

Serverless Cloud Function

SDK Documentation

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

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

Trademark Notice



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

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

SDK Documentation

Python

Node.js

PHP

Java

Go

.NET

C++

Ruby

SDK Documentation

Python

Last updated : 2021-11-11 10:49:28

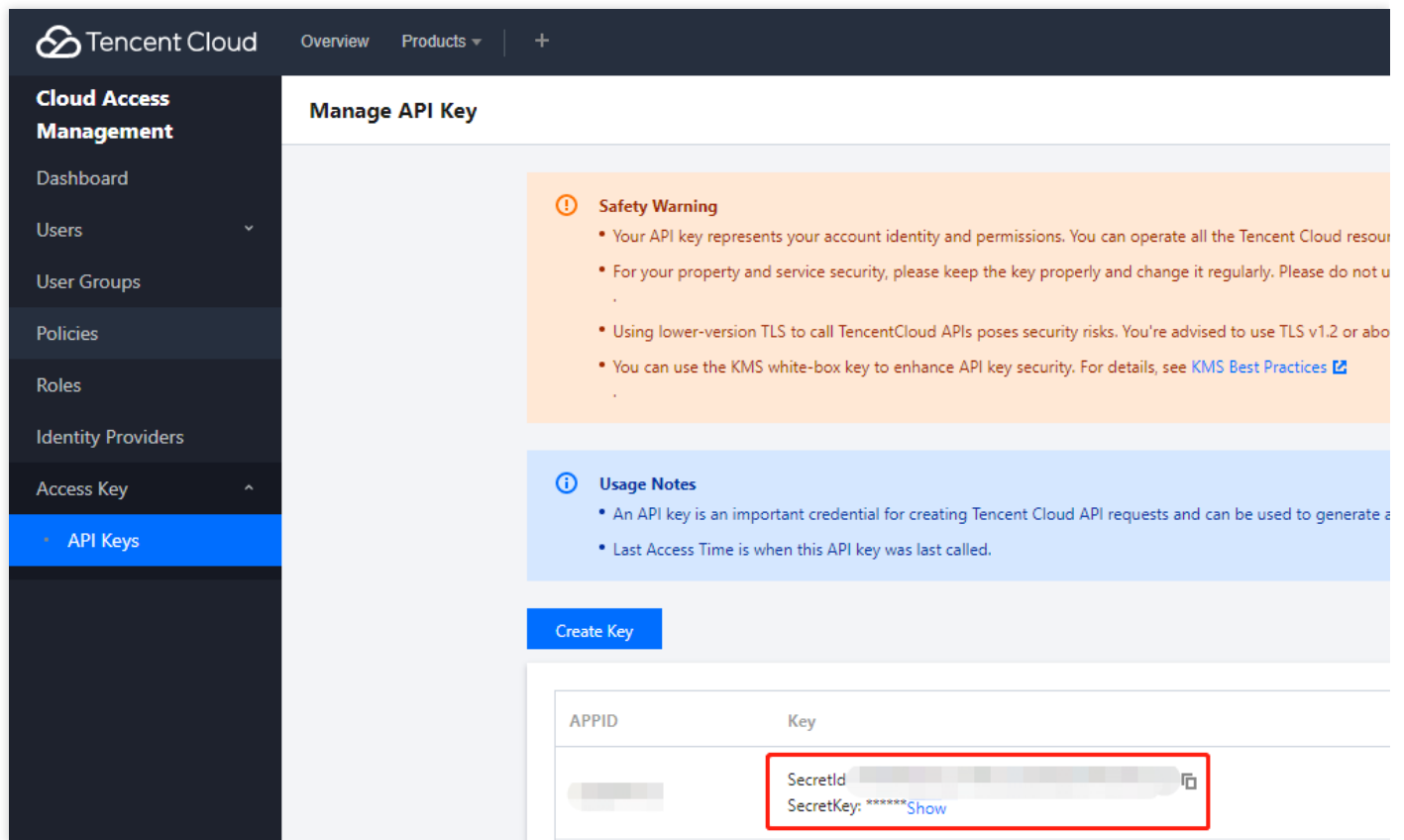
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform. SDK 3.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document describes how to use, debug, and connect to TencentCloud APIs with the SDK for Python 3.0 as an example.
- This version currently supports various Tencent Cloud products such as CVM, VPC, and CBS and will support more products in the future.

Dependent Environment

- Python 2.7 and 3.6–3.9.
- Get the security credential, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose it to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by product. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding product .

Installing SDK

Method 1. Install through pip (recommended)

You can install the Tencent Cloud SDK for Python into your project through pip. If you haven't installed pip in your project environment yet, install it first as instructed in [Installation](#).

Run the following command on the command line to install the SDK for Python.

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

Note :

If you have both Python 2 and Python 3 environments, you need to use the pip3 command to install.

Users in the Chinese mainland can use a Tencent Cloud mirror source to speed up the download by running `pip install -i https://mirrors.tencent.com/pypi/simple/ --upgrade tencentcloud-sdk-python` for example.

Note :

- If you only want to use the package of a specific product, such as CVM, you can install it separately, but this method cannot work together with the full installation method. For example, run `pip install --upgrade tencentcloud-sdk-python-common tencentcloud-sdk-python-cvm`.

Method 2. Install through source package

Go to the [GitHub code hosting page](#) to download the latest code, decompress it, and run the following command:

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

Using SDK

The following takes the instance list querying API as an example.

- [Simplified](#)
- [Detailed](#)

```
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
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)
```

Common Client Call Method

Starting from `v3.0.396`, Tencent Cloud SDK for Python supports the use of `Common Client` mode for requests. You only need to install the `tencentcloud-sdk-python-common` package to initiate calls to any Tencent Cloud product.

Note :

You must clearly know the parameters required by the called API; otherwise, the call may fail.

For more information on `Common Client`, please see [example](#).

More Examples

You can find more detailed samples in the `examples` directory in the [GitHub repository](#).

Relevant Configuration

Proxy

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

- Specify the proxy when initializing `HttpProfile`. For more information, please see the [example](#).
- You need to set the system environment variable `https_proxy`.

Otherwise, it may not be called normally, and a connection timeout exception will be thrown.

