SDK 中心

Python

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

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

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

Python

最近更新时间：2025-06-20 11:29:02

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。
为方便 Python 开发者调试和接入腾讯云产品 API，本文向您介绍适用于 Python 的腾讯云开发工具包，提供了简单示例来帮助您快速获取腾讯云 Python SDK 并开始调用。

依赖环境

1. 支持的版本：Python 2.7，3.6-3.12。
2. 从 腾讯云控制台 开通相应产品。

依赖问题

本项目依赖 requests 库，由于 requests 库2.30.0及以上的版本适配了 urllib3 2.0版本，如果您在使用 SDK 时遇到以下错误：

```
ImportError: urllib3 v2.0 only supports OpenSSL 1.1.1+, currently the
'ssl' module is compiled with 'OpenSSL 1.0.x'
```

可以使用如下方法之一解决：

- 将 urllib3 的版本降低到1.26.x版本（requsets 库的依赖版本是 urllib3 >= 1.21.1, < 3）。
- 将 python 环境使用 openssl 1.1.1+版本重新编译。

获取安装

安装 Python 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。SecretKey 必须严格保管，避免泄露。

通过 Pip 安装（推荐）

您可以使用 pip 安装方式将腾讯云 API Python SDK 安装到您的项目中。如果您的项目环境尚未安装 pip，请参见 pip 官网安装。
通过 pip 方式安装或更新请在命令行中执行以下命令：

安装指定产品 SDK（推荐）

例如，安装指定产品包：

```
pip install --upgrade tencentcloud-sdk-python-common  # 安装公共包，必选
pip install --upgrade tencentcloud-sdk-python-指定产品包名缩写  # 如 CVM 产品包：tencentcloud-sdk-python-cvm
```

具体产品的包名缩写请参考 products.md 中的包名字段。

安装全产品 SDK

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

全产品 SDK 包含了所有云产品的调用代码，体积偏大，对体积敏感的场景，推荐安装指定产品 SDK。

注意事项

- 安装全产品 SDK 和安装指定产品的 SDK 两种方式只能选择其中一种。
- 如果同时安装多个产品的包，建议多个产品的包和 common 包保持在同一个版本。
- 无法使用官方源的用户可以使用镜像源提高下载速度，例如：
  `pip install -i https://mirrors.tencent.com/pypi/simple/ --upgrade tencentcloud-sdk-python-cvm`
  。
- 如果同时有 python2 和 python3 环境， python3 环境需要使用 pip3 命令安装。

通过源码包安装

前往 Github 仓库 或者 Gitee 仓库 下载最新代码，解压后执行以下命令：

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

示例

以云服务器 CVM 产品的 DescribeInstances 接口为例：

简化版

```
import os
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import
TencentCloudSDKException
from tencentcloud.cvm.v20170312 import cvm_client, models
```

```
try:
    # 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中，请参考本文凭证管理章
节。
    # 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
    # cred = credential.Credential("secretId", "secretKey")
    cred = credential.Credential(
        os.environ.get("TENCENTCLOUD_SECRET_ID"),
        os.environ.get("TENCENTCLOUD_SECRET_KEY"))
    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)
```

详细版

```
# -*- coding: utf-8 -*-
import os
import sys
import logging

from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import
TencentCloudSDKException
# 导入对应产品模块的client models。
from tencentcloud.cvm.v20170312 import cvm_client, models

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

```
    # 实例化一个http选项，可选的，没有特殊需求可以跳过。
    httpProfile = HttpProfile()
    # 如果需要指定proxy访问接口，可以按照如下方式初始化hp
    # httpProfile = HttpProfile(proxy="http://用户名:密码@代理IP:代理端口")
    httpProfile.protocol = "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.keepAlive = True  # 状态保持，默认是False
    httpProfile.reqMethod = "GET"  # get请求(默认为post请求)
    httpProfile.reqTimeout = 30    # 请求超时时间，单位为秒(默认60秒)
    httpProfile.endpoint = "cvm.ap-shanghai.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()
    clientProfile.signMethod = "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.httpProfile = httpProfile

    # 实例化要请求产品(以cvm为例)的client对象，clientProfile是可选的。
    client = cvm_client.CvmClient(cred, "ap-shanghai", clientProfile)

    # 打印日志按照如下方式，也可以设置log_format，默认为 '%(asctime)s %
(process)d %(filename)s L%(lineno)s %(levelname)s %(message)s'
    # client.set_stream_logger(stream=sys.stdout, level=logging.DEBUG)
    # client.set_file_logger(file_path="/log", level=logging.DEBUG) 日志
文件滚动输出，最多10个文件，单个文件最大512MB
    # client.set_default_logger() 去除所有log handler，默认不输出

    # 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 = models.DescribeInstancesRequest()

    # 填充请求参数,这里request对象的成员变量即对应接口的入参。
    # 您可以通过官网接口文档或跳转到request对象的定义处查看请求参数的定义。
    respFilter = models.Filter()  # 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

    # python sdk支持自定义header如 X-TC-TraceId、X-TC-Canary，可以按照如下方式
指定，header必须是字典类型的
    headers = {
        "X-TC-TraceId": "ffe0c072-8a5d-4e17-8887-a8a60252abca"
```

```
    }
    req.headers = headers

    # 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，headers为可选参数。
    # 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
    print(resp.to_json_string(indent=2))

    # 也可以取出单个值。
    # 您可以通过官网接口文档或跳转到response对象的定义处查看返回字段的定义。
    print(resp.TotalCount)
except TencentCloudSDKException as err:
    print(err)
```

Common Client 调用方式

从 `3.0.396` 版本开始，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。

> **注意：**
> 您必须明确了解您调用的接口所需的参数，否则可能会导致调用失败。

Common Client 的示例请参见 example。

更多示例

您可以在 GitHub 的 examples 目录下找到更详细的示例代码。

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，参考 example 。
2. 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。

证书问题

在 Mac 操作系统安装 Python 3.6 或更高版本时，可能会遇到证书错误。错误信息如下：

