

起多节点树状区块链

```
"config": {
 "chainId": 7,
 "homesteadBlock": 0,
 "eip150Block": 0,
 "eip155Block": 0,
 "eip158Block": 0,
 "byzantiumBlock": 0,
 "constantinopleBlock": 0,
 "petersburgBlock": 0
"alloc": {
00000000000000"},
00000000000000"},
},
"difficulty": "0x20000",
"extraData": "",
"gasLimit": "0xffffffff",
"nonce": "0x00000000000000042",
"timestamp": "0x00"
}
```

事件监听需要通过websocket服务,ganache-cli提供的测试环境无法提供socket服务,故直接使用geth创建私有链,同时启动websocket,参数设置要如下完全一致:

(注意rpcport和wsport(是websocket通信的端口))

```
# 初始化
geth1 --datadir ./gethdata --networkid 91036 --port 30303 --rpc --rpcaddr 127.0.0.1 --
rpcport 8545 --rpcapi 'db,net,eth,web3,personal' --rpccorsdomain "*" --ws --wsaddr "lo
```

起多节点树状区块链 1

```
calhost" --wsport "8546" --wsorigins "*" --nodiscover --allow-insecure-unlock --dev.pe riod 1 --syncmode "full" init ./genesis.json
```

```
# 启动
geth1 --datadir ./gethdata --networkid 91036 --port 30303 --rpc --rpcaddr 127.0.0.1 --
rpcport 8545 --rpcapi 'db,net,eth,web3,personal' --rpccorsdomain "*" --ws --wsaddr "lo
calhost" --wsport "8546" --wsorigins "*" --nodiscover --allow-insecure-unlock --dev.pe
riod 1 --syncmode "full" console
```

```
#第二个节点
#初始化
geth1 --datadir ./gethdata --networkid 91036 --port 30304 --rpc --rpcaddr 127.0.0.1 --
rpcport 8547 --rpcapi 'db,net,eth,web3,personal' --rpccorsdomain "*" --ws --wsaddr "lo
calhost" --wsport "8548" --wsorigins "*" --nodiscover --allow-insecure-unlock --dev.pe
riod 1 --syncmode "full" init ./genesis.json

#启动
geth1 --datadir ./gethdata --networkid 91036 --port 30304 --rpc --rpcaddr 127.0.0.1 --
rpcport 8547 --rpcapi 'db,net,eth,web3,personal' --rpccorsdomain "*" --ws --wsaddr "lo
calhost" --wsport "8548" --wsorigins "*" --nodiscover --allow-insecure-unlock --dev.pe
riod 1 --syncmode "full" console
```

```
# 创建账户
personal.newAccount("123456")
personal.newAccount("123456")
personal.newAccount("123456")

# 解锁账户
personal.unlockAccount(eth.accounts[0],"123456",15000)
personal.unlockAccount(eth.accounts[1],"123456",15000)
personal.unlockAccount(eth.accounts[2],"123456",15000)
personal.unlockAccount(eth.accounts[3],"123456",15000)

# 查看账户余额
eth.getBalance(eth.accounts[0])
```

获取第一个节点信息 在1的控制台中输入:

起多节点树状区块链 2

> admin.nodeInfo.enode

第二个节点添加c1为静态节点,在c2中输入:

> admin.addPeer("")

//注意,将其中的0.0.0.0替换为c1的IP地址 admin.addPeer("enode://82e2a1e2d86906266625fc20104 6aa4a195d897a00e5cdb3e29765ce805be4aeb360861d7e8b46dcd455aa406101e795942f49e9857fa6d64 0f5d850fef2cca7@10.108.21.132:30303")

返回true,并用

> admin.peers

查看是否互相链接,若连接成功,则显示节点c1的信息。由于此时网络中只有2个节点,也可以用

> net.peerCount

查看当前网络中的对等节点个数,显示结果为

> 1

PS:节点重启后,需要再次执行上述admin.addPeer命令。

查验同步在c1创建账号,成功后,开始挖矿,此时c2中会同步显示区块验证信息。

起多节点树状区块链 3