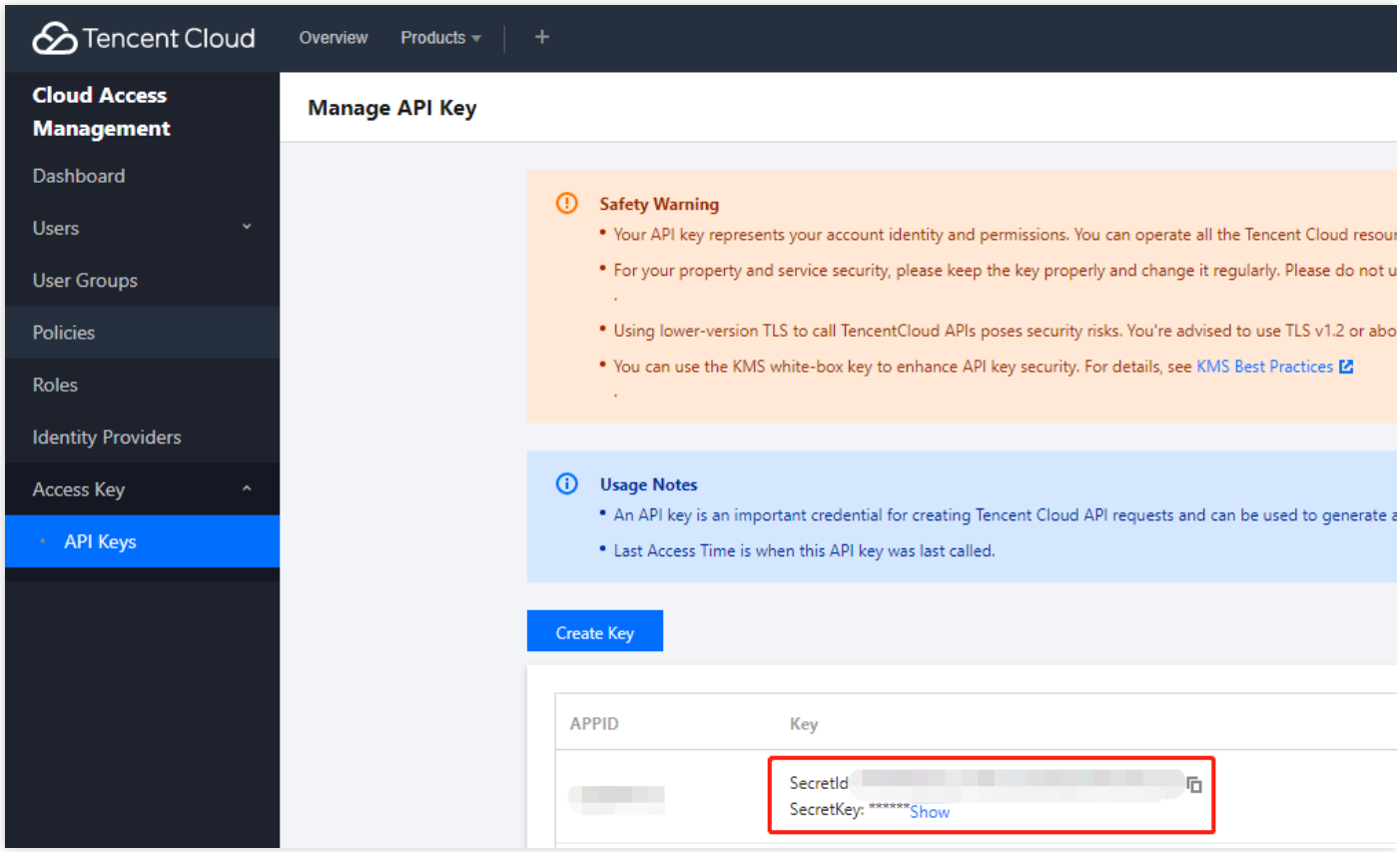
最近更新时间：2021-11-08 14:30:13

简介

- 欢迎使用腾讯云开发者工具套件（SDK）3.0，SDK 3.0 是云 API 3.0 平台的配套工具。SDK 3.0 实现了统一化，各个语言版本的 SDK 具备使用方法相同、接口调用方式相同、错误码和返回包格式相同等优点。
- 本文以 GO SDK 3.0 为例，介绍如何使用、调试并接入腾讯云产品 API。
- 目前已支持云服务器 CVM、私有网络 VPC 、云硬盘 CBS 等 腾讯云产品，后续会支持其他云产品接入。

