

云函数
SDK文档
产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【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。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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。

SDK文档

最近更新时间：2024-12-02 20:36:46

开发准备

安装 SDK 前，需要先获取安全凭证。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端验证签名字符串的密钥。SecretKey 必须严格保管，避免泄露。

安装 SDK

[Python](#)

[Node.js](#)

[PHP](#)

[Java](#)

[Go](#)

[.NET](#)

[C++](#)

API

SCF 常用的 API 如下，更多 API 可参考 [API 文档](#)。

接口名称	接口功能
CreateFunction	创建函数
DeleteFunction	删除函数
GetFunction	获取函数详细信息
Invoke	运行函数
ListFunctions	获取函数列表
UpdateFunctionCode	更新函数代码
UpdateFunctionConfiguration	更新函数配置

使用示例

Python

Node.js

PHP

以 Python3.6 为例：

```
# -*- coding: utf8 -*-

import json
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudS
# 导入对应产品模块的client models
from tencentcloud.scf.v20180416 import scf_client,models

# 对应接口的接口名
action = 'Invoke'

# 接口参数,输入需要调用的函数名, RequestResponse(同步) 和 Event(异步)
action_params = {
    'FunctionName': "function-name",
    'InvocationType': "Event"
}

print('Start SCF')

def main_handler(event, context):
    try:
        # 实例化一个认证对象, 入参需要传入腾讯云账户secretId, secretKey
        cred = credential.Credential("SecretId", "SecretKey")

        # 实例化要请求产品的client对象, 以及函数所在的地域
        client = scf_client.ScfClient(cred, "ap-shanghai")

        # 调用接口, 发起请求, 并打印返回结果
        ret = client.call(action, action_params)

        print(json.loads(ret) ["Response"] ["Result"] ["RetMsg"])

    except TencentCloudSDKException as err:
        print(err)
```

以 `Node.js12.16` 为例：

说明：

如果在层中使用 SDK，请在代码中指定绝对路径，即 `/opt/node_modules/tencentcloud-sdk-nodejs`。

```
'use strict';
const tencentcloud = require("/var/user/node_modules/tencentcloud-sdk-nodejs");
// 导入对应产品模块的client models。
const ScfClient = tencentcloud.scf.v20180416.Client;
const models = tencentcloud.scf.v20180416.Models;

const clientConfig = {
  // 腾讯云认证信息
  credential: {
    secretId: "secretId",
    secretKey: "secretKey",
  },
  // 产品地域
  region: "ap-beijing",
  profile: {}
}

exports.main_handler = (event, context) => {
  console.log(event)
  // console.log(context)

  // 实例化要请求产品的client对象，以及函数所在的地域
  const client = new ScfClient(clientConfig);

  console.log("Start SCF")
  // 通过client对象调用想要访问的接口，需要传入请求对象以及响应回调函数
  client.Invoke({"FunctionName": "function-name", "InvocationType": "Event"}, functi
    // 请求异常返回，打印异常信息
    if (err) {
      console.log(err);
      return;
    }
    // 请求正常返回，打印response对象
    console.log("success");
  });
};
```

SCF 内置 SDK 使用示例

不同版本 `Node.js` 运行环境内置的 `tencentcloud-sdk-nodejs` 版本有差异，具体版本信息请参考[Node.js 环境内置库](#)。

以 Node.js12.16 为例：

```
'use strict';

const tencentcloud = require("tencentcloud-sdk-nodejs");
const Credential = tencentcloud.common.Credential;

// 导入对应产品模块的client models。
const ScfClient = tencentcloud.scf.v20180416.Client;
const models = tencentcloud.scf.v20180416.Models;

exports.main_handler = (event, context) = {
  console.log(event)
  // console.log(context)

  // 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey
  let cred = new Credential("SecretId", "SecretKey");

  // 实例化要请求产品的client对象，以及函数所在的地域
  let client = new ScfClient(cred, "ap-beijing");

  // 实例化一个请求对象,调用invoke()
  console.log("Start SCF")
  let request = new models.InvokeRequest();
  // 接口参数,输入需要调用的函数名, RequestResponse(同步) 和 Event(异步)
  let params = '{"FunctionName":"function-name", "InvocationType":"Event"}'
  request.from_json_string(params);
  // 通过client对象调用想要访问的接口，需要传入请求对象以及响应回调函数
  client.Invoke(request, function(err, response) {
    // 请求异常返回，打印异常信息
    if (err) {
      console.log(err);
      return;
    }
    // 请求正常返回，打印response对象
    console.log(response.to_json_string());
  });
};
```

示例如下：

```
<?php
require_once '/var/user/tencentcloud-sdk-php/TCloudAutoLoader.php'; #注意引用路径
use TencentCloud\Common\Credential;
use TencentCloud\Common\Profile\ClientProfile;
use TencentCloud\Common\Profile\HttpProfile;
```

```
use TencentCloud\\Common\\Exception\\TencentCloudSDKException;
use TencentCloud\\Scf\\V20180416\\ScfClient;
use TencentCloud\\Scf\\V20180416\\Models\\InvokeRequest;
function main_handler($event, $context) {
    print "good";
    print "\\n";
    var_dump($event);
    var_dump($context);
    try {
        // 实例化一个认证对象，入参需要传入腾讯云账户secretId, secretKey
        $cred = new Credential("SecretId", "SecretKey");
        $httpProfile = new HttpProfile();
        $httpProfile->setEndpoint("scf.tencentcloudapi.com");

        $clientProfile = new ClientProfile();
        $clientProfile->setHttpProfile($httpProfile);
        // 实例化要请求产品的client对象，以及函数所在的地域
        $client = new ScfClient($cred, "ap-shanghai", $clientProfile);
        $req = new InvokeRequest();
        // 接口参数,输入需要调用的函数名, RequestResponse(同步) 和 Event(异步)
        $params = '{"FunctionName":"function-name", "InvocationType":"Event"}';
        $req->fromJsonString($params);
        $resp = $client->Invoke($req);
        print_r($resp->toJsonString());
    }
    catch(TencentCloudSDKException $e) {
        echo $e;
    }
    return "hello";
}
?>
```

打包部署

如果需要在云函数控制台中部署函数，并使用 SDK 调用其他函数，则需要把 tencentcloud 的库和函数代码一起打包成 zip 文件。

注意在控制台创建函数时的执行方法，需要和 zip 文件里的代码文件和执行函数对应。

最终生成的 zip 包如果大于50MB，需要通过 COS 上传。

云 API 默认限频为每秒20次，如需提升并发上限，可以 [提交工单](#) 申请。

API Explorer

[API Explorer](#) 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度。

相关信息

您也可以使用腾讯云云函数 SDK（Tencentserverless SDK），该 SDK 集成云函数业务流接口，简化云函数的调用方法，使您无需再进行公有云 API 接口的封装。详情请参见 [函数间调用 SDK](#)。