FAQs

Certificate issue

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

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

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

Node.js

Last updated : 2021-08-11 10:24:57

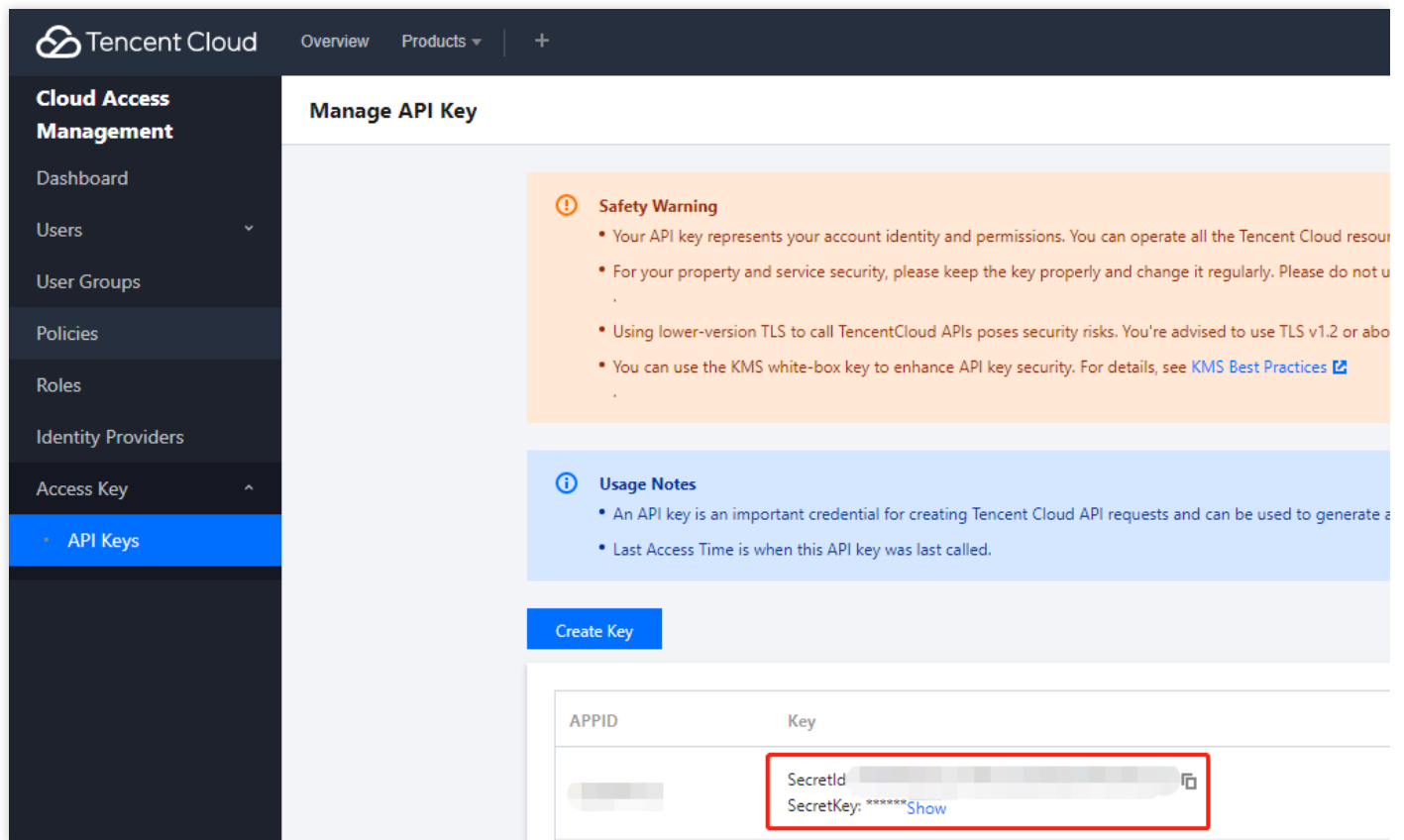
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) for Node.js 4.0, a companion tool for the TencentCloud API 3.0 platform. The new SDK 4.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document takes SDK for Node.js 4.0 as an example to describe how to use the SDK with simple samples. It makes it easier for Node.js developers to debug and access Tencent Cloud product APIs.
- This version currently supports various Tencent Cloud services such as CVM, VPC, and CBS and will support more services in the future.

Dependent Environment

- Node.js v10.0.0 and above.
- Activate your product in the [Tencent Cloud console](#).
- Get the security credential, which consists of `SecretID` and `SecretKey`. `SecretID` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose your `SecretKey` to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by product. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding product.

Installing SDK

Method 1. Install through npm

Installation through npm is the recommended way to use the SDK for Node.js. npm is a dependency manager for Node.js that supports the dependencies your project requires and installs them into your project. For more information, please visit [npm official website](#).

1. Run the following installation command:

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

2. Import the corresponding module code into your code. For more information, please see the SDK sample below.

Note :

- The above import method downloads the SDKs of all Tencent Cloud products to your local system. You can replace `tencentcloud-sdk-nodejs` with a specific product SDK name such as `tencentcloud-sdk-nodejs-cvm/cbs/vpc` to import the SDK of the specific product. In the code, you can change `require("tencentcloud-sdk-nodejs")` to `require("tencentcloud-sdk-nodejs-cvm/cbs/vpc")` and keep the rest unchanged, which can greatly save the storage space. For more information, please see the sample.

Method 2. Install through source package

1. Go to the [GitHub code hosting page](#) to download the source code package.
2. Decompress the source package to an appropriate location in your project.
3. Import the corresponding module code into your code. For more information, please see the SDK sample below.

Using SDK

Sample

```
const tencentcloud = require("tencentcloud-sdk-nodejs")
// Import the client models of the corresponding product module
const CvmClient = tencentcloud.cvm.v20170312.Client
const clientConfig = {
  // Tencent Cloud authentication information
  credential: {
    secretId: "secretId",
    secretKey: "secretKey",
  },
  // Product region
  region: "ap-shanghai",
  // Optional instance configuration
  profile: {
    signMethod: "HmacSHA256", // Signature algorithm
    httpProfile: {
      reqMethod: "POST", // Request method
    },
  },
}
```

```
reqTimeout: 30, // Request timeout period in seconds, which is 60s by default
},
},
}
// Instantiate the client object of the requested product (with CVM as an example)
const client = new CvmClient(clientConfig)
// Call the API you want to access through the client object; you need to pass in
the request object and the response callback function
client.DescribeZones().then(
  (data) => {
    console.log(data)
  },
  (err) => {
    console.error("error", err)
  }
)
```

Note :

During execution, you need to replace `secretId` and `secretKey` with real values.

In projects that support typescript, use the following method to call:

```
import * as tencentcloud from "tencentcloud-sdk-nodejs"
// Import the client models of the corresponding product module
const CvmClient = tencentcloud.cvm.v20170312.Client
const clientConfig = {
  // Tencent Cloud authentication information
  credential: {
    secretId: "secretId",
    secretKey: "secretKey",
  },
  // Product region
  region: "ap-shanghai",
  // Optional instance configuration
  profile: {
    signMethod: "HmacSHA256", // Signature algorithm
    httpProfile: {
      reqMethod: "POST", // Request method
      reqTimeout: 30, // Request timeout period in seconds, which is 60s by default
    },
  },
}
```

```
// Instantiate the client object of the requested product (with CVM as an example)
const client = new CvmClient(clientConfig)
// Call the API you want to access through the client object; you need to pass in
// the request object and the response callback function
client.DescribeZones().then(
  (data) => {
    console.log(data)
  },
  (err) => {
    console.error("error", err)
  }
)
```

The input parameters for instantiating `Client` support the `clientConfig` data structure. For more information, please see [ClientConfig](#).

More samples

For more demos, please go to the `examples` directory.

Relevant Configuration

Proxy

If there is a proxy in your environment, you need to set the system environment variable `https_proxy`; otherwise, it may not be called normally, and a connection timeout exception will be thrown.

Legacy SDK

We recommend you use the new version of the SDK for Node.js. If you have to use a legacy SDK, please go to the [GitHub repository](#).