依赖环境

- Go 1.9 版本及以上（如使用 go mod 需要 Go 1.14），并设置好 GOPATH 等必须的环境变量。
- 获取安全凭证。安全凭证包含 SecretId 及 SecretKey 两部分。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。前往 API 密钥管理 页面，即可进行获取，如下图所示：



> 注意：
> **您的安全凭证代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。**

- 获取调用地址。调用地址（endpoint）一般形式为 `*.tencentcloudapi.com`，产品的调用地址有一定区别，例如，云服务器的调用地址为 `cvm.tencentcloudapi.com`。具体调用地址可参见对应产品的 API 文档。

安装 SDK

方式一、通过 go get 安装（推荐）

推荐使用腾讯云镜像加速下载：

系统平台	运行命令
Linux / MacOS	`export GOPROXY=https://mirrors.tencent.com/go/`
Windows	`set GOPROXY=https://mirrors.tencent.com/go/`

v1.0.170后可以按照产品下载，您只需下载基础包和对应的产品包（如 CVM）即可，不需要下载全部的产品，从而加快您构建镜像或者编译的速度。当然您也可以按照之前的方式一次性下载腾讯云所有产品的包。

> 说明：
> - 按需安装方式：仅支持使用 **Go Modules** 模式进行依赖管理，即环境变量 `GO111MODULE=auto` 或者 `GO111MODULE=on`，并且在您的项目中执行了 `go mod init xxx`。如果您使用 GOPATH，请参见：全部安装方式。
> - 全部安装方式：支持 GOPATH 和 Go Modules。

安装方式	安装说明	运行命令
按需安装（推荐）	安装公共基础包	`go get -v -u github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common`
	安装对应的产品包	

	（如 CVM）	`go get -v -u github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/cvm`
全部安装	一次性下载腾讯云所有产品的包	`go get -v -u github.com/tencentcloud/tencentcloud-sdk-go`

说明：
为了支持 go mod，SDK 版本号从 v3.x 降到了 v1.x。并于2021.05.10移除了所有 `v3.0.*` 和 `3.0.*` 的 tag，如需追溯以前的 tag，请参见项目根目录下的 `commit2tag` 文件。

方式二、通过源码安装

前往代码托管地址 Github 或者 Gitee 下载最新代码，解压后安装到 $GOPATH/src/github.com/tencentcloud 目录下。

使用 SDK

每个接口都有一个对应的 Request 结构和一个 Response 结构。例如，云服务器的查询实例列表接口 DescribeInstances 有对应的请求结构体 DescribeInstancesRequest 和返回结构体 DescribeInstancesResponse 。

下面以云服务器查询实例列表接口为例，介绍 SDK 的基础用法。

- 简化版
- 详细版

```
package main
import (
"fmt"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/regions"
cvm "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/cvm/v20170312"
)
func main() {
credential := common.NewCredential("secretId", "secretKey")
client, _ := cvm.NewClient(credential, regions.Guangzhou, profile.NewClientProfi
```

```
le())
request := cvm.NewDescribeInstancesRequest()
response, err := client.DescribeInstances(request)
if _, ok := err.(*errors.TencentCloudSDKError); ok {
fmt.Printf("An API error has returned: %s", err)
return
}
if err != nil {
panic(err)
}
fmt.Printf("%s\n", response.ToJsonString())
}
```

说明：
出于演示的目的，有一些非必要的代码，例如对默认配置的修改，以尽量展示 SDK 的功能。在实际编写代码使用 SDK 的时候，建议尽量使用默认配置，酌情修改。

更多示例

更多示例参见 examples 目录。对于复杂接口的 Request 初始化例子，可以参见 例一 。对于使用json字符串初始化 Request 的例子，可以参见 例二 。

相关配置

如无特殊需要，建议您使用默认配置。

在创建客户端前，如有需要，您可以通过修改 `profile.ClientProfile` 中字段的值进行一些配置。

```
// 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()
```

具体的配置项说明如下：

请求方式

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"
```

超时时间

SDK有默认的超时时间，如非必要请不要修改默认设置。如有需要请在代码中查阅以获取最新的默认值。
单位：秒

```
cpf.HttpProfile.ReqTimeout = 30
```

指定域名

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，例如云服务器的上海金融区域名： cvm.ap-shanghai-fsi.tencentcloudapi.com

```
cpf.HttpProfile.Endpoint = "cvm.tencentcloudapi.com"
```

签名方式

SDK默认用 `TC3-HMAC-SHA256` 进行签名，它更安全但是会轻微降低性能。

```
cpf.SignMethod = "HmacSHA1"
```

DEBUG模式

DEBUG模式会打印更详细的日志，当您需要进行详细的排查错误时可以开启。
默认为 `false`

