cds节点资料汇总

内部文件：V.0.0.1

颁布时间：2019-11-08

**目 录**

[ 文件版本说明 2](#_Toc8016)

[ 参考资料 2](#_Toc543)

[ 手册目的 2](#_Toc14334)

[ 声明 2](#_Toc26539)

[ 名词定义和缩略语说明 2](#_Toc24634)

[1 下载和编译 3](#_Toc11640)

[1.1 开源地址 3](#_Toc30707)

[1.2 编译 3](#_Toc15635)

[1.2.1 安装JDK 3](#_Toc3695)

[1.2.2 安装编译工具Maven3 3](#_Toc19666)

[1.2.3 编译 3](#_Toc31795)

[1、 启动节点 4](#_Toc26789)

[1.1、 配置文件 4](#_Toc17661)

[2.1.1、启动端口以及同步节点 4](#_Toc22168)

[2.1.2、节点数据存放路径 4](#_Toc19152)

[2.1.3、rpc功能 4](#_Toc23613)

[2.2、启动节点 5](#_Toc25123)

[3 节点接口 5](#_Toc21658)

[3.1 使用web3链接节点 5](#_Toc6314)

[3.2 查询地址余额 5](#_Toc16974)

[3.3 获得交易随机数nonce 6](#_Toc19289)

[3.4 创建节点钱包 6](#_Toc10244)

[3.5 获取节点钱包地址 6](#_Toc19856)

[3.6 使用节点钱包发送交易 7](#_Toc374)

[3.7 使用外部钱包发送交易 8](#_Toc12944)

[4 一些值得注意的地方 9](#_Toc18828)

[4.1 使用windows服务器报无法加载dll动态链接库 9](#_Toc5032)

[4.2 使用windows服务器,发现P2P网络不稳定 9](#_Toc7766)

[4.3 区块链浏览器 9](#_Toc16571)

[4.4 关于交易确认 9](#_Toc23024)

[4.5 关于智能合约 9](#_Toc30225)

* 文件版本说明

表 1 版本说明

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 发布时间 | 修订章节 | 作者 |
| V.0.0.1 | 2019-11-08 | 汇总材料新建 | Prings |
|  |  |  |  |
|  |  |  |  |

* 参考资料

[1] https://jingyan.baidu.com/article/e8cdb32b2699cb37042bad59.html

[2] https://blog.csdn.net/williamyi96/article/details/78268595

[3] https://www.cnblogs.com/xixihuang/p/5569203.html

* 手册目的

节点资料汇总,用于上币过程中的技术说明和问题解答

* 声明

可自由使用和传播

* 名词定义和缩略语说明

表 2 名词定义及缩略语说明

| 序号 | 缩写 | 说明 |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# 下载和编译

## 1.1 开源地址

暂无

## 1.2 编译

### 1.2.1 安装JDK

节点主体使用java语言开发,因此,您需要确保正确安装Java8(或以上版本,OpenJdk或oracle Jdk均测试运行通过),安装详情请参考:

Windows: https://jingyan.baidu.com/article/e8cdb32b2699cb37042bad59.html

linux(以ubantu为例): https://blog.csdn.net/williamyi96/article/details/78268595

### 1.2.2 安装编译工具Maven3

编译代码的工具是Maven3

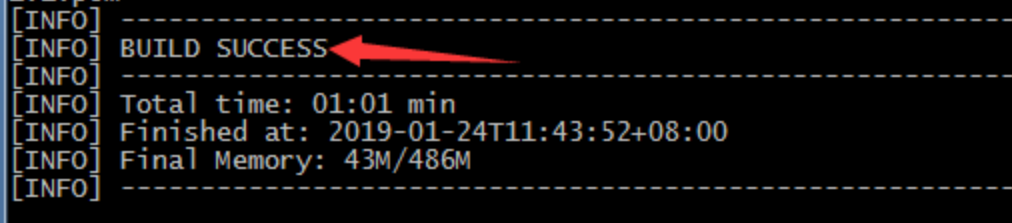
安装详情请参考:https://www.cnblogs.com/xixihuang/p/5569203.html