PHP

Last updated : 2021-08-10 16:14:49

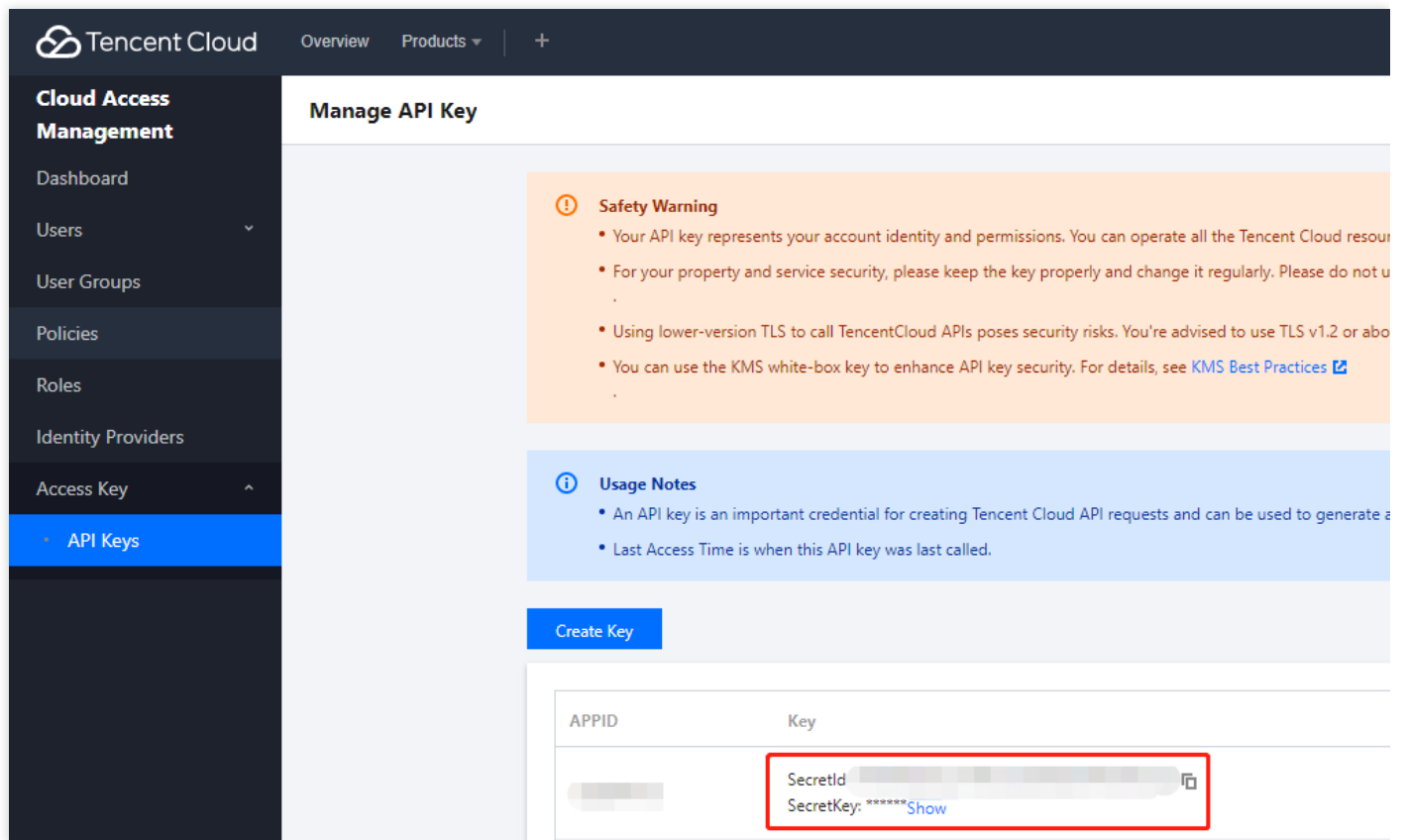
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform. SDK 3.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document describes how to use, debug, and connect to TencentCloud APIs with the SDK for PHP 3.0 as an example.
- This version currently supports various Tencent Cloud products such as CVM, VPC, and CBS and will support more products in the future.

Dependent Environment

- PHP 5.6.33 or above.
- Get the security credential, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose it to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by product. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding product .

Installing SDK for PHP 3.0

Installation through Composer is the recommended way to use the SDK for PHP. Composer is a dependency manager for PHP. For more information, please visit [Composer official website](#).

Note :

Composer requires PHP 5.3.2+ and above, and `openssl` needs to be enabled.

Step 1. Install Composer

- For Windows, go to [Composer official website](#) to download the installation package and install it.
- For Unix, install it by running the following command on the command line:

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

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

Step 2. Add a mirror source

If you are in the Chinese mainland, you can use a Tencent Cloud mirror source to speed up the download by running the following command in the opened command window:

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

Step 3. Add dependencies

In the opened command window, run the command to install the SDK (in the specified location). For example, to install in the `C:\Users\...>` directory, open the command window at the specified location and run the following command:

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

Step 4. Add references

Import the following code. Note: this example is for reference only. Composer will generate a `vendor` directory in the project root directory, whose actual absolute path is `/path/to/` (if you perform this operation in the project root directory, you can omit the absolute path).

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

Note :

- If you only want to install the package of a certain product, you can use `composer require tencentcloud/product name`, such as `composer require tencentcloud/cvm`.

Using SDK

- We recommend you use [API 3.0 Explorer](#), which provides various capabilities such as online call, signature verification, SDK code generation, and quick API search, greatly improving the ease of use of TencentCloud API 3.0 and SDK.
- You can also refer to the examples in the [examples](#) directory in the SDK repository for more usage information.

The following uses the instance querying API `DescribeInstances` as an example:

- [Simplified](#)
- [Detailed](#)

```
<?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;
}
```

More samples

You can find more detailed samples in the `examples` directory in the [GitHub repository](#).

Relevant Configuration

Proxy

If there is a proxy in your environment, you need to set the system environment variable `https_proxy`; otherwise, it may not be called normally, and a connection timeout exception will be thrown. You can also use GuzzleHttp to proxy the configuration:

```
$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);
```

FAQs

Show All

Certificate issue

展开&收起

If there is a problem with your PHP environment certificate, errors similar to `cURL error 60: See http://curl.haxx.se/libcurl/c/libcurl-errors.html` may occur, which can be solved as follows:

1. Download the certificate file `cacert.pem` at <https://curl.haxx.se/ca/cacert.pem> and save it to the PHP installation path.
2. Edit the `php.ini` file: delete the semicolon comment (;) before the `curl.cainfo` configuration item and set the value to the absolute path of the saved certificate file `cacert.pem`.
3. Restart the services that depend on PHP.

php_curl extension

展开&收起

GuzzleHttp, which this SDK depends on, needs to have the `php_curl` extension enabled. Check whether the `php.ini` environment in your environment is enabled. For example, on Linux with PHP 7.1, for services hosted under Apache, you can open `/etc/php/7.1/apache2/php.ini` to see whether the `extension=php_curl.dll` configuration item has been commented. Please delete the comment before it and restart Apache.

Web access exception

展开&收起

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

```
cURL error 0: The cURL request was retried 3 times and did not succeed. The most likely reason for the failure is that cURL was unable to rewind the body of the request and subsequent retries resulted in the same error. Turn on the debug option
```

to see what went wrong. See <https://bugs.php.net/bug.php?id=47204> for more information. (see <http://curl.haxx.se/libcurl/c/libcurl-errors.html>)

This error may occur in different cases. You can run `php -r "echo sys_get_temp_dir();"` to print the absolute path of the default system temporary directory and set `sys_temp_dir` in `php.ini` to this value, and then check whether this error is fixed.