```
cpf.Debug = true
```

代理

如果是有代理的环境下，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。

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。

错误处理

从 `v1.0.181` 开始，腾讯云 GO SDK 会将各个产品的返回的错误码定义为常量，您可以直接调用处理，无需手动定义。如果您使用 IDE (如 Goland)进行开发，可以使用他们的代码提示功能直接选择。例如：

```
...//Your other go code
// Handling errors
response, err := client.DescribeInstances(request)
if terr, ok := err.(*errors.TencentCloudSDKError); ok {
code :=terr.GetCode()
if code == cvm.FAILEDOPERATION_ILLEGALTAGKEY{
fmt.Printf("Handling error: FailedOperation.IllegalTagKey,%s", err)
}else if code == cvm.UNAUTHORIZEDOPERATION{
fmt.Printf("Handling error: UnauthorizedOperation,%s", err)
}else{
fmt.Printf("An API error has returned: %s", err)
}
return
}
...
```

同时，各个接口函数的注释部分也列出了此接口可能会返回的错误码，方便您进行处理：

```
// DescribeInstances
// 本接口 (DescribeInstances) 用于查询一个或多个实例的详细信息。
//
//
//
```

```
// * 可以根据实例`ID`、实例名称或者实例计费模式等信息来查询实例的详细信息。过滤信息详细请见过滤器`Filter`。
//
// * 如果参数为空，返回当前用户一定数量（`Limit`所指定的数量，默认为20）的实例。
//
// * 支持查询实例的最新操作（LatestOperation）以及最新操作状态(LatestOperationState)。
//
// 可能返回的错误码:
// FAILEDOPERATION_ILLEGALTAGKEY = "FailedOperation.IllegalTagKey"
// FAILEDOPERATION_ILLEGALTAGVALUE = "FailedOperation.IllegalTagValue"
// FAILEDOPERATION_TAGKEYRESERVED = "FailedOperation.TagKeyReserved"
// INTERNALSERVERERROR = "InternalServerError"
// INVALIDFILTER = "InvalidFilter"
// INVALIDFILTERVALUE_LIMITEXCEEDED = "InvalidFilterValue.LimitExceeded"
// INVALIDHOSTID_MALFORMED = "InvalidHostId.Malformed"
// INVALIDINSTANCEID_MALFORMED = "InvalidInstanceId.Malformed"
// INVALIDPARAMETER = "InvalidParameter"
// INVALIDPARAMETERVALUE = "InvalidParameterValue"
// INVALIDPARAMETERVALUE_IPADDRESSMALFORMED = "InvalidParameterValue.IPAddressMalformed"
// INVALIDPARAMETERVALUE_INVALIDIPFORMAT = "InvalidParameterValue.InvalidIpFormat"
// INVALIDPARAMETERVALUE_INVALIDVAGUENAME = "InvalidParameterValue.InvalidVagueName"
// INVALIDPARAMETERVALUE_LIMITEXCEEDED = "InvalidParameterValue.LimitExceeded"
// INVALIDPARAMETERVALUE_SUBNETIDMALFORMED = "InvalidParameterValue.SubnetIdMalformed"
// INVALIDPARAMETERVALUE_TAGKEYNOTFOUND = "InvalidParameterValue.TagKeyNotFound"
// INVALIDPARAMETERVALUE_VPCIDMALFORMED = "InvalidParameterValue.VpcIdMalformed"
// INVALIDSECURITYGROUPID_NOTFOUND = "InvalidSecurityGroupId.NotFound"
// INVALIDSGID_MALFORMED = "InvalidSgId.Malformed"
// INVALIDZONE_MISMATCHREGION = "InvalidZone.MismatchRegion"
// RESOURCENOTFOUND_HPCCLUSTER = "ResourceNotFound.HpcCluster"
// UNAUTHORIZEDOPERATION_INVALIDTOKEN = "UnauthorizedOperation.InvalidToken"
func (c *Client) DescribeInstances(request *DescribeInstancesRequest) (response *DescribeInstancesResponse, err error){
...
}
```

Common Request

从 `v1.0.189` 开始，腾讯云 GO SDK 支持使用 `泛用型的API调用方式(Common Request)` 进行请求。您只需安装 `common` 包, 即可向任何产品发起调用。

> 说明：
> 您必须明确知道您调用的接口所需参数，否则可能会调用失败。

目前仅支持使用 POST 方式，且签名方法必须使用 签名方法 v3。详细使用请参阅示例：使用 Common Request 进行调用。

请求重试

网络错误重试

当发生临时网络错误或超时时，SDK 可以被配置为自动重试。默认不开启。通过 `ClientProfile` 配置重试次数和重试间隔时间。

> 说明：
> 通过反射检查 `Request` 结构体是否存在 `ClientToken` 字段，存在该字段则认为是幂等请求。幂等请求才会在网络错误时自动重试，非幂等请求会抛出异常，防止请求多次重放造成结果不一致。

