

# 树状区块链启动和创建节点

```
# geth
personal.newAccount("123456") # 创建账户
eth.accounts[0]
personal.unlockAccount(eth.accounts[1], "123456", 15000) # 解锁账户
personal.unlockAccount(eth.accounts[1], "123456", 15000) # 解锁账户
personal.unlockAccount(eth.accounts[2], "123456", 15000) # 解锁账户
eth.blockNumber # 查看区块数
eth.accounts # 枚举用户
eth.getBalance() # 查看账户余额
eth.getBlock() # 获取区块
admin.addPeer() # 添加其他节点
miner.start(1) # 挖矿
miner.stop() # 停止
```

# 复现

树状区块链需要使用修改后以太坊geth客户端:

• 将新的geth文件(设置了树状区块链的文件)替换本地原有的geth位置

```
[ifpop@arch ~]$ whereis geth
geth: /usr/bin/geth
[ifpop@arch ~]$ sudo cp /usr/bin/geth /usr/bin/geth1
[ifpop@arch ~]$ sudo cp ./geth /usr/bin/geth
```

这样操作就可以使用两种 geth ,测试是否更换成功:

```
[ifpop@arch ~]$ geth version
Geth
Version: 1.9.12-stable
Git Commit: fe8bdfb728b96fd445511ed71932ef67229e2738
Git Commit Date: 20210425
Architecture: amd64
Protocol Versions: [65 64 63]
Go Version: go1.13.9
Operating System: linux
GOPATH=
GOROOT=/usr/local/go
[ifpop@arch ~]$ geth1 version
Geth
Version: 1.10.3-stable
Git Commit: 991384a7f6719e1125ca0be7fb27d0c4d1c5d2d3
Git Commit Date: 20210505
Architecture: amd64
Go Version: go1.16.3
Operating System: linux
GOPATH=
GOROOT=go
```

# 部署合约之前先创建节点并初始化余额

- 为什么要有余额呢?因为在区块链的操作是会消耗余额的,所以先初始化足够的余额
- 在 genesis.json 所在文件夹下,初始化和启动移动区块链(如果启动报错(bind: invalid argument),是因为文件夹的路径 太深了,剪切到浅一点的文件夹就好了)

```
# 初始化区块链
$ geth --identity "MyEth" --rpc --rpcaddr 127.0.0.1 --rpcport "8545" --rpccorsdomain "*" --datadir gethdata --port "30303" --n
odiscover --rpcapi "eth,net,personal,web3" --networkid 91036 init genesis.json
# 启动区块链
$ geth --identity "MyEth" --rpcaddr 127.0.0.1 --rpc --rpcport "8545" --rpccorsdomain "*" --datadir gethdata --port "30303" --no
discover --rpcapi "eth,net,personal,web3" --networkid 91036 --allow-insecure-unlock --dev.period 1 console
```

如果同时开两个区块链节点的话,第二个rpcport 设置为8546,port 设置为30304,第一个部署map合约,第二个部署traffic合约

```
# 初始化

$ geth --identity "MyEth" --rpc --rpcaddr 127.0.0.1 --rpcport "8546" --rpccorsdomain "*" --datadir gethdata --port "30304" --nodis

cover --rpcapi "eth,net,personal,web3" --networkid 91036 init genesis.json

# 启动

$ geth --identity "MyEth" --rpcaddr 127.0.0.1 --rpc --rpcport "8546" --rpccorsdomain "*" --datadir gethdata --port "30304" --nodisc

over --rpcapi "eth,net,personal,web3" --networkid 91036 --allow-insecure-unlock --dev.period 1 console
```

### 执行完上述两个命令后就进入了控制台:

```
Welcome to the Geth JavaScript console!

instance: Geth/MyEth/v1.9.12-stable-fdd6b536/linux-amd64/go1.13.9

coinbase: 0x87e2d88cd788e8b8be2682e356cb1f81bec3116e

at block: 0 (Thu Jan 01 1970 08:00:00 GMT+0800 (CST))

datadir: /home/ubuntu/桌面/db-udgraduate/undergraduate-dbin2021/src/MapStore_VN/gethdata

modules: admin:1.0 debug:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0
```

### 在控制台创建账户,参考命令:

```
> personal.newAccount("123456")
```

#### 控制台返回

```
INFO [05-29|10:34:46.970] Your new key was generated address=0x87E2D88cd788E8B8BE2682E356CB1f81BEC3116E
WARN [05-29|10:34:46.971] Please backup your key file! path=/home/ubuntu/桌面/db-udgraduate/undergraduate-dbin2021/src/MapS
WARN [05-29|10:34:46.971] Please remember your password!
"0x87e2d88cd788e8b8be2682e356cb1f81bec3116e"
```