Problem with installation through source code

展开&收起

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

Java

Last updated : 2021-10-12 18:06:58

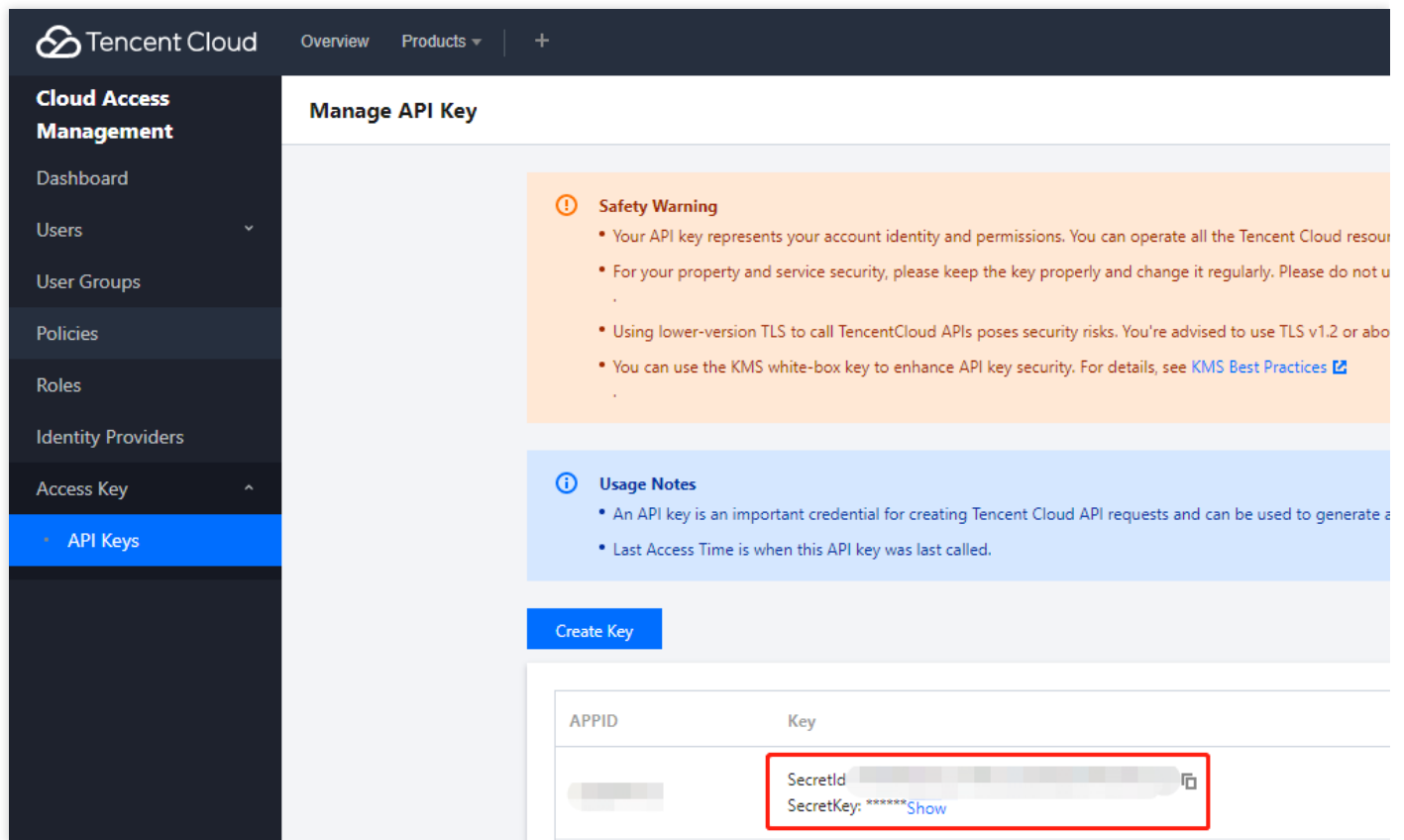
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform. SDK 3.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document describes how to use, debug, and connect to TencentCloud APIs with the SDK for Java 3.0 as an example.
- This version currently supports various Tencent Cloud services such as CVM, VPC, and CBS and will support more services in the future.

Dependent Environment

- JDK 7 and above.
- Get the security credential, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose it to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by service. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding service.

Installing SDK

Method 1. Install through Maven (recommended)

Maven is a dependency management tool for Java that supports the dependencies your project requires and installs them into your project.

1. Go to [Maven official website](#) to download the corresponding Maven installation package for your system and install it. For more information on Maven, please see [Welcome to Apache Maven](#).
2. Add Maven dependencies for your project by adding the following dependencies under the `<dependencies>` tag in Maven's `pom.xml`. You can find the latest version (v3.1.322) in the [Maven repository](#).

```
<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. -->
<!-- Please query the latest version at https://search.maven.org/search?q=tence
ntcloud-sdk-java, which is as follows -->
<version>3.1.322</version>
</dependency>
```

Note :

- The version number here is just an example, and you can view the latest version number in the [Maven repository](#).
 - v4.0.11 shown in the [Maven repository](#) was disused but has not been completely deleted due to the Maven index update issue.
 - The above import method downloads the SDKs of all Tencent Cloud products to your local system. You can replace the `artifactId` with a specific product SDK name such as `tencentcloud-sdk-java-cvm/cbs/vpc` to import the SDK of the specific product. The code can be used in the same way, and the major packages are the same. For more information, please see the samples. The latest version can also be queried in the [Maven repository](#), which can greatly save the storage space.
3. Set the mirror source to speed up the download. To do so, edit the `settings.xml` configuration file of Maven and add the mirror configuration in the `mirrors` section:

```
<mirror>
<id>tencent</id>
<name>tencent maven mirror</name>
<url>https://mirrors.tencent.com/nexus/repository/maven-public/</url>
<mirrorOf>*</mirrorOf>
</mirror>
```

Method 2. Install through source package

1. Go to the [GitHub code hosting page](#) to download the source code package.
2. Decompress the source package to an appropriate location in your project.
3. You need to put the jar package under the `vendor` directory in a path that can be found by Java.
4. For importing methods, please see the sample.

Using SDK

The following uses the instance querying API `DescribeInstances` as an example:

- [Simplified](#)
- [Detailed](#)

```
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());
        }
    }
}
```

More samples

You can find more detailed samples in the `examples` directory on the [GitHub page](#).

Relevant Configuration

Proxy

Starting from v3.0.96, an HTTP proxy can be set separately:

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

Language

Starting from v3.1.16, we add the support for the common parameter `Language` to meet the globalization requirements of certain products. As before, `Language` is not passed in by default subject to the decisions of each service API, which is usually Chinese but can also be English by default in some products. Currently, valid values are Chinese and English, which can be set in the following way:

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

Support for HTTP

The SDK supports both HTTP and HTTPS protocols, and the switch between protocols can be implemented by setting the `setProtocol()` method in `HttpProfile`:

```
HttpProfile httpProfile = new HttpProfile();
httpProfile.setProtocol("http://"); // HTTP protocol
httpProfile.setProtocol("https://"); // HTTPS protocol
```

Support for Log Printout

Starting from v3.1.80, the SDK supports printing logs.

First, when creating the `ClientProfile` object, set the `debug` mode to true, and the SDK exception information and traffic information will be printed.

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

Tencent Cloud SDK for Java uses the `commons.logging` class to print logs as shown below:

```
September 10, 2020 5:14:30 PM com.tencentcloudapi.cvm.v20170312.CvmClient info
Information: send request, request url: https://cvm.ap-shanghai.tencentcloudapi.com/?Nonce=367595572&Action=DescribeInstances&Filters.0.Values.1=ap-shanghai-2&Ver
```



```
sion=2017-03-12&Filters.0.Values.0=ap-shanghai-1&SecretId=AKIDziAMHwiVO0LPCizu61e1iCQP7YiaOX7Q&Filters.0.Name=zone&RequestClient=SDK_JAVA_3.1.129&Region=ap-shanghai&SignatureMethod=HmacSHA256&Timestamp=1599729270&Signature=DcGRPdquMZZRPj1NFXP5bsOGnRlaT2KXy7aegNhZa00%3D. request headers information:
September 10, 2020 5:14:32 PM com.tencentcloudapi.cvm.v20170312.CvmClient info
Information: receive response, response url: https://cvm.ap-shanghai.tencentcloudapi.com/?Nonce=367595572&Action=DescribeInstances&Filters.0.Values.1=ap-shanghai-2&Version=2017-03-12&Filters.0.Values.0=ap-shanghai-1&SecretId=AKIDziAMHwiVO0LPCizu61e1iCQP7YiaOX7Q&Filters.0.Name=zone&RequestClient=SDK_JAVA_3.1.129&Region=ap-shanghai&SignatureMethod=HmacSHA256&Timestamp=1599729270&Signature=DcGRPdquMZZRPj1NFXP5bsOGnRlaT2KXy7aegNhZa00%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.internal.http.RealResponseBody@8646db9
```

You can configure the log printout class as needed, such as `log4j`, in the following way:

- Configure the `pom` file and set the `log4j` version.

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

- Set the environment variable to `log4j` and create a log class.

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

Legacy SDK

We recommend you use the new version of SDK. If you need a legacy version, add the following dependencies to your Maven's `pom.xml`:

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

FAQs

Version upgrade

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

Dependency conflict

Currently, the SDK depends on OkHttp 2.5.0. If it is mixed with other packages that depend on OkHttp 3, an error may be reported, such as `Exception in thread "main" java.lang.NoSuchMethodError: okio.BufferedSource.rangeEquals(JLokio/ByteString;)Z`. The reason is that OkHttp 3 depends on Okio 1.12.0, while OkHttp 2.5.0 depends on Okio 1.6.0. When Maven parses dependencies, it gives top priority to the shortest path in sequence, so if the SDK is declared in the pom.xml dependency first, Okio 1.6.0 will be used, and an error will be reported.

