

SDK 中心

Node.js



腾讯云

【 版权声明 】

©2013–2025 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或 95716。

Node.js

最近更新时间：2025-03-11 17:54:22

简介

欢迎使用腾讯云开发者工具套件（SDK），Node.js SDK 4.0 是云 API 3.0 平台的配套工具。为方便 Node.js 开发者调试和接入腾讯云产品 API，这里向您介绍适用于 Node.js 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 Node.js SDK 并开始调用。

依赖环境

1. Node.js 10.0.0 版本及以上。
2. 注意：部分产品需在 [腾讯云控制台](#) 开通服务后才能使用。

获取安装

使用 SDK 需要 API 密钥，可前往 [腾讯云控制台 - 访问密钥](#) 页面申请，API 密钥包括 SecretID 和 SecretKey，密钥须严格保管，避免泄露。

通过 npm 安装

通过 npm 获取安装是使用 Node.js SDK 的推荐方法，关于 npm 详细可参考 [NPM 官网](#)。

安装指定产品 SDK（推荐）

安装 CVM SDK：

```
npm install tencentcloud-sdk-nodejs-cvm --save
```

安装 VPC SDK：

```
npm install tencentcloud-sdk-nodejs-vpc --save
```

具体产品的缩写表请参考 [products.md](#) 中的包名字段。

安装指定产品 SDK 后，注意修改引入的包名：

```
- const tencentcloud = require("tencentcloud-sdk-nodejs")
+ const { cvm } = require("tencentcloud-sdk-nodejs-cvm")

- const CvmClient = tencentcloud.cvm.v20170312.Client
+ const CvmClient = cvm.v20170312.Client
```

安装全产品 SDK

```
npm install tencentcloud-sdk-nodejs --save
```

全产品 SDK 包含了所有云产品的调用代码，体积偏大，对体积敏感的场景，推荐安装指定产品 SDK。

通过源码包安装

1. 前往 [Github 仓库](#) 或者 [Gitee 仓库](#) 下载源码压缩包。
2. 解压源码包到您项目合适的位置，例如 `sdk/tencentcloud-sdk-nodejs`。
3. 执行 `npm install ./sdk/tencentcloud-sdk-nodejs`。
4. 使用 `require("tencentcloud-sdk-nodejs")` 的方式引入 SDK，具体可参考示例。

示例

以云服务器产品查询可用区列表接口为例。

```
const tencentcloud = require("tencentcloud-sdk-nodejs")

// 导入对应产品模块的client models。
const CvmClient = tencentcloud.cvm.v20170312.Client

// 实例化要请求产品(以cvm为例)的client对象
const client = new CvmClient({
  // 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。
  credential: {
    secretId: process.env.TENCENTCLOUD_SECRET_ID,
    secretKey: process.env.TENCENTCLOUD_SECRET_KEY,
  },
  // 产品地域
  region: "ap-shanghai",
  // 可选配置实例
  profile: {
    signMethod: "TC3-HMAC-SHA256", // 签名方法
    httpProfile: {
      reqMethod: "POST", // 请求方法
      reqTimeout: 30, // 请求超时时间，默认60s
      // proxy: "http://127.0.0.1:8899" // http请求代理
    },
  },
})
```

```
})  
// 通过client对象调用想要访问的接口 (Action)，需要传入请求对象 (Params) 以及响应回调函数  
// 即: client.Action(Params).then(res => console.log(res), err =>  
console.error(err))  
// 如: 查询云服务器可用区列表  
client.DescribeZones().then(  
  (data) => {  
    console.log(data)  
  },  
  (err) => {  
    console.error("error", err)  
  }  
)  
)
```

在支持 typescript 项目中，采用如下方式调用。

```
import * as tencentcloud from "tencentcloud-sdk-nodejs"  
  
const CvmClient = tencentcloud.cvm.v20170312.Client  
  
// ...
```

实例化 `Client` 的入参支持 `clientConfig` 数据结构和说明，详情请参见 [ClientConfig](#)。

Common Client

从 4.0.714 版本开始，腾讯云 Node.js SDK 支持使用泛用性的 API 调用方式(Common Client)进行请求。您只需要安装 `tencentcloud-sdk-nodejs-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的接口所需参数，否则可能会调用失败。

详细使用请参阅示例：[使用 Common Client 进行调用](#)。

更多示例

请参考 [examples](#) 目录。

相关配置

代理

如果是有代理的环境下，需要配置代理，请在创建 Client 时传入 `profile.httpProfile.proxy` 参数，或设置系统环境变量 `http_proxy`，否则可能无法正常调用，抛出连接超时的异常。

凭证管理

除显式传入凭证外，从 `v4.0.506` 起支持 [腾讯云实例角色](#)。

在您为实例绑定角色后，您可以在实例中访问相关元数据接口获取临时凭证。用法可参考 [js示例代码](#) 或 [ts示例代码](#)。

```
// ...
const CvmRoleCredential = require("tencentcloud-sdk-
nodejs/tencentcloud/common/cvm_role_credential").default

new XxxClient({
  // ...
  credential: new CvmRoleCredential(),
  // ...
})
```

常见问题

- webpack 打包出错/浏览器报错

请务必不要将此 SDK 直接用于 Web 前端（包括小程序等），暴露密钥在这些环境非常不安全。

正确做法是在自己的服务端引用此 SDK，并保存好密钥，做好请求鉴权；前端再调用服务端执行业务流程。

- The "original" argument must be of type Function.

通常是因为 Node.js 版本低于 10.0.0，或处于非 Node.js 环境，请再次确认执行环境。

- 请求不通

设置环境变量 `NODE_DEBUG=http` 来开启请求日志输出来定位问题，例如：

```
# MacOS
NODE_DEBUG=http node app.js
# windows cmd
set NODE_DEBUG=http & node app.js
# windows powershell
$env:NODE_DEBUG='http' ; node app.js
```

如需要配置代理，请查阅上文代理配置，或设置环境变量 `http_proxy`，例如：

```
# MacOS
http_proxy=http://代理地址:代理端口 node app.js
```

- 整数类型值超出 JavaScript 最大安全整数 使用新的数据类型 `BigInt`。