```
package main
import (
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/regions"
cvm "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/cvm/v20170312"
)
func main() {
credential := common.NewCredential("secretId", "secretKey")
prof := profile.NewClientProfile()
prof.NetworkFailureMaxRetries = 3 // 定义最大重试次数
prof.NetworkFailureRetryDuration = profile.ExponentialBackoff // 定义重试建个时间
client, _ := cvm.NewClient(credential, regions.Guangzhou, prof)
// ...
}
```

更多用法请参见 测试文件。

限频重试

当发生 API 限频时，SDK 可以被配置为自动重试。默认不开启。通过 `ClientProfile` 配置重试次数和重试间隔时间。

```
package main
import (
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
"github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/regions"
cvm "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/cvm/v20170312"
)
func main() {
credential := common.NewCredential("secretId", "secretKey")
prof := profile.NewClientProfile()
prof.RateLimitExceededMaxRetries = 3 // 定义最大重试次数
prof.RateLimitExceededRetryDuration = profile.ExponentialBackoff // 定义重试建个时
间
client, _ := cvm.NewClient(credential, regions.Guangzhou, prof)
// ...
}
```

幂等标识符

当网络超时重试或限频重试开启时，会自动向请求中注入 `ClientToken` 参数（如果请求存在 `ClientToken` 字段且为空）。当用户手动指定 `ClientToken` 时，会跳过注入流程。

> 说明：
> 注入的 `ClientToken` 在 `100000/s` 并发量以下提供全局唯一性。

常见问题

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。

.NET

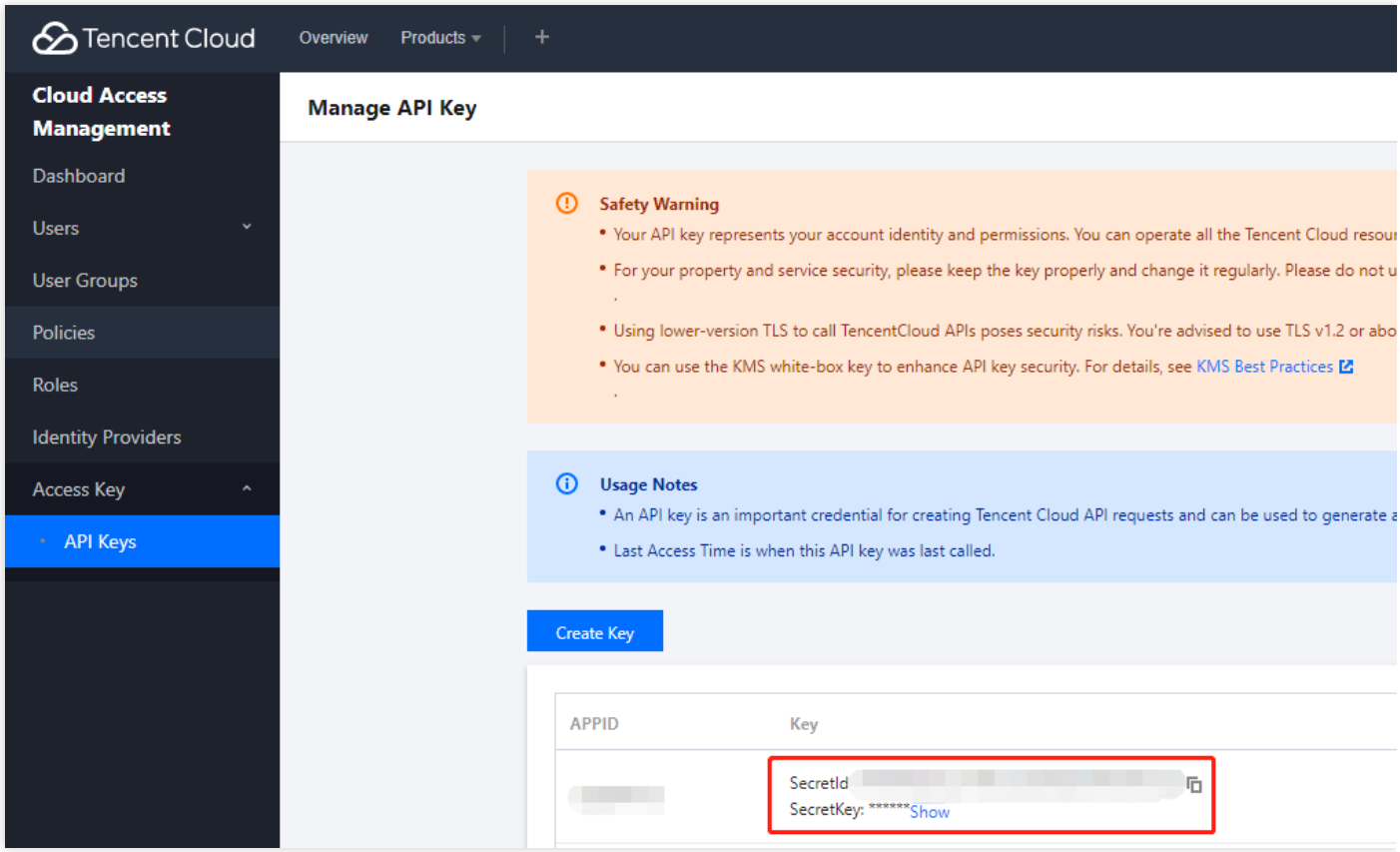
最近更新时间：2021-08-10 16:14:49

简介

- 欢迎使用腾讯云开发者工具套件（SDK）3.0，SDK 3.0 是云 API 3.0 平台的配套工具。SDK 3.0 实现了统一化，各个语言版本的 SDK 具备使用方法相同、接口调用方式相同、错误码和返回包格式相同等优点。
- 本文以 .NET SDK 3.0 为例，介绍如何使用、调试并接入腾讯云产品 API。
- 目前已支持云服务器 CVM、私有网络 VPC 、云硬盘 CBS 等 腾讯云产品，后续会支持其他云产品接入。

依赖环境

- .NET Framework 4.5+ 或者 .NET Core 2.1。
- 获取安全凭证。安全凭证包含 SecretId 及 SecretKey 两部分。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。前往 API 密钥管理 页面，即可进行获取，如下图所示：



> 注意：
> **您的安全凭证代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。**

- 获取调用地址。调用地址（endpoint）一般形式为 `*.tencentcloudapi.com`，产品的调用地址有一定区别，例如，云服务器的调用地址为 `cvm.tencentcloudapi.com`。具体调用地址可参考对应产品的 API 文档。

安装 SDK

方式一、通过 nuget 安装（推荐）

通过命令行安装（**这里的版本仅作为示例，实际请选择最新版本**）：