Solution before the SDK is upgraded to OkHttp 3:

1. Clearly specify the dependency on Okio 1.12.0 in pom.xml:

Note :

There may be other packages that require a higher version, in which case, take the highest compatible version as a workaround; for example, when other packages use OkHttp 4, the corresponding version may be Okio 2.2.2.

```
<dependency>
<groupId>com.squareup.okio</groupId>
<artifactId>okio</artifactId>
<version>1.12.0</version>
</dependency>
```

2. Put the SDK at the end of the dependency (note that if it has been compiled before, you need to delete the OkHttp package cached by Maven first). Taking the CMQ SDK that depends on OkHttp 3 as an example, the format is as follows (pay attention to the workaround version number):

```
<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>
```

Certificate issue

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

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

Common Client

Starting from v3.1.303, Tencent Cloud SDK for Java supports the use of `Common Client` mode for requests. You only need to install the `Common` package to initiate calls to any Tencent Cloud product.

Note :

You must clearly know the parameters required by the called API; otherwise, the call may fail.

Currently, only the POST method is supported, and the signature algorithm must be signature algorithm v3. For detailed usage, please see the [Using Common Client to Call](#) sample.

Support for Request Retry

Starting from v3.1.310, Tencent Cloud SDK for Java supports request retry. For each request, you can set the number of retries between 0 and 10. If an API request fails, it will be retried at intervals of 1s until it succeeds or the number of retries is used up.

For detailed usage, please see the [Using `retry` to Retry Request](#) sample.

Credential Management

Tencent Cloud SDK for Java currently supports the following methods for credential management:

Environment variable

By default, the environment variables `TENCENTCLOUD_SECRET_ID` and `TENCENTCLOUD_SECRET_KEY` are read to get `secretId` and `secretKey`. The relevant code is as follows:

```
Credential cred = new EnvironmentVariableCredentialsProvider().getCredentials();
```

Configuration file

The configuration file path should be as follows:

- Windows: `c:\Users\NAME\.tencentcloud\credentials`
- Linux: `~/.tencentcloud/credentials` or `/etc/tencentcloud/credentials`

The configuration file is in the following format:

```
[default]
secret_id = xxxxx
secret_key = xxxxx
```

The relevant code is as follows:

```
Credential cred = new ProfileCredentialsProvider().getCredentials();
```

Role assumption

For related concepts of role assumption, please see [Role Overview](#).

To use this method, you must create a role in the Tencent Cloud CAM console as instructed in [Creating a Role](#).

After you have a role, you can get temporary credentials through the persistent key and `roleArn`, and the SDK will automatically refresh the temporary credentials. The relevant code is as follows:

```
Credential cred = new STSCredential("secretId", "secretKey", "roleArn", "roleSessionName");
```

Instance role

For related concepts of instance role, please see [Managing Roles](#).

After you bind a role to an instance, you can access the relevant metadata APIs on the instance to get temporary credentials, and the SDK will automatically refresh the temporary credentials. The relevant code is as follows:

```
Credential cred = new CvmRoleCredential();
```

Credential supply chain

Tencent Cloud SDK for Java provides a credential supply chain. By default, it will try to get credentials in the order of environment variable -> configuration file -> instance role and return the first obtained credential. The relevant code is as follows:

```
Credential cred = new DefaultCredentialsProvider().getCredentials();
```

For detailed usage of credential management, please see the [Using Credential Supply Chain](#) sample.

Go

Last updated : 2021-11-08 14:31:54

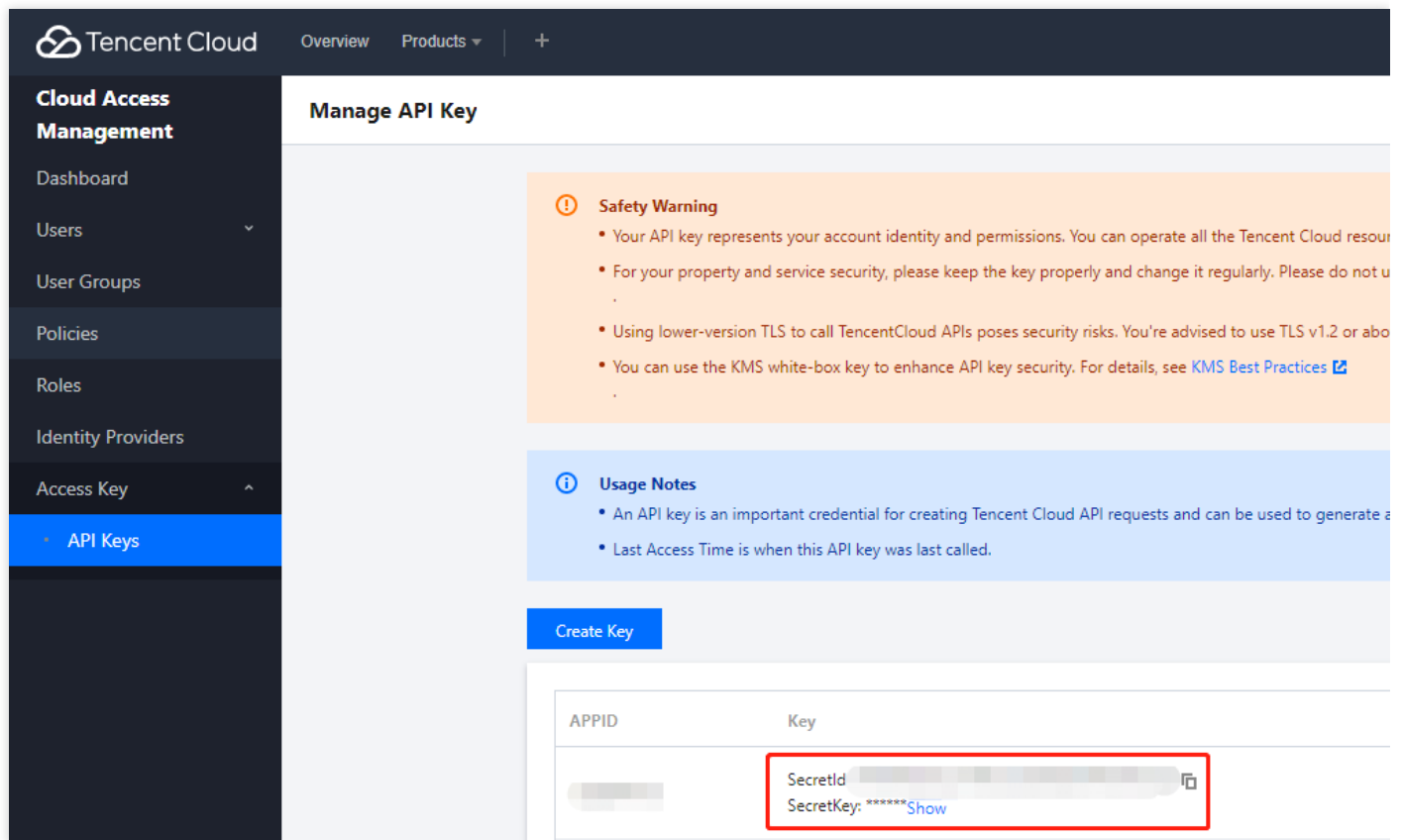
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform. SDK 3.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document describes how to use, debug, and connect to TencentCloud APIs with the SDK for Go 3.0 as an example.
- This version currently supports various Tencent Cloud products such as CVM, VPC, and CBS and will support more products in the future.

Dependent Environment

- Go 1.9 or above (Go 1.14 is required if `go mod` is used). Plus, the necessary environment variables such as `GOPATH` should be set properly.
- Get the security credential, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose it to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by product. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding product.

Installing SDK

Method 1. Install through go get (recommended)

We recommend you use a Tencent Cloud mirror for faster download:

OS	Command
Linux/macOS	

	<pre>export GOPROXY=https://mirrors.tencent.com/go/</pre>
Windows	<pre>set GOPROXY=https://mirrors.tencent.com/go/</pre>

Starting from v1.0.170, you can download packages by product. You only need to download the basic package and the corresponding product package (such as CVM) instead of downloading the packages of all Tencent Cloud products, which speeds up the image build and compilation. Of course, you can also download the packages of all products at once in the same way as before.

