SDK 中心

.NET

腾讯云

【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。

.NET

最近更新时间：2023-10-08 11:15:22

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。后续所有的云服务产品都会接入进来。新版 SDK 实现了统一化，具有各个语言版本的 SDK 使用方法相同，接口调用方式相同，统一的错误码和返回包格式这些优点。
为方便 .NET 开发者调试和接入腾讯云产品 API ，本文向您介绍适用于 .NET 的腾讯云开发工具包，并提供首次使用开发工具包的简单示例。让您快速获取腾讯云 .NET SDK 并开始调用。

依赖环境

1. 依赖环境：兼容 .NET standard 2.0（.NET Framework 4.5+ 或者 .NET Core 2.1）。
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。
4. 下载相关资料并做好相关文件配置。

获取安装

安装 .NET 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nuget 安装(推荐)

1. 通过命令行安装：`dotnet add package TencentCloudSDK` ，其他信息请到 nuget 获取。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。
2. 通过 Visual Studio 的添加包。

通过源码安装

前往 Github 仓库 或者 Gitee 仓库 下载最新代码，解压后使用 Visual Studio 2017 打开编译。

示例

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。

简略版

```
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
            {
                // 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。
                // 这里采用的是从环境变量读取的方式，需要在环境变量中先设置这两个值。
                Credential cred = new Credential {
                    SecretId =
Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID"),
                    SecretKey =
Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY")
                };
                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());
            }
        }
    }
}
```

详细版

```
using System;
using System.Threading.Tasks;
using TencentCloud.Common;
using TencentCloud.Common.Profile;
```

```
using TencentCloud.Cvm.V20170312;
using TencentCloud.Cvm.V20170312.Models;

namespace TencentCloudExamples
{
    class DescribeInstances
    {
        static void Main(string[] args)
        {
            try
            {
                // 必要步骤：
                // 实例化一个认证对象，入参需要传入腾讯云账户密钥对 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。
                // 这里采用的是从环境变量读取的方式，需要在环境变量中先设置这两个值。
                Credential cred = new Credential {
                    SecretId =
Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID"),
                    SecretKey =
Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY")
                };

                // 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();
                // 指定签名算法为签名方法v3（TC3-HMAC-SHA256），可以处理超过1MB的请求。
（从3.0.711版本起，默认已设置为签名方法v3）
                clientProfile.SignMethod = ClientProfile.SIGN_TC3SHA256;
                // 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
                HttpProfile httpProfile = new HttpProfile();
                // SDK默认使用POST方法。
                // 如果您一定要使用GET方法，可以在这里设置。GET方法无法处理一些较大的请求。
                httpProfile.ReqMethod = "POST";
                // SDK有默认的超时时间，非必要请不要进行调整。
                // 如有需要请在代码中查阅以获取最新的默认值。
                httpProfile.Timeout = 10; // 请求连接超时时间，单位为秒(默认60秒)
                // SDK会自动指定域名。通常是不需要特地指定域名的，但是如果您访问的是金融区的
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
                httpProfile.Endpoint = ("cvm.tencentcloudapi.com");
                // 代理服务器，当您的环境下有代理服务器时设定
                httpProfile.WebProxy =
Environment.GetEnvironmentVariable("HTTPS_PROXY");
```

```
            clientProfile.HttpProfile = httpProfile;

            // 实例化要请求产品(以cvm为例)的client对象
            // 第二个参数是地域信息，可以直接填写字符串ap-guangzhou，或者引用预设的常量，clientProfile是可选的
            CvmClient client = new CvmClient(cred, "ap-guangzhou", clientProfile);

            // 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
            // 您可以直接查询SDK源码确定DescribeInstancesRequest有哪些属性可以设置，
            // 属性可能是基本类型，也可能引用了另一个数据结构。
            // 推荐使用IDE进行开发，可以方便的跳转查阅各个接口和数据结构的文档说明。
            DescribeInstancesRequest req = new DescribeInstancesRequest();

            // 基本类型的设置。
            // 此接口允许设置返回的实例数量。此处指定为只返回一个。
            // SDK采用的是指针风格指定参数，即使对于基本类型您也需要用指针来对参数赋值。
            // SDK提供对基本类型的指针引用封装函数
            req.Limit = 1;
            // 数组类型的设置。
            // 此接口允许指定实例 ID 进行过滤，但是由于和接下来要演示的 Filter 参数冲突，先注释掉。
            // req.InstanceIds = new string[] { "ins-r8hr2upy" };

            // 复杂对象的设置。
            // 在这个接口中，Filters是数组，数组的元素是复杂对象Filter，Filter的成员Values是string数组。
            // 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
            Filter respFilter = new Filter(); // 创建Filter对象, 以zone的维度来查询cvm实例
            respFilter.Name = "zone";
            respFilter.Values = new string[] { "ap-guangzhou-1", "ap-guangzhou-2" };
            req.Filters = new Filter[] { respFilter }; // Filters 是成员为Filter对象的列表

            //// 这里还支持以标准json格式的string来赋值请求参数的方式。下面的代码跟上面的参数赋值是等效的
            //string strParams = "{\"Filters\":[{\"Name\":\"zone\",\"Values\":[\"ap-guangzhou-1\",\"ap-guangzhou-2\"]}]}";
            //req = DescribeInstancesRequest.FromJsonString<DescribeInstancesRequest>(strParams);

            // 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
            DescribeInstancesResponse resp = client.DescribeInstancesSync(req);
```

```
            // 使用同步接口调用结果
            // DescribeInstancesResponse resp = client.DescribeInstancesSync(req);

            // 输出json格式的字符串回包
            Console.WriteLine(AbstractModel.ToJsonString(resp));

            // 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
            Console.WriteLine(resp.TotalCount);

        }
        catch (Exception e)
        {
            Console.WriteLine(e.ToString());
        }
        Console.Read();
    }
  }
}
```

CommonClient

从 v3.0.713 开始，腾讯云 DOTNET 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 common 包， 即可向任何产品发起调用。

> ⚠ **注意：**
> 您必须明确知道您调用接口的参数内容，否则会调用失败。

详细使用请参阅示例： 使用 Common Client 进行调用 。
更多示例参见 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 TencentCloudExamples 示例项目中由于是控制台应用程序，所以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。
解决的办法是将 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。

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。

旧版SDK

我们推荐您使用新版 SDK ， 如果需要旧版 SDK ，请访问 qcloudapi sdk for dotnet。