```
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，并且 Python 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，因此只能通过执行以下命令来安装证书以解决此问题：

```
sudo "/Applications/Python 3.6/Install Certificates.command"
```

虽然 Python 2 版本不应该有同样的问题，但是在个别用户环境上确实也观察到有类似的情况，也一样可以通过以下命令解决：

```
sudo /Applications/Python 2.7/Install Certificates.command
```

腾讯云 Python SDK 默认使用 certifi 库提供的证书。如果您需要指定其他证书，可以进行如下设置：

```
# 指定证书
httpProfile.certification = "/path/to/certification"
```

如果您想跳过证书校验，可以将参数设置为 False，进行如下设置：

```
# 跳过证书校验
httpProfile.certification = False
```

凭证管理

腾讯云 Python SDK 目前支持以下几种方式进行凭证管理：

环境变量

默认情况下，SDK 会读取环境变量 `TENCENTCLOUD_SECRET_ID` 和 `TENCENTCLOUD_SECRET_KEY` 获取 SecretId 和 SecretKey。相关代码：

```
from tencentcloud.common import credential
cred = credential.EnvironmentVariableCredential().get_credential()
```

配置文件

SDK 支持读取配置文件中的凭证信息。配置文件的路径要求如下：

- Windows：

```
c:\Users\NAME\.tencentcloud\credentials
```

- Linux：

```
~/.tencentcloud/credentials 或 /etc/tencentcloud/credentials
```

配置文件的格式为 .ini 格式，示例如下：

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

您可以使用以下代码获取凭证：

```
from tencentcloud.common import credential
cred = credential.ProfileCredential().get_credential()
```

角色扮演

有关角色扮演的相关概念请参见 腾讯云角色概述。如果使用此种方式，您需要在腾讯云访问管理控制台上创建一个角色，具体创建过程请参见 腾讯云角色创建。在您拥有角色后，您可以使用以下代码获取临时凭证：

```
from tencentcloud.common import credential
cred = credential.STSAssumeRoleCredential("SecretId", "SecretKey",
"RoleArn", "RoleSessionName")
```

有关角色扮演的详细使用方式可以参考示例： 使用角色

实例角色

有关实例角色的相关概念请参见 腾讯云实例角色。
在您为实例绑定角色后，您可以在实例中访问相关元数据接口获取临时凭证，SDK 会自动刷新临时凭证。相关代码如下：

```
from tencentcloud.common import credential
cred = credential.CVMRoleCredential().get_credential()
```

凭证提供链

腾讯云 Python SDK 提供了凭证提供链，它会按照**环境变量 > 配置文件 > 实例角色 > TKE OIDC 凭证**的顺序尝试获取凭证，并返回第一个获取到的凭证。相关代码如下：

```
# 使用默认凭证提供链
from tencentcloud.common import credential
cred = credential.DefaultCredentialProvider().get_credential()

# 使用环境变量
from tencentcloud.common import credential
cred = credential.EnvironmentVariableCredential().get_credential()

# 使用配置文件
from tencentcloud.common import credential
cred = credential.ProfileCredential().get_credential()

# 使用实例角色
from tencentcloud.common import credential
cred = credential.CVMRoleCredential().get_credential()

# 使用TKE OIDC凭证
from tencentcloud.common import credential
cred = credential.DefaultTkeOIDCRoleArnProvider().get_credential()
```

具体示例代码可以参考 credential_providers.py 中的示例。

地域容灾

从 3.0.923 版本开始，腾讯云 Python SDK 支持地域容灾功能。当某个域名请求失败时，SDK 会自动切换到容灾域名。使用方式如下：

使用地域容灾功能时，有三种状态相互转换：**关闭**、**全开**和**半开**状态。

- **关闭状态**：使用主要域名进行请求，如果出现错误，则会切换到全开状态。
- **全开状态**：使用容灾域名进行请求。当达到一定时间时，会切换到半开状态。
- **半开状态**：此时会将少量请求发送到主要域名，如果请求失败，则切换到全开状态。当请求成功数达到一定数量时，会切换到关闭状态。

以下是相关设置示例：

```
# 简单开启方式，此时所有的配置都是默认值
from tencentcloud.common.profile.client_profile import ClientProfile
clientProfile = ClientProfile()
clientProfile.disable_region_breaker = False  # False表示使用地域容灾
```

```
# 自定义配置
from tencentcloud.common.profile.client_profile import ClientProfile,
RegionBreakerProfile
regionBreakerProfile = RegionBreakerProfile(
    backup_endpoint="ap-beijing.tencentcloudapi.com",  # 备用地域，格式
${region}.tencentcloudapi.com，必须是存在的域名，默认值为ap-
guangzhou.tencentcloudapi.com
    max_fail_num=3,  # 最大失败数，默认值5
    max_fail_percent=0.5,  # 最大失败率，默认值0.75。当失败数达到最大失败数，且
失败率达到最大的失败率时，或者连续失败数达到5次，关闭状态切换到开启状态
    window_interval=60,  # 计数窗口，单位s，默认300。处于关闭状态时，时间超过窗口
则重新计数
    timeout=30,  # 全开时间，单位s，默认60。处于全开状态达到超过该时间，切换为半开
状态
    max_requests=3  # 最大成功请求数，默认5。处于半开状态时，请求主域名达到该数量
则切换为关闭状态
)
clientProfile = ClientProfile()
clientProfile.disable_region_breaker = False  # 使用地域容灾必须要将这个值置
为false
clientProfile.region_breaker_profile = regionBreakerProfile
```