Note :

- On-Demand installation method: you can only use the **Go Modules** mode for dependency management, that is, the environment variable `GO111MODULE` should be `auto` or `on`, and `go mod init xxx` should be executed in your project . If you use `GOPATH`, please see the [full installation method](#).
- Full installation method: it supports both `GOPATH` and `Go Modules`.

Installation Method	Description	Command
On-Demand installation (recommended)	Install the common basic package	<pre>go get -v -u github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common</pre>
	Install the corresponding product package (such as CVM)	<pre>go get -v -u github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/cvm</pre>
Full installation	Download the packages of all Tencent Cloud products at once	<pre>go get -v -u github.com/tencentcloud/tencentcloud-sdk-go</pre>

Note :

In order to support `go mod`, the SDK version number has been reduced from v3.x to v1.x, and all tags of `v3.0.*` and `3.0.*` were removed on May 10, 2021. If you need to backtrack previous tags, please refer to the `commit2tag` file in the root directory of the project.

Method 2. Install through source code

Go to the [GitHub](#) or [Gitee](#) code hosting page to download the latest code, decompress, and install it in the `$GOPATH/src/github.com/tencentcloud` directory.

Using SDK

Each API has a corresponding request structure and a response structure. For example, the

`DescribeInstances` API for querying the CVM instance list has a request structure `DescribeInstancesRequest` and a response structure `DescribeInstancesResponse`.

The following uses the CVM instance list querying API as an example to describe the basic usage of the SDK.

- [Simplified](#)
- [Detailed](#)

```
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.NewClientProfile())
    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())
}
```

Note :

For the purpose of demonstration, some nonessential items such as modification of the default configuration have been included to show the features of the SDK. When you write code to use the SDK, we recommend you use the default configuration as much as possible and make changes as needed.

More Samples

For more samples, please see the [examples](#) directory. For the sample of initializing a request for a complicated API, please see [example 1](#). For the sample of initializing a request by using a JSON string, please see [example 2](#).

Relevant Configuration

If you have no special needs, we recommend you use the default configuration.

Before creating a client, if necessary, you can make some configuration changes by modifying the field values in

```
profile.ClientProfile .
```

```
// Nonessential steps
// Instantiate a client configuration object. You can specify the timeout period
// and other configuration items
cpf := profile.NewClientProfile()
```

The specific configuration items are as described below:

Request method

The SDK uses the POST method by default. If you have to use the GET method, you can set it here, **but the GET method cannot handle some large requests.**

```
cpf.HttpProfile.ReqMethod = "POST"
```

Timeout period

The SDK has a default timeout period. Do not adjust it unless absolutely necessary. If needed, check in the code to get the latest default value.

Unit: seconds

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

Specifying domain name

The SDK automatically specifies the domain name, so you generally don't need to do so. However, if you are accessing a service in a finance zone, you have to manually specify the domain name. For example, the CVM domain name of the Shanghai Finance Zone is `cvm.ap-shanghai-fsi.tencentcloudapi.com`.

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

Signature algorithm

The SDK uses `TC3-HMAC-SHA256` for signing by default, which is more secure but slightly reduces the performance.

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

DEBUG mode

More detailed logs will be printed in DEBUG mode, which can be enabled when you need to troubleshoot in detail. The default value is `false`.

```
cpf.Debug = true
```

Proxy

If there is a proxy in your environment, you need to set the system environment variable `https_proxy`; otherwise, it may not be called normally, and a connection timeout exception will be thrown.

Enabling DNS cache

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

Ignoring server certificate verification

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

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

Note :

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

Error Handling

Starting from `v1.0.181` , Tencent Cloud SDK for Go defines the error codes returned by each product as constants. You can directly call them for handling without manual definition required. If you use an IDE (such as GoLand) for development, you can use their code hint features to choose directly; for example,

```
...//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
}
...
```

At the same time, the comment section of each API function also lists the error codes it may return for your reference:

```
// DescribeInstances
// This API is used to query the details of one or multiple instances.
//
```

```
//
//
// * You can query the details of an instance according to its `ID`, name, billing mode, or other information. For more information on how to filter information, please see `Filter`.
//
// * If no parameter is defined, the status of a certain number of instances under the current account will be returned. The number is specified by `Limit` and is 20 by default.
//
// * The latest operation (LatestOperation) and the latest operation status (LatestOperationState) of the instance can be queried.
//
// Possible error codes:
// 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"
// RESOURCE_NOTFOUND_HPCCLUSTER = "ResourceNotFound.HpcCluster"
// UNAUTHORIZEDOPERATION_INVALIDTOKEN = "UnauthorizedOperation.InvalidToken"
func (c *Client) DescribeInstances(request *DescribeInstancesRequest) (response *DescribeInstancesResponse, err error){
    ...
}
```

Common Request

Starting from `v1.0.189`, Tencent Cloud SDK for Go supports the use of `Common Request` mode for requests. You only need to install the `common` package to initiate calls to any Tencent Cloud product.

Note :

You must clearly know the parameters required by the called API; otherwise, the call may fail.

Currently, only the POST method is supported, and the signature algorithm must be signature algorithm v3. For detailed usage, please see the [Using Common Request to Call](#) sample.

Request Retry

Retry upon network error

The SDK can be configured to automatically retry when a network error or timeout occurs. This is not enabled by default. You can configure the number of retries and retry interval through `ClientProfile`.

Note :

The `Request` structure is checked for whether it has the `ClientToken` field, and if so, it will be considered an idempotent request. Only idempotent requests will be retried automatically in case of network errors, while exceptions will be thrown for non-idempotent requests to prevent inconsistent results caused by multiple replays of the requests.

```
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 // Define the maximum number of retry attempts
    prof.NetworkFailureRetryDuration = profile.ExponentialBackoff // Define the retry
    interval time
```

```
client, _ := cvm.NewClient(credential, regions.Guangzhou, prof)
// ...
}
```

For more usage information, please see the [testing file](#).

Retry upon frequency limiting

The SDK can be configured to automatically retry when API call frequency limiting is triggered. This is not enabled by default. You can configure the number of retries and retry interval through `ClientProfile`.

```
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 // Define the maximum number of retry attempts
    prof.RateLimitExceededRetryDuration = profile.ExponentialBackoff // Define the retry interval time
    client, _ := cvm.NewClient(credential, regions.Guangzhou, prof)
    // ...
}
```

Idempotence flag

When retry upon network timeout or frequency limiting is enabled, the `ClientToken` parameter will be automatically injected into the request (if the request has an empty `ClientToken` field). If you manually specify `ClientToken`, the injection process will be skipped.

Note :

The injected `ClientToken` provides global uniqueness below a `100000/s` concurrency.

