

# SDK 中心

# Python

# 产品文档



腾讯云

**【 版权声明 】**

©2013–2023 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 】**

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。

# Python

最近更新时间：2023-01-11 20:20:38

## 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：

APPID	密钥	创建时间	最近访问时间	最近访问服务	状态	操作
	SecretId: [redacted] SecretKey: ***** <a href="#">显示</a>	2020-11-20 10:35:35	-	-	已启用	禁用

### ⚠ 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`
- 更多 SDK 支持的云产品，请参见 [云产品名列表](#)。

### 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

```
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

# 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
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 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。