### 1.2.3 编译

从githup上下载源代码,解压,进入代码目录cds-main-blockchain-node

在windows cmd或者linux终端下进入安装目录

执行mvn install命令即可执行代码编译,编译成功之后,出现如图所示的



编译执行完毕后会在目录下出现target文件夹

进入该文件夹可以看到cds-main-blockchain-node-0.0.1.jar.即为产品包

# 启动节点

## 配置文件

在代码目录下，存在外部配置文件blockchain.conf，用来控制节点的行为，具体如下：

### 2.1.1、启动端口以及同步节点

peer {

listen.port = 30777 #启动端口

active = [ #活跃节点 更多的节点意味着更快的同步速度

{

ip = ""

port = 30777

nodeId = ""

}

]

trusted = [ #信任节点 更多的节点意味着更快的同步速度

{

ip = ""

nodeId = ""

}

]

}

### 2.1.2、节点数据存放路径

database {

dir = "./database" #数据存放路径

}

### 2.1.3、rpc功能

modules {

rpc {

enabled = true #是否启用rpc功能

port = 8866

ipWhiteList=["192.168.\*.\*","47.75.96.\*"] #白名单

}

}

## 2.2、启动节点

启动节点的命令:

Windows:

java -jar -Xms1024m -Xmx4096m cds-blockchain-node-0.0.1.jar

linux(以ubantu为例):

nohup java -jar -Xms1024m -Xmx4096m cds-blockchain-node-0.0.1.jar >/dev/null 2>&1 &

注：产品包所在目录下，创建config文件夹 用来放置配置文件blockchain.conf

# 3 节点接口

节点继承以太坊并实现了大部分web3的接口(当然您使用web3j也是兼容的),在调用web3之前,请确认外部配置user.conf已经启动了json rpc模块并且正确配置了IP白名单.

以下所有示例代码均使用web3.js(web3J原理上相同),nodejs依赖配置如下:

{

"name": "web3 test",

"private": true,

"version": "0.0.1",

"engines": {

"node": ">=5.0.0"

},

"devDependencies": {

"web3": "^1.0.0-beta.34",

"solc": "^0.4.22",

"ethereumjs-tx": "^1.3.7"

},

}

## 3.1 使用web3链接节点

链接web3目前的方式支持http,节点代码如下

// 加载web3模块

var Web3 = require('web3');

// 创建web3链接

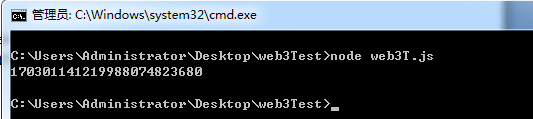
var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

## 3.2 查询地址余额

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

web3.eth.getBalance("0x55b6f781cc8dd985953ef36a7dce36ea4a09109f").then(console.log);



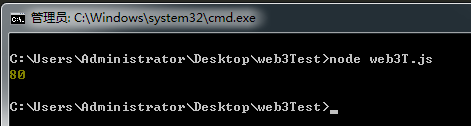
余额跟以太坊一样,18位精度,可以通用web3的toWei和fromWei方法

## 3.3 获得交易随机数nonce

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

web3.eth.getTransactionCount("0x55b6f781cc8dd985953ef36a7dce36ea4a09109f","pending").then(console.log);

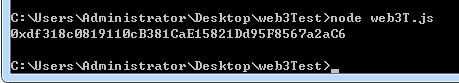


## 3.4 创建节点钱包

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

web3.eth.personal.newAccount('!@superpassword').then(console.log);



## 3.5 获取节点钱包地址

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

web3.eth.getAccounts().then(console.log);



## 3.6 使用节点钱包发送交易

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

var fromAddress = "0x543b6e1125c84810c9f714f81147b553Eb2B2977";//节点钱包地址

var receiverAccount = "0xdf318c0819110cB381CaE15821Dd95F8567a2aC6";