```
dotnet add package TencentCloudSDK --version 3.0.0
# 其他信息请前往 www.nuget.org/packages/TencentCloudSDK 获取
```

> 注意：
> 命令需要在项目的主目录下执行。

通过 Visual Studio 添加包：
例如，创建一个 HelloWorld 项目：

```
dotnet new console -o HelloWorld
# 进入到 HelloWorld 项目主目录
dotnet run
# 输出为：Hello World!
dotnet add package TencentCloudSDK --version 3.0.0
# 为项目下载 SDK 依赖
```

> 说明：
> - 如果想单独安装某个产品，例如云服务器 CVM，则添加依赖 TencentCloudSDK.Cvm 即可。

方式二、通过源码安装

前往 Github 代码托管地址 下载最新代码，解压后安装到你的工作目录下，使用 Visual Studio 2017 打开编译。

使用 SDK

每个接口都有一个对应的 Request 结构和一个 Response 结构。例如，云服务器的查询实例列表接口 DescribeInstances 有对应的请求结构体 DescribeInstancesRequest 和返回结构体 DescribeInstancesResponse。

下面以云服务器查询实例列表接口为例，介绍 SDK 的基础用法。

- 简化版
- 详细版

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Cvm.V20170312;
using TencentCloud.Cvm.V20170312.Models;
namespace TencentCloudExamples
{
class DescribeInstances
{
static void Main(string[] args)
{
try
{
Credential cred = new Credential {"SecretId", "SecretKey"};
CvmClient client = new CvmClient(cred, "ap-guangzhou");
DescribeInstancesRequest req = new DescribeInstancesRequest();
DescribeInstancesResponse resp = client.DescribeInstancesSync(req);
Console.WriteLine(AbstractModel.ToJsonString(resp));
}
catch (Exception e)
{
Console.WriteLine(e.ToString());
}
}
}
}
```

