



梧桐链基础版  
SDK  
部署说明



苏州同济区块链研究院  
Suzhou Tongji Blockchain Research Institute

# 目录

- 一、 介绍.....1
- 二、 操作系统.....1
- 三、 **SDK 必备文件解析** .....2
- 四、 启动 **SDK** 具体步骤 .....3
- 五、 检测部署是否成功.....4
- 六、 节点常见问题及解决方法.....4



## 一、 介绍

SDK 为开发者提供区块信息写入、查询、读取等操作，使得接入梧桐链的难度大大降低。同时提供 HTTP Restful 的应用网关，使得应用系统的接入更加简单、灵活。SDK 接口使用具体说明详见 [API 文档](#)。

## 二、 操作系统

### 1. 最低配置

配置选项	配置要求
操作系统	Linux/Windows/mac
内存	$\geq 4\text{G}$
CPU	$\geq 2\text{C}$
硬盘	$\geq 100\text{G}$
网络	百兆及以上

注：本文所示操作均以 ubuntu 16.04 为例

### 2. 推荐配置

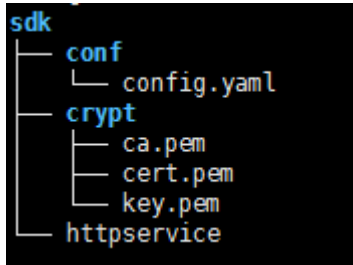
配置选项	配置要求
操作系统	Linux 操作系统内核 3.1 以上，64 位，推荐使用 ubuntu 16.04.3
内存	$\geq 8\text{G}$
CPU	$\geq 4\text{C}$
硬盘	$\geq 1\text{TB}$
网络	百兆及以上

注：本文所示操作均以 ubuntu 16.04 为例



### 三、 SDK 必备文件解析

SDK 目录文件：



配置文件 **config.yaml**（目录 **./sdk/conf/config.yaml**）

*# http 监听端口，端口可根据网络实际情况修改，默认为 8080，一般 HTTP 端口为 80。*

**port: 8080**

*#核心节点，此处设置区块链所有节点服务器的 IP 地址和通讯端口；SDK 服务器需要能和节点服务器正常链接；如果需要在 SDK 服务器和节点之间使用 TLS 通道加密，则设置参数 **usetls** 为 **true**。*

**[[peer]]**

**address = "Peer IP:Port"**

**usetls = false**

**[[peer]]**

**address = " Peer IP: Port "**

**usetls = false**

*#证书配置，若 **peer** 的参数设置中 **usetls** 为 **true**，则此处需要配置相应证书。证书包含根证书，以及该 SDK 服务器获得的证书文件和密钥。证书的加密签名算法必须和节点服务器保持一致，梧桐链支持标准算法和国密算法。**ca.pem** 为根证书，与 **peer** 节点 TLS 的 **ca** 证书一致，**cert.pem**，**key.pem** 分别代表用户的数字签名证书和私钥。*

**[rpc]**

**TLSCaPath = "./crypt/ca.pem"**

**TLSCertPath = "./crypt/cert.pem"**

**TLSKeyPath = "./crypt/key.pem"**

**TLSServerName = "test.example.com"**

**HashTyp = "sm3" # hash 算法，支持 sm3, sha256**

*#消息摘要算法，消息摘要算法必须和证书的加密算法类型匹配。此处*



参数值可为 *sm3* 和 *sha256*，分别对应国密和标准算法。

KeyType = "sm2" # 加密算法，支持 sm2,ecc

#非对称加密算法，非对称加密算法必须和证书的加密算法类型匹配。

此处参数值可为 *sm2* 和 *ecc*，分别对应国密和标准算法。

Timeout = "30s" # 超时时间

# 钱包配置

[wallet]

Enable = true

DupDuration= "3s" # 重复数据检测时间范围，重复数据检查策略为  
ByTime 时有效

TxTimeout= "10m" # 交易上链超时时长

SyncInterval= "1s"

# 监控服务端口

[monitor]

port = 9090

# 智能合约

[smartcontract]

# 合约类型，支持 docker

category = "docker"

## 四、 启动 SDK 具体步骤

### 1. 证书生成

SDK 的 TLS 验证证书可选。

#### 1) TLS 验证证书 (crypt 文件夹，目录：./peer/crypt)

A. 按照[节点部署说明中的七、生成证书](#)生成多套证书密钥对，SDK 和节点使用的是同一套 TLS 验证的根证书生成的密钥对。

B. 将生成的文件移动到 crypt 文件夹中。

crypt 文件夹必备 ca.pem (CA 证书)，cert.pem (证书)，key.pem (私钥)。

如果开启 TLS 认证，则每个节点和 sdk 都分别需要一对同一个 CA 根证书生成的不同的密钥对。

#### 2) 证书文件名应当与 SDK 配置文件中 TLS 证书配置一致，可以修改文件名，也可以修改配置文件。

修改文件名：

mv [当前文件名] [配置文件中的文件名]

mv cert1.pem cert.pem



2. 更改配置文件（配置文件解析具体[三、SDK 配置文件解析](#)。更改节点地址，是否启用 TLS 等配置）

```
cd conf
vim config.yaml
```