//解锁节点钱包

web3.eth.personal.unlockAccount(fromAddress, "test").then(openAccountState=>{

if(openAccountState){

web3.eth.sendTransaction({

from: fromAddress,

to: receiverAccount,

value: web3.utils.toWei("0.1", "ether"),

gasLimit: "21000",

gasPrice: web3.utils.toWei("15", "gwei")

}).then(function(receipt){

console.log(receipt);

});

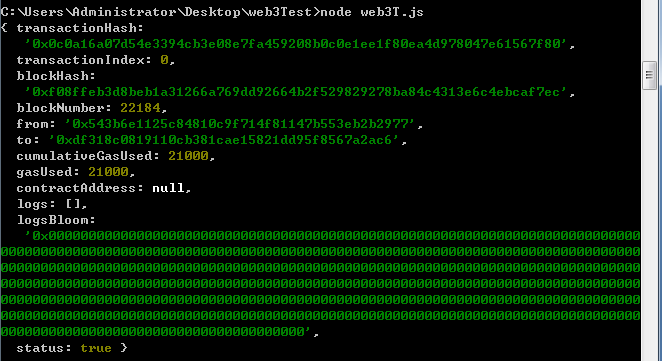
}else{

console.log("节点账户解锁失败");

}

});

值得注意的是,cds节点更换了挖矿算法,所以出块速度大约为2分钟每块,也就是说获得交易回执的时间大约是2分钟,所以运行该代码最长需要2分钟会出现如下输出,请您耐心等待



## 3.7 使用外部钱包发送交易

对于一些安全级别较高的应用,钱包一般不会交由节点进行管理,您可以使用任何以太坊钱包生成工具和签名工具,来生成钱包地址和签名交易,但无论何种方法,最终,您需要使用web3.eth. sendSignedTransaction来发送交易,样例代码如下:

var Web3 = require('web3');

var web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8866"));

var Tx = require('ethereumjs-tx');

//地址密钥

var privateKey = Buffer.from('a6b6f705ae5bb91c3ad1049995490c028d7c1d64f116e740a9fdc5bd49c75c1b', 'hex');

//发送地址

var fromAddress = "0x578174A9303c5D48Ca7cB2450033Ef08C450dCCE";

//接受地址

var receiverAccount = "0xdf318c0819110cB381CaE15821Dd95F8567a2aC6";

var rawTx = {

from: fromAddress,

to: receiverAccount,

value: web3.utils.toHex(web3.utils.toWei("0.01", "ether")),

gasLimit: web3.utils.toHex(21000),

gasPrice: web3.utils.toHex(web3.utils.toWei("15", "gwei"))

}

var tx = new Tx(rawTx);

tx.sign(privateKey);

var serializedTx = tx.serialize();

web3.eth.sendSignedTransaction('0x' + serializedTx.toString('hex'),function(err, hash){

if (!err) {

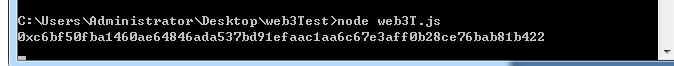
console.log(hash);

} else {

console.error(err);

}

});



# 4 一些值得注意的地方

## 4.1 使用windows服务器报无法加载dll动态链接库

如果您的服务器是windows,节点需要一些额外的动态链接库,一般您只需要安装Visual Studio 2017即可.

## 4.2 使用windows服务器,发现P2P网络不稳定

如果进程出现节点重连的情况,请您注意服务器的时间戳是否正确,windows同步时间戳的方法网络上很多,请自行参考.

## 4.3 区块链浏览器

http://47.75.96.91/

## 4.4 关于交易确认

为了避免因区块分叉而回滚交易的情况,我们建议您需要再节点中进行交易的二次确认.根据目前的交易量情况,二次确认的区块时间可以是当前最新块的前6至8块,如果节点已经生成交易回执信息且状态成功,则可以认为交易已经成功.

## 4.5 关于智能合约

智能合约在solc4.25下编译并部署测试成功,内部核对过EVM字节码无误