更多示例

更多示例请参见 github TencentCloudExamples 目录。

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。

相关配置

代理

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。

常见问题

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。

C++

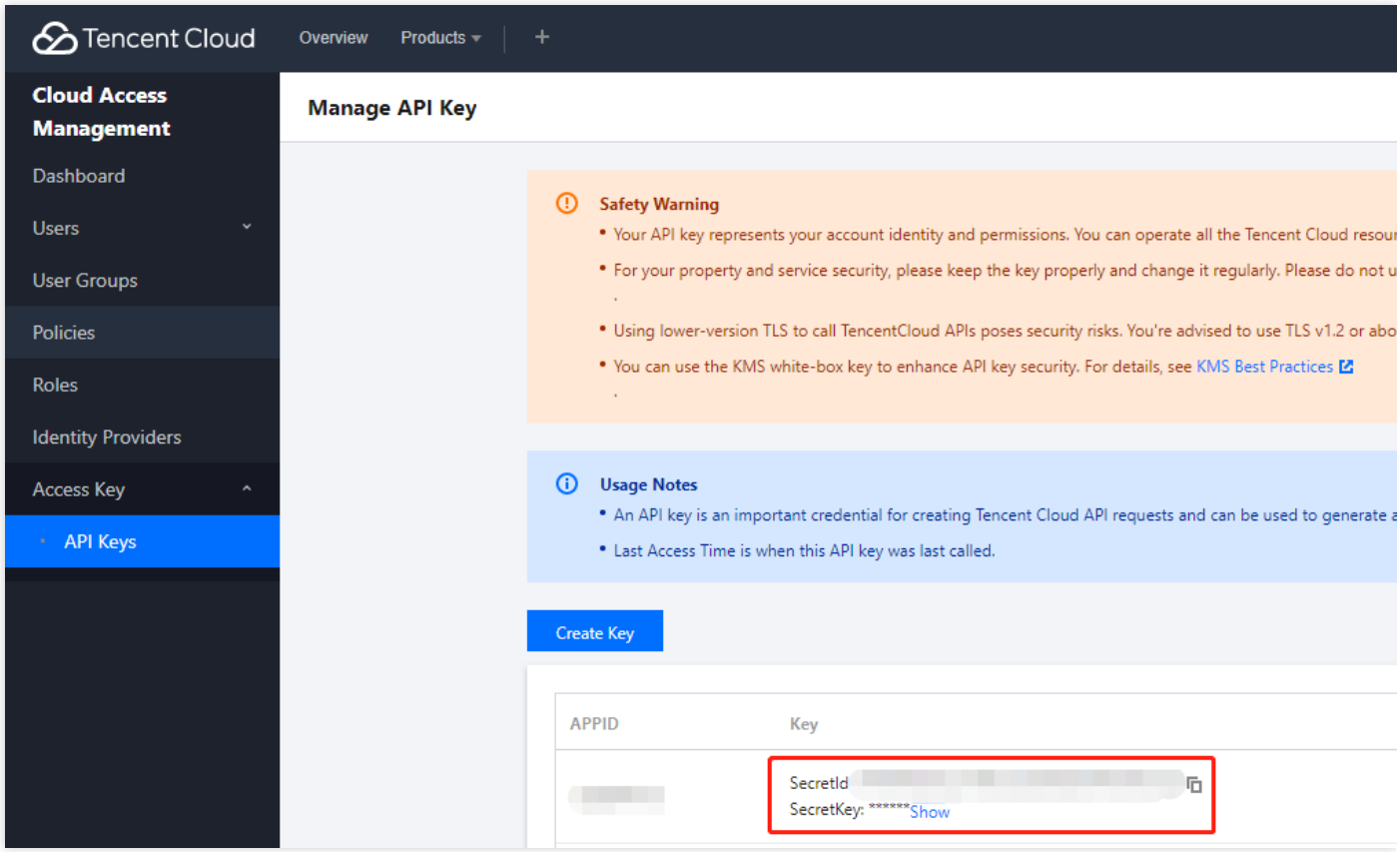
最近更新时间：2021-08-10 16:14:50

简介

- 欢迎使用腾讯云开发者工具套件（SDK）3.0，SDK 3.0 是云 API 3.0 平台的配套工具。SDK 3.0 实现了统一化，各个语言版本的 SDK 具备使用方法相同、接口调用方式相同、错误码和返回包格式相同等优点。
- 本文以 C++ SDK 3.0 为例，介绍如何使用、调试并接入腾讯云产品 API。
- 目前已支持云服务器 CVM、私有网络 VPC 、云硬盘 CBS 等 腾讯云产品，后续会支持其他云产品接入。