FAQs

Package import failed

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

.NET

Last updated : 2021-08-10 16:14:49

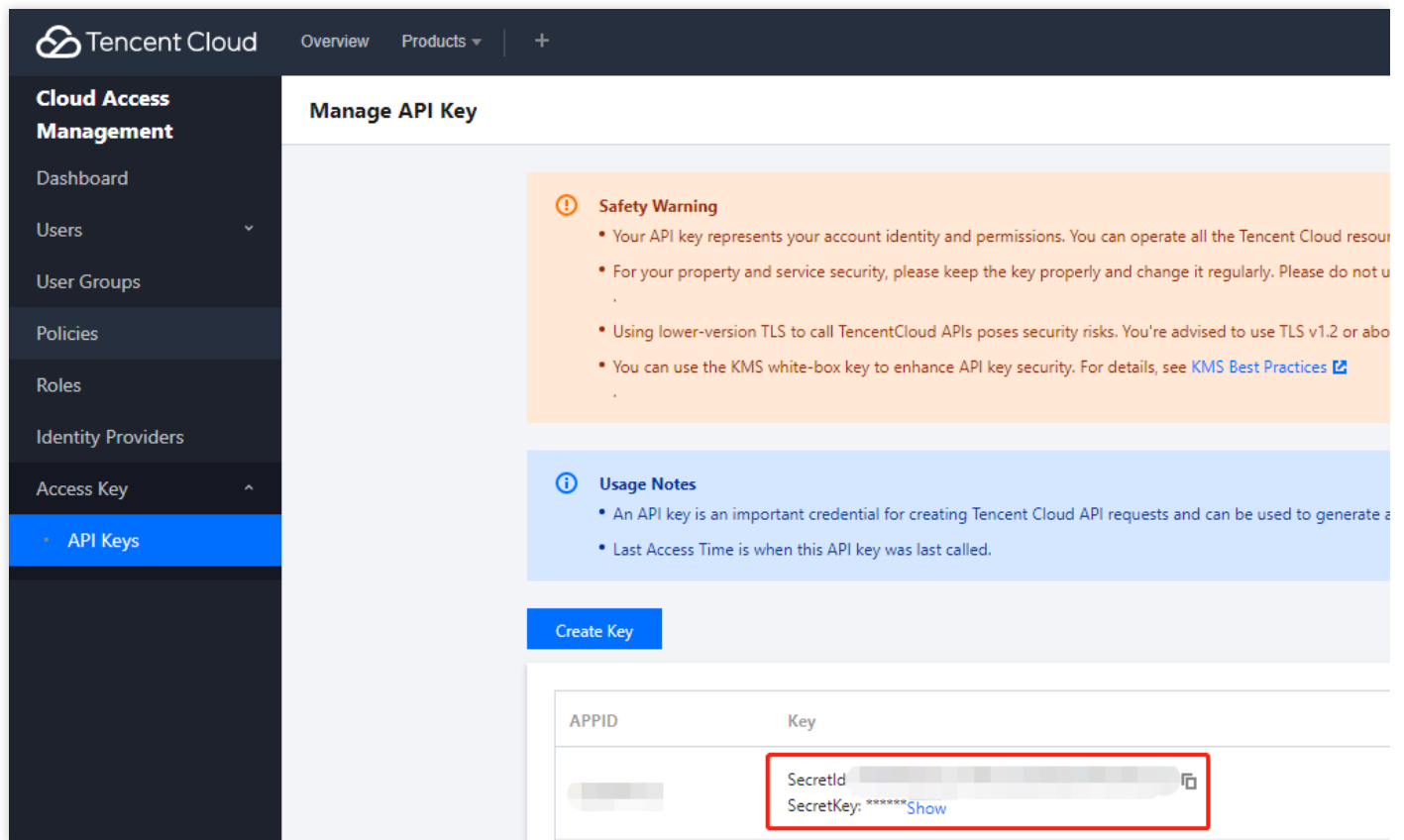
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform. SDK 3.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document describes how to use, debug, and connect to TencentCloud APIs with the SDK for .NET 3.0 as an example.
- This version currently supports various Tencent Cloud products such as CVM, VPC, and CBS and will support more products in the future.

Dependent Environment

- .NET Framework 4.5+ or .NET Core 2.1.
- Get the security credential, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose it to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by product. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding product .

Installing SDK

Method 1. Install through NuGet (recommended)

Install on the command line (the version here is only an example. Please choose the latest version):

```
dotnet add package TencentCloudSDK --version 3.0.0
# For other information, please go to www.nuget.org/packages/TencentCloudSDK
```

Note :

The commands should be executed in the home directory of the project.

Add a package through Visual Studio:

For example, to create a HelloWorld project, run:

```
dotnet new console -o HelloWorld
# Enter the home directory of the HelloWorld project
dotnet run
# The output is: Hello World!
dotnet add package TencentCloudSDK --version 3.0.0
# Download SDK dependencies for the project
```

Note :

- If you want to install the package of a certain product such as CVM, just add the dependency

```
TencentCloudSDK.Cvm .
```

Method 2. Install through source code

Go to the [GitHub code hosting page](#) to download the latest code, decompress and install it in your working directory, and then open it with Visual Studio 2017 for compiling.

Using SDK

Each API has a corresponding request structure and a response structure. For example, the

`DescribeInstances` API for querying the CVM instance list has a request structure

`DescribeInstancesRequest` and a response structure `DescribeInstancesResponse` .

The following uses the CVM instance list querying API as an example to describe the basic usage of the SDK.

- [Simplified](#)
- [Detailed](#)

```
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());
            }
        }
    }
}
```

More samples

For more samples, please see the `TencentCloudExamples` directory in the [GitHub repository](#).

Sync and async calls

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

Note :

In the sample, as it is a console application, you can call the async APIs synchronously, i.e.,

`ConfigureAwait(false).GetAwaiter().GetResult()`. When developing ASP applications or Windows Forms applications, you cannot call the async APIs synchronously in the response method of UI controls; otherwise, the UI will stop responding.

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

Relevant Configuration

Proxy

If you use the SDK to call an API in a proxy environment, you need to set the system environment variable

`https_proxy` (as shown in the sample code); otherwise, it may not be called normally, and a connection timeout exception will be thrown.