上述返回内容中绿色字符串为账户名称,必须copy记录一下,(ctrl+shift+c)

下一步,退出Geth控制台(注意不是关闭窗口),参考命令:

```
> exit
```

此时节点账户已经创建好,下一步将账户名称粘贴到 genesis.json , alloc字段即为用户账户信息字段,但是原有的账户信息并不是在你刚创建的区块链私链上,所以将账户名称改成刚刚创建账户返回的唯一ID

```
{
 "config": {
  "chainId": 91036,
 "homesteadBlock": 0,
 "eip150Block": 0,
 "eip155Block": 0.
 "eip158Block": 0.
 "byzantiumBlock": 0,
 "constantinopleBlock": 0,
 "petersburgBlock": 0
 3456789", "txtime":1}
},
 "difficulty": "0x20000",
 "extraData": ""
"gasLimit": "0xffffffff",
"timestamp": "0x00"
```

紫色字符串为账户名称要粘贴的位置,更改之后保存,进行下一步,将初始化后产生的gethdata文件夹里的geth删除,参考命令:

```
$ ls
$ cd gethdata
$ ls
$ rm -rf geth
$ cd ..
```

再次初始化,就会读 genesis.json 文件中的相关配置,赋予账户余额(balance字段),参考命令:

```
geth --identity "MyEth" --rpc --rpcaddr 127.0.0.1 --rpcport "8545" --rpccorsdomain "*" --datadir gethdata --port "30303" --nodisco ver --rpcapi "eth,net,personal,web3" --networkid 91036 init genesis.json
```

### 进入控制台,参考命令:

```
geth --identity "MyEth" --rpcaddr 127.0.0.1 --rpc --rpcport "8545" --rpccorsdomain "*" --datadir gethdata --port "30303" --nodiscov er --rpcapi "eth,net,personal,web3" --networkid 91036 --allow-insecure-unlock --dev.period 1 console
```

#### 查看账户余额

```
> eth.getBalance(eth.accounts[0])
5e+40
```

# 部署合约

#### 在控制台,首先解锁账户

```
>personal.unlockAccount(eth.accounts[0],"123456",15000)
true
```

# • 信誉值计算合约部署

//部署合约contract在移动区块链上,将下述代码直接粘贴到控制台  $abi = JSON.parse('[{\"constant\":false, \"inputs\":[{\"internalType\":\"string\", \"name\":\"uuid\", \"type\":\"string\"}], \"name\":\"and \"inputs\":[{\ inputs\":[{\ inputs\":[[{\ inputs\":[{\ inputs\":[[inputs\":[[inputs\":[[inputs\":[[inputs\":[[inputs\":[[inputs\":[[input$  $\label{thm:control} $$ \":'"revalue\", `"outputs\":[], "payable\":false, \"stateMutability\":\"nonpayable\", \"type\":\"function\"}, {\"constant\":true, \"inputs\":[{\"internalType\":\"string\", \"name\":\"getQuality\", \"outputs\":[{\"internalType\":\"string\"}], \"name\":\"getQuality\", \"outputs\":[{\"internalType\":\"string\"}], \"name\":\"getQuality\", \"outputs\":[{\"internalType\":\"string\"}], \"name\":\"getQuality\", \"outputs\":\"internalType\":\"in$ \"outputs\":[{\"internalType\":\"uint256\",\"name\":\"\",\"type\":\"uint256\"},{\"internalType\":\"uint256\",\"name \":\"\",\"type\":\"uint256\"},{\"internalType\":\"uint256\",\"name\":\"\",\"type\":\"uint256\"},{\"internalType\":\"uint256\"}, \",\"name\":\\",\"type\":\\"uint256\\"}],\\"payable\\":false,\\"same\\":\\\",\\"type\\":\\"uint256\\"}],\\"payable\\":false,\\"same\\":\\\",\\"type\\\":\\\"uint256\\\"}],\\"payable\\":false,\\\"same\\":\\\",\\\"type\\\":\\\"uint256\\\"}],\\\"payable\\\":false,\\\"same\\":\\\",\\\"type\\\":\\\"uint256\\\"}],\\\"payable\\\":false,\\\"same\\":\\\",\\\"type\\\":\\\"uint256\\\"],\\\"payable\\\":false,\\\"same\\":\\\",\\\"type\\\":\\\"uint256\\\"],\\\"payable\\\":false,\\\"same\\":\\\",\\"type\\\":\\\"uint256\\\"],\\\"payable\\\":false,\\"same\\":\\"uint256\\",\\"type\\":\\"uint256\\"],\\\"payable\\":false,\\"same\\":\\"uint256\\",\\"type\\":\\"uint256\\"],\\"payable\\":false,\\"same\\"same\\";  $tate Mutability \":\"view\", \"type\":\"function\"\, \\"constant\":false, \"inputs\":[{\"internalType\":\"string\", \"name\":\"uuid\\",\"type\":\"string\"\, {\"internalType\":\"uint256\", \"name\":\"time\", \"type\":\"uint256\"\, {\"internalType\":\"uint256\", \"name\":\"_lonOri\", \"type\":\"uint256\"\, {\"internalType\":\"uint256\"\, \"name\":\"_lonOri\", \"type\":\"uint256\"\, \\"internalType\":\"uint256\"\, \"name\":\"_lonOri\", \"type\":\"uint256\"\, \\"internalType\":\"uint256\"\, \\"internalType$ e\":\"uint256\",\"name\":\"\_latFix\",\"type\":\"uint256\"},{\"internalType\":\"uint256\",\"name\":\"\_lonFix\",\"type\":\"uint256\",\"name\":\"\_lonFix\",\"type\":\"internalType\":\"uint256\",\"name\":\"\_lonFix\",\"type\":\"internalType\":\"internalType\":\"internalType\":\"internalType\":\"internalType\":\"lonFix\",\"type\":\"function\"}]') 3603602081101561006757600080fd5b810190808035906020019064010000000081111561008457600080fd5b82018360208201111561009657600080fd5b803590602001918460018302840111640100000000831117156100b857600080 fd5b91908080601 f016020809104026020016040519081016040528093929190 $818152602001838380828437600081840152601 \\ f19601f820116905080830192505050505050919291929050506103b3565b60405180828152602001915$ 6020019092919080359060200190929190803590602001909291908035906020019092919080359060200190929190803590602001909291905050506104b1565b005b6102c66004803603602081101561022357600080fd5b81019080803590602001906401000000081111561024057600080fd5b82018360208201111561025257600080fd5b80 35906020019184600183028401116401000000008311171561027457600080 fd 5b91908080601 f0160208091040260200160405190810160405280939291908