依赖环境

- C++ 11 或更高版本的编译器，即 GCC 4.8 或以上版本。暂时仅支持 Linux 安装环境，不支持 Windows 环境。
- 获取安全凭证。安全凭证包含 SecretId 及 SecretKey 两部分。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。前往 API 密钥管理 页面，即可进行获取，如下图所示：



> 注意：
> **您的安全凭证代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。**

- 获取调用地址。调用地址（endpoint）一般形式为 `*.tencentcloudapi.com`，产品的调用地址有一定区别，例如，云服务器的调用地址为 `cvm.tencentcloudapi.com`。具体调用地址可参考对应产品的 API 文档。
- 安装 cmake 编译工具（cmake 3.0 或以上版本），例如：

```
ubuntu
sudo apt-get install cmake
centos
yum install cmake3
```

- 安装依赖库 libcurl，建议安装最新版的 libcurl 库，否则可能存在 libcurl 库内存泄露 bug 问题。

```
ubuntu
sudo apt-get install libcurl4-openssl-dev
centos
yum install libcurl-devel
```

- 安装依赖库 openssl，安装示例如下：

```
ubuntu
sudo apt-get install libssl-dev
centos
yum install openssl-devel
```

- 安装依赖库 libuuid，安装示例如下：

```
ubuntu
sudo apt-get install uuid-dev
centos
yum install libuuid-devel
```

安装 SDK

从源代码构建 SDK

1. 前往 Github 代码托管地址 下载最新代码。
2. 进入 SDK 创建生成必要的构建文件，这里的 `path/to/` 是指 tencentcloud-sdk-cpp 包的实际路径。

```
cd <path/to/tencentcloud-sdk-cpp>
mkdir sdk_build
cd sdk_build
cmake ..
make
sudo make install
```

使用SDK

> 注意：
> 示例不能直接运行，需要将密钥等信息改为真实可用的信息，最好配置在环境变量，避免暴露在代码中。

以 cvm 产品的 DescribeInstances 接口为例：

- 简化版
- 详细版

```
#include <tencentcloud/core/TencentCloud.h>
#include <tencentcloud/core/Credential.h>
#include <tencentcloud/cvm/v20170312/CvmClient.h>
#include <tencentcloud/cvm/v20170312/model/DescribeInstancesRequest.h>
#include <tencentcloud/cvm/v20170312/model/DescribeInstancesResponse.h>
#include <tencentcloud/cvm/v20170312/model/Instance.h>
#include <iostream>
#include <string>
using namespace TencentCloud;
using namespace TencentCloud::Cvm::V20170312;
using namespace TencentCloud::Cvm::V20170312::Model;
using namespace std;
int main()
{
TencentCloud::InitAPI();
string secretId = "<your secret id>";
string secretKey = "<your secret key>";
Credential cred = Credential(secretId, secretKey);
DescribeInstancesRequest req = DescribeInstancesRequest();
CvmClient cvm_client = CvmClient(cred, "ap-guangzhou");
auto outcome = cvm_client.DescribeInstances(req);
if (!outcome.IsSuccess())
{
```

```
cout << outcome.GetError().PrintAll() << endl;
TencentCloud::ShutdownAPI();
return -1;
}
DescribeInstancesResponse rsp = outcome.GetResult();
cout<<"RequestId="<<rsp.GetRequestId()<<endl;
cout<<"TotalCount="<<rsp.GetTotalCount()<<endl;
if (rsp.InstanceSetHasBeenSet())
{
vector<Instance> instanceSet = rsp.GetInstanceSet();
for (auto itr=instanceSet.begin(); itr!=instanceSet.end(); ++itr)
{
cout<<(*itr).GetPlacement().GetZone()<<endl;
}
}
TencentCloud::ShutdownAPI();
return 0;
}
```

Demo 用例编译执行：

```
cd example/cvm/v20170312
mkdir build
cd build
cmake ..
make
./DescribeInstances
```

如果报错动态库找不到，可指定动态库路径。例如，libtencentcloud-sdk-cpp-core.so 安装到了 /usr/local/lib 路径下：

```
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
./DescribeInstances
```

> 说明：
> 更多示例请参考 example 目录。

单元测试

依赖库 gtest

安装示例如下：

```
git clone https://github.com/google/googletest
cd googletest
cmake CMakeLists.txt
make
```