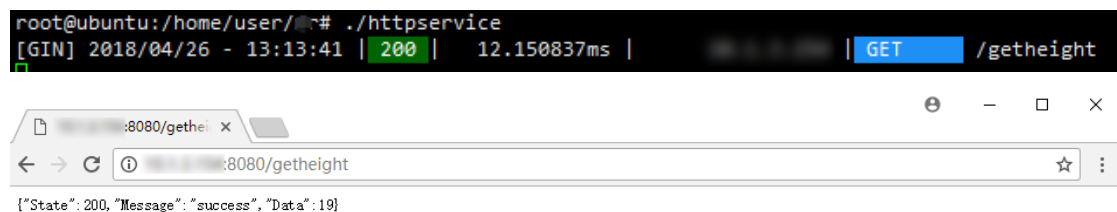
3. 后台启动命令

此处为 Linux 系统启动命令，mac/Windows 系统环境直接启动程序即可。

```
nohup ./httpservice &
```

## 五、检测部署是否成功

1. 可通过在浏览器输入 `http://服务器 IP 地址:端口/getheight`，如果有返回区块链高度说明部署成功。



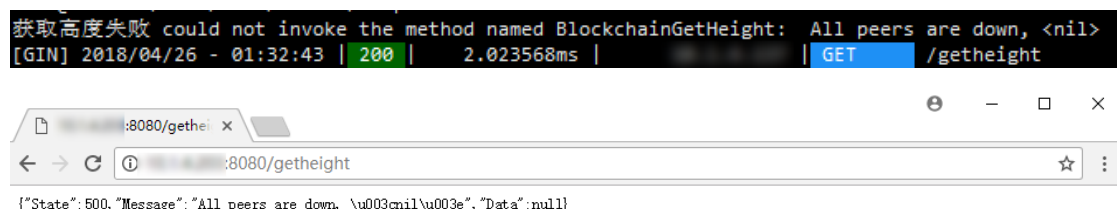
2. 在部署 SDK 的服务器中输入 `curl http://127.0.0.1:端口/getheight`，如果有返回区块链高度说明部署成功。

```
root@ubuntu:~# curl http://127.0.0.1:9999/getheight
{"State":200,"Message":"success","Data":102}root@ubuntu:~# ^C
```

## 六、节点常见问题及解决方法

1. All peers are down, <nil>

报错信息：



解决方法：

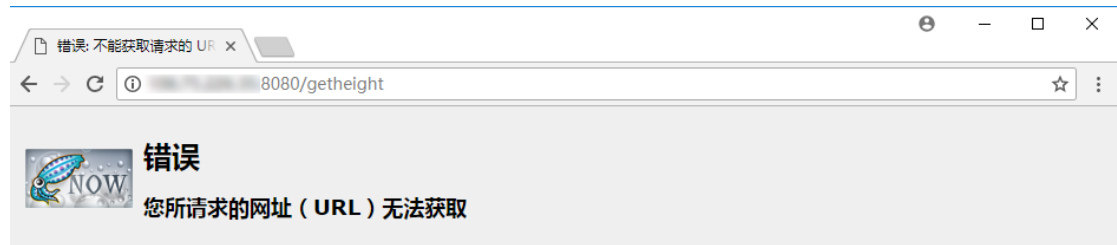
检查节点是否启动或节点已断开



配置文件错误，没有连接上节点，检查配置文件中 **peer** 地址和端口是否正确配置。

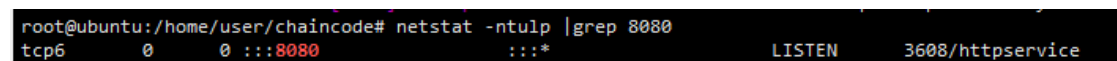
## 2. 连接超时，无法获取 URL

报错信息：



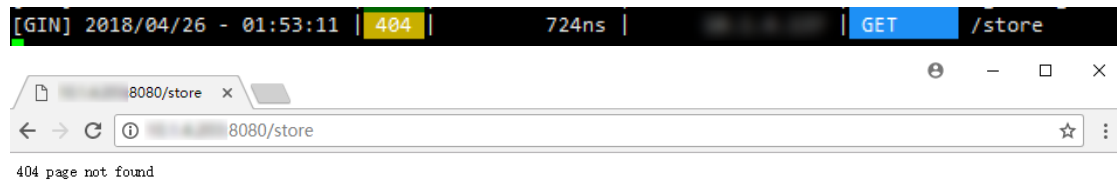
解决方法：

检查端口是否开放，应开放端口。使用命令 `netstat -ntulp | grep 8080` 查看端口是否开放，若已开放，则显示如下图。



## 3. 404 page not found

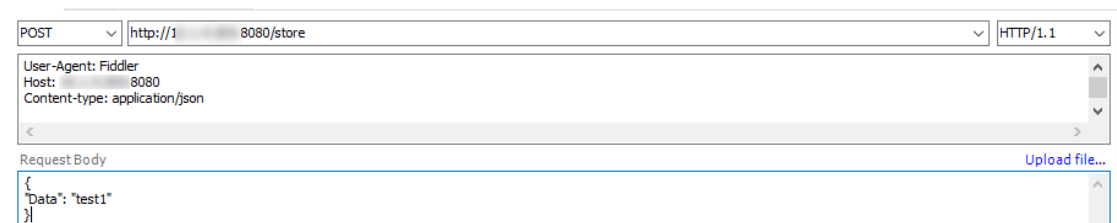
报错信息：



解决办法：

SDK 中 **store** 存证交易等须使用 **post** 传入 **json** 参数（传入的参数格式详见 **sdk** 文档）。

传入参数：



返回结果：

```
{"State":200,"Message":"success","Data":{"Figure":"uN5rIT0lR7cE/e5J/qcY5qybddOTazmnOJB1636NtVA=","OK":true}}
```