

# Serverless Cloud Function

## SDK Documentation

### Product Documentation



## Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# SDK Documentation

Last updated : 2024-12-02 16:29:12

## Preparations for Development

Before installing the SDK and using TencentCloud API for the first time, you need to apply for security credentials in the Tencent Cloud console, which consists of `SecretId` and `SecretKey`. `SecretId` is used to identify the API requester, while `SecretKey` is a key used for signature string encryption and authentication by the server.

Please keep your `SecretKey` private and avoid disclosure.

## Installing SDK

[Python](#)

[Node.js](#)

[PHP](#)

[Java](#)

[Go](#)

[.NET](#)

[C++](#)

## API

The commonly used APIs of SCF are as follows. For more APIs, please see the [API documentation](#).

API	Description
<a href="#">CreateFunction</a>	Creates function
<a href="#">DeleteFunction</a>	Deletes function
<a href="#">GetFunction</a>	Gets function details
<a href="#">Invoke</a>	Executes function
<a href="#">ListFunctions</a>	Gets function list
<a href="#">UpdateFunctionCode</a>	Updates function code
<a href="#">UpdateFunctionConfiguration</a>	Updates function configuration

## Use Cases

Python

Node.js

PHP

Take `Python3.6` as an example:

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

import json
from tencentcloud.common import credential
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
# Import the client models of the corresponding product module
from tencentcloud.scf.v20180416 import scf_client,models

# API name of the corresponding API
action = 'Invoke'

# API parameter. Enter the name of the function to be invoked, `RequestResponse` (s
action_params = {
    'FunctionName': "function-name",
    'InvocationType': "Event"
}

print('Start SCF')

def main_handler(event, context):
    try:
        # Instantiate an authentication object. The Tencent Cloud account `secretId`
        cred = credential.Credential("SecretId", "SecretKey")

        # Instantiate the client object to request the product and the region where
        client = scf_client.ScfClient(cred, "ap-shanghai")

        # Call the API, initiate the request, and print the returned result
        ret = client.call(action, action_params)

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

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

Take `Node.js12.16` as an example:

**Note:**

If you use the SDK in a layer, please specify the absolute path in the code, i.e.,

```
/opt/node_modules/tencentcloud-sdk-nodejs .
```

```
'use strict';
const tencentcloud = require("/var/user/node_modules/tencentcloud-sdk-nodejs");
// Import the client models of the corresponding product module
const ScfClient = tencentcloud.scf.v20180416.Client;
const models = tencentcloud.scf.v20180416.Models;

const clientConfig = {
  // Tencent Cloud authentication information
  credential: {
    secretId: "secretId",
    secretKey: "secretKey",
  },
  // Product region
  region: "ap-beijing",
  profile: {}
}

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

  // Instantiate the client object to request the product and the region where th
  const client = new ScfClient(clientConfig);

  console.log("Start SCF")
  // Call the API you want to access through the client object; you need to pass
  client.Invoke({"FunctionName": "function-name", "InvocationType": "Event"}, functi
  // The request returns an exception and the exception information is printe
  if (err) {
    console.log(err);
    return;
  }
  // The request is returned normally, and the `response` object is printed
  console.log("success");
});
};
```

**Usage sample of SCF's built-in SDK**

The version of the built-in `tencentcloud-sdk-nodejs` varies by `Node.js` runtime environment version. For more information, please see [Notes on Node.js](#).

Take `Node.js12.16` as an example:

```
'use strict';

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

// Import the client models of the corresponding product module
const ScfClient = tencentcloud.scf.v20180416.Client;
const models = tencentcloud.scf.v20180416.Models;

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

  // Instantiate an authentication object. The Tencent Cloud account `secretId` and
  let cred = new Credential("SecretId", "SecretKey");

  // Instantiate the client object to request the product and the region where the f
  let client = new ScfClient(cred, "ap-beijing");

  // Instantiate a request object and call `invoke()`
  console.log("Start SCF")
  let request = new models.InvokeRequest();
  // API parameter. Enter the name of the function to be invoked, `RequestResponse`
  let params = '{"FunctionName":"function-name", "InvocationType":"Event"}'
  request.from_json_string(params);
  // Call the API you want to access through the client object; you need to pass in
  client.Invoke(request, function(err, response) {
    // The request returns an exception and the exception information is printed
    if (err) {
      console.log(err);
      return;
    }
    // The request is returned normally, and the `response` object is printed
    console.log(response.to_json_string());
  });
};
```

Example:

```
<?php
require_once '/var/user/tencentcloud-sdk-php/TCloudAutoLoader.php'; # Pay attention
```

```
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 {
        // Instantiate an authentication object. The Tencent Cloud account `secretId`
        $cred = new Credential("SecretId", "SecretKey");
        $httpProfile = new HttpProfile();
        $httpProfile->setEndpoint("scf.tencentcloudapi.com");

        $clientProfile = new ClientProfile();
        $clientProfile->setHttpProfile($httpProfile);
        // Instantiate the client object to request the product and the region
        $client = new ScfClient($cred, "ap-shanghai", $clientProfile);
        $req = new InvokeRequest();
        // API parameter. Enter the name of the function to be invoked, `Request`
        $params = '{"FunctionName":"function-name", "InvocationType":"Event"}';
        $req->fromJsonString($params);
        $resp = $client->Invoke($req);
        print_r($resp->toJsonString());
    }
    catch(TencentCloudSDKException $e) {
        echo $e;
    }
    return "hello";
}
?>
```

## Packaging and Deployment

If you need to deploy a function in the SCF console and use the SDK to invoke other functions, you need to package the `tencentcloud` library and function code together into a zip file.

Please note that the execution method specified when the function is created in the console must correspond to the code file and execution function in the zip file.

If the generated zip package is larger than 50 MB, it should be uploaded through COS.

The default call rate limit for TencentCloud API is 20 calls per second. If you need to increase the limit for high concurrence, please [submit a ticket](#) for application.

## API Explorer

[API Explorer](#) provides various capabilities such as online call, signature verification, SDK code generation, and quick API search, greatly improving the ease of use of TencentCloud API.

## Relevant Information

You can also use Tencent SCF SDK (Tencentserverless SDK), which integrates SCF business flow APIs to simplify the function invocation method and eliminates your need to encapsulate public TencentCloud APIs. For more information, please see [Calling SDK Across Functions](#).