将生成的 libgtest.a libgtest_main.a 静态库，以及 gtest 的头文件，拷贝到系统目录下。

配置环境变量

- ENV_SecretId: 密钥 ID
- ENV_SecretKey: 密钥 Key

测试

执行以下脚本

```
sh function_test.sh
```

相关配置

代理

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。

Ruby

最近更新时间：2021-08-10 16:14:50

简介

欢迎使用腾讯云开发者工具套件（SDK），本 SDK 是云 API 3.0 平台的配套工具。

依赖环境

1. 依赖环境：Ruby 2.3 及以上版本。
2. 从 腾讯云控制台 开通相应产品。
3. 获取 SecretId、SecretKey 以及调用地址（endpoint），endpoint 一般形式为*.tencentcloudapi.com，如 CVM 的调用地址为 cvm.tencentcloudapi.com，具体参考各产品说明。

获取安装

安装 Ruby SDK 前，先获取安全凭证。在第一次使用云 API 之前，用户首先需要在 腾讯云控制台 上申请安全凭证，安全凭证包括 SecretId 和 SecretKey。

- SecretId 是用于标识 API 调用者的身份。
- SecretKey 是用于加密签名字符串和服务器端验证签名字符串的密钥。

> 注意：
> SecretKey 必须严格保管，避免泄露。

通过源码包安装

前往 Github 代码托管地址 下载最新代码，以安装 cvm sdk 为例，解压后：

```
$ cd tencentcloud-sdk-ruby/tencentcloud
$ cd tencentcloud-sdk-common
$ gem build tencentcloud-sdk-common.gemspec
$ gem install tencentcloud-sdk-common-1.0.0.gem
$ cd ../tencentcloud-sdk-cvm
$ gem build tencentcloud-sdk-cvm.gemspec
$ gem install tencentcloud-sdk-cvm-1.0.0.gem
```

注意：
上述版本号请以实际为准。

示例

以查询实例列表接口为例。

简化版

```
# -*- coding: UTF-8 -*-
require 'tencentcloud-sdk-common'
require 'tencentcloud-sdk-cvm'
include TencentCloud::Common
include TencentCloud::Cvm::V20170312
begin
cre = Credential.new('SecretId', 'SecretKey')
req = DescribeInstancesRequest.new(nil, nil, 0, 1)

cli = Client.new(cre, 'ap-guangzhou')
cli.DescribeInstances(req)
rescue TencentCloudSDKException => e
puts e.message
puts e.backtrace.inspect
end
```

详细版

```
# -*- coding: UTF-8 -*-
require 'tencentcloud-sdk-common'
require 'tencentcloud-sdk-cvm'
begin
include TencentCloud::Common
# 导入对应产品模块的client module
include TencentCloud::Cvm::V20170312
# 实例化一个认证对象，入参需要传入腾讯云账户secretId，secretKey,此处还需注意密钥对的保密
cred = Credential.new('SecretId', 'SecretKey')
# 实例化一个http选项
httpProfile = HttpProfile.new()
# 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.new(proxy='http://用户名:密码@代理IP:代理端口')
```

```
httpProfile.scheme = "https" # 在外网互通的网络环境下支持http协议(默认是https协议),建议使用https协议
httpProfile.req_method = "GET" # get请求(默认为post请求)
httpProfile.req_timeout = 30 # 请求超时时间，单位为秒(默认60秒)
httpProfile.endpoint = "cvm.tencentcloudapi.com" # 指定接入地域域名(默认就近接入)
# 实例化一个client选项，可选的，没有特殊需求可以跳过。
clientProfile = ClientProfile.new()
clientProfile.sign_method = "TC3-HMAC-SHA256" # 指定签名算法
clientProfile.language = "en-US" # 指定展示英文（默认为中文）
clientProfile.http_profile = httpProfile
clientProfile.debug = true # 打印debug日志
# 实例化要请求产品(以cvm为例)的client对象，clientProfile是可选的。
client = Client.new(cred, "ap-shanghai", clientProfile)
# 实例化一个cvm实例信息查询请求对象,每个接口都会对应一个request对象。
req = DescribeInstancesRequest.new()
# 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。
respFilter = Filter.new() # 创建Filter对象，以zone的维度来查询cvm实例。
respFilter.Name = "zone"
respFilter.Values = ["ap-shanghai-1", "ap-shanghai-2"]
req.Filters = [respFilter] # Filters 是成员为Filter对象的列表
# 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.DescribeInstances(req)
# 输出json格式的字符串回包
puts resp.serialize
# 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。
puts resp.TotalCount
rescue TencentCloudSDKException => e
puts e.message
puts e.backtrace.inspect
end
```

相关配置

代理

如果是有代理的环境下，可通过两种方式设置代理：

1. 在初始化HttpProfile时指定 `proxy`
2. 需要设置系统环境变量 `https_proxy`

注意：
否则可能无法正常调用，抛出连接超时的异常。