

# SDK 中心

# Ruby



腾讯云

## 【 版权声明 】

©2013–2024 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內容。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或 95716。

# Ruby

最近更新时间：2023-10-08 11:14:12

## 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。SecretKey 必须严格保管，避免泄露。

## 通过源码包安装

前往 [Github 仓库](#) 或者 [Gitee 仓库](#) 下载最新代码，以安装cvm sdk为例，解压后：

```
$ cd tencentcloud-sdk-ruby
$ 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
  # 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
  # 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
  # cred = Credential.new('SecretId', 'SecretKey')
  cred = Credential.new(ENV["TENCENTCLOUD_SECRET_ID"],
ENV["TENCENTCLOUD_SECRET_KEY"])
  cli = Client.new(cred, 'ap-guangzhou')
  req = DescribeInstancesRequest.new(nil, nil, 0, 1)
  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。
  # 为了保护密钥安全，建议将密钥设置在环境变量中或者配置文件中。
  # 硬编码密钥到代码中有可能随代码泄露而暴露，有安全隐患，并不推荐。
  # cred = Credential.new('SecretId', 'SecretKey')
  cred = Credential.new(ENV["TENCENTCLOUD_SECRET_ID"],
ENV["TENCENTCLOUD_SECRET_KEY"])

  # 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。
# 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 = 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。
# 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。
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`。

否则可能无法正常调用，抛出连接超时的异常。