FAQs

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

C++

Last updated : 2021-08-10 16:14:50

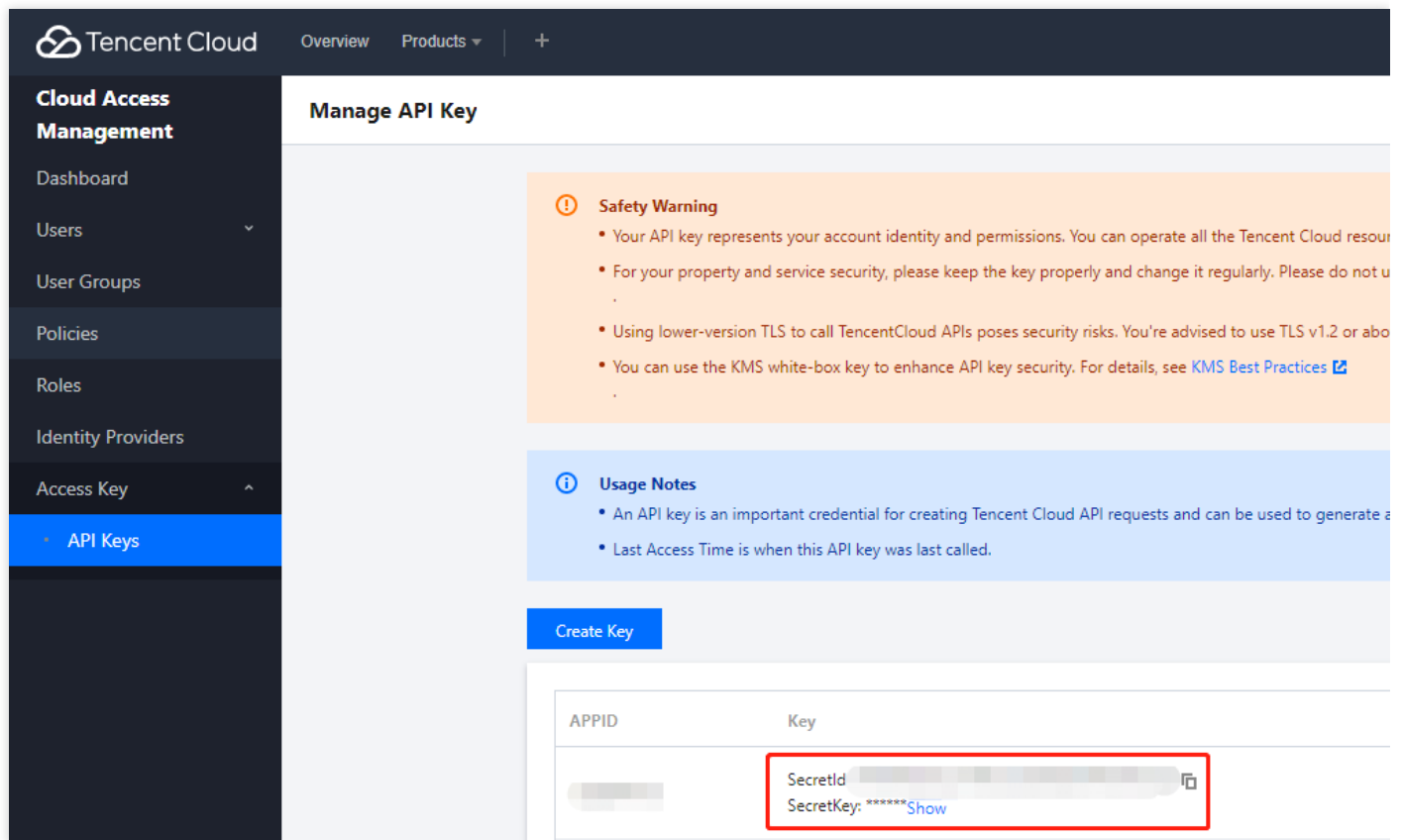
Overview

- Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform. SDK 3.0 is unified and features the same SDK usage, API call methods, error codes, and returned packet formats for different programming languages.
- This document describes how to use, debug, and connect to TencentCloud APIs with the SDK for C++ 3.0 as an example.
- This version currently supports various Tencent Cloud products such as CVM, VPC, and CBS and will support more products in the future.

Dependent Environment

- The compiler for C++ v11 or above is required, i.e., GCC v4.8 or above. Currently, only the Linux installation environment is supported, while Windows is not.
- Get the security credential, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the

server. You can get them on the [API Key Management](#) page as shown below:



Note :

Your security credential represents your account identity and granted permissions, which is equivalent to your login password. Do not disclose it to others.

- Get the calling address (endpoint), which is generally in the format of `*.tencentcloudapi.com` and varies by product. For example, the endpoint of CVM is `cvm.tencentcloudapi.com`. For specific endpoints, please see the [API documentation](#) of the corresponding product.
- Install [CMake](#) 3.0 or above. Below is the sample code for installation:

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

- Install the [libcurl](#) dependency library. We recommend you install the latest version of `libcurl`; otherwise, there may be a `libcurl` memory leak bug.

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

- Install the [openssl](#) dependency library. Below is the sample code for installation:

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

- Install the `libuuid` dependency library. Below is the sample code for installation:

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

Installing SDK

Building SDK from source code

1. Go to the [GitHub code hosting page](#) to download the latest source code package.
2. Enter the SDK and create necessary build files. Here, `path/to/` represents the actual path of the `tencentcloud-sdk-cpp` package.

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

Using SDK

Note :

The demo cannot be run directly. You need to replace the key and other information with your real and usable information. It is better to configure them in environment variables to avoid exposing them in the code.

The following describes how to call the `DescribeInstances` API of CVM.

- [Simplified](#)
- [Detailed](#)

```
#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;  
}
```

Compile and run the demo code:

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

If the system returns an error indicating that a dynamic library is not found, you can specify the path to the dynamic library. For example, if the `libtencentcloud-sdk-cpp-core.so` library is installed in `/usr/local/lib`, then add the following line:

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

Note :

For more samples, please see the [example](#) directory.

Unit Testing

Creating dependent library for gtest

The sample installation code is as follows:

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

Copy the generated `libgtest.a` and `libgtest_main.a` static libraries and the header file of gtest to the system directory.

Configuring environment variables

- ENV_SecretId: secret ID
- ENV_SecretKey: secret Key

Testing

Run the following script:

```
sh function_test.sh
```

Relevant Configuration

Proxy

If you use the SDK to call an API in a proxy environment, you need to set the system environment variable

`https_proxy` (as shown in the sample code); otherwise, it may not be called normally, and a connection timeout exception will be thrown.

Ruby

Last updated : 2021-08-10 16:14:50

Overview

Welcome to Tencent Cloud Software Development Kit (SDK) 3.0, a companion tool for the TencentCloud API 3.0 platform.

Dependent Environment

1. Dependent environment: Ruby 2.3 or above.
2. Activate your product in the [Tencent Cloud console](#).
3. Get the `SecretID` , `SecretKey` , and endpoint. The general format of endpoint is `*.tencentcloudapi.com` . For example, the endpoint of CVM is `cvm.tencentcloudapi.com` . For more information, please see the documentation of the specified product.

Getting and Installing

Before installing the SDK for Ruby and using TencentCloud API for the first time, you need to apply for security credentials in the [Tencent Cloud console](#), which consists of `SecretId` and `SecretKey` .

- `SecretId` is used to identify the API requester.
- `SecretKey` is used to encrypt the string to sign that can be verified on the server.

Note :

You must keep the `SecretKey` private and avoid disclosure.

Installing through source package

Go to the [GitHub code hosting page](#) to download the latest code, decompress it, and run the following command (with the CVM SDK as an example):

```
$ 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
```

Note :

For the above version numbers, the actual values shall prevail.

Example

The following takes the instance list querying API as an example.

Simplified version

```
# -*- 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
```

Detailed version

```
# -*- coding: UTF-8 -*-
require 'tencentcloud-sdk-common'
require 'tencentcloud-sdk-cvm'
begin
  include TencentCloud::Common
  # Import the client module of the corresponding product module
  include TencentCloud::Cvm::V20170312
  # Instantiate an authentication object. Pass in `secretId` and `secretKey` of your
  # Tencent Cloud account as the input parameters and keep them confidential
```

```
cred = Credential.new('SecretId', 'SecretKey')
# Instantiate an HTTP option
httpProfile = HttpProfile.new()
# If you need to specify the proxy for API access, you can initialize HttpProfile
as follows
# httpProfile = HttpProfile.new(proxy='http://username:password@proxy IP:proxy po
rt')
httpProfile.scheme = "https" # HTTP is supported if the network environment has a
ccess to the public network, and HTTPS is used by default and recommended
httpProfile.req_method = "GET" # GET request (POST request is used by default)
httpProfile.req_timeout = 30 # Specify the request timeout value in seconds. The
default value is 60s
httpProfile.endpoint = "cvm.tencentcloudapi.com" # Specify the endpoint. If you d
o not specify the endpoint, nearby access is enabled by default
# (Optional) Instantiate a client option
clientProfile = ClientProfile.new()
clientProfile.sign_method = "TC3-HMAC-SHA256" # Specify the signature algorithm
clientProfile.language = "en-US" # Specify to display in English (the default val
ue is Chinese)
clientProfile.http_profile = httpProfile
clientProfile.debug = true # Print `debug` logs
# Instantiate the client object of the requested product (with CVM as an exampl
e). `clientProfile` is optional.
client = Client.new(cred, "ap-shanghai", clientProfile)
# Instantiate a CVM instance information query request object. Each API correspon
ds to a request object
req = DescribeInstancesRequest.new()
# Populate the request parameters. Here, the member variables of the request obje
ct are the input parameters of the corresponding API.
# You can view the definition of the request parameters in the API documentation
at the official website or by redirecting to the definition of the request objec
t.
respFilter = Filter.new() # Create a `Filter` object to query CVM instances in th
e `zone` dimension.
respFilter.Name = "zone"
respFilter.Values = ["ap-shanghai-1", "ap-shanghai-2"]
req.Filters = [respFilter] # `Filters` is a list of `Filter` objects
# Initialize the request by calling the `DescribeInstances` method on the client
object. Note: the request method name corresponds to the request object
# The returned `resp` is an instance of the `DescribeInstancesResponse` class whi
ch corresponds to the request object
resp = client.DescribeInstances(req)
# A string return packet in JSON format is outputted
puts resp.serialize
# You can also take a single value.
# You can view the definition of the return field in the API documentation at the
official website or by redirecting to the definition of the response object.
```

```
puts resp.TotalCount
rescue TencentCloudSDKException => e
puts e.message
puts e.backtrace.inspect
end
```

Relevant Configuration

Proxy

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

1. Specify the `proxy` when initializing `HttpProfile` .
2. You need to set the system environment variable `https_proxy` .

Note :

Otherwise, it may not be called normally, and a connection timeout exception will be thrown.