 $805182526020820191506020810190506020830392506103\\cdot 8565\\b6001836020036101000\\a0380198251168184511680821785525050505050505090500191505$ 0998152602001604051809103902060020160009054906101000a900460ff1661043c57600090506104ac565b600a826040518082805190602001908083835b518252602082019150602081019050602083039250610544565b6001836020036101000a0380198251168184511680821785525050505050509050019150509060619160606191606061916060619160915060208101905060208303925061063b565b6001836020036101000a038019825116818451168082178552505050505090500191505090815260200160405190602001908083835b602083106107b65780518252602082019150602081019050602083039250610793565b6001836020036101000a0380198251168184 $5001915050908152602001604051809103902060080160008381526020019081526020016000208190555084600a\\ 886040518082805190602001908083835b6$ 808217855250505050505050905001915050908152602001604051809103902060070160008381526020019081526020016000208190555061088387610d8e565054600a886040518082805190602001908083835b60208310610bb85780518252602082019150602081019050602083039250610b95565b60018360200361010380198251168184511680821785525050505050505050509050019150509081526020016040518091039020600701600085815260200190815260200160002054600a8b6040518082805190602001908083835b60208310610d325780518252602082019150602081019050602083039250610d0f565b6001836020036101000a03 801982511681845116808217855250505050505050505090500191505090815260200160405180910390206008016000868152602001908152602001600020549550905190602001908083835b60208310610f275780518252602082019150602081019050602083039250610f04565b6001836020036101000a038019825116818465b600a856040518082805190602001908083835b602083106110ae578051825260208201915060208101905060208303925061108b565b6001836020036101000a038019825116818451168082178552505050505050505090500191505090815260200160405180910390206004016000848152602001908152602001600020501000a03801982511681845116808217855250505050505050905001915050908152602001604051809103902060050160008581526020019081526020016000200190815260200160002054600a866040518082805190602001908083835b60208310611431578051825260208201915060208101905060208303925061140e038152602001908152602001600020546000a876040518082805190602001908083835b602083106115735780518252602082019150602081019050602083039250611550565b6001836020036101000a038019825116818451168082178552505050505050905001915050908152602001604051809103902060080160008681526020019081526020016000205403115b156115ca576001840393505b6001831180156116da5750610258600a866040518082805190602001908083835b650509050019150509081526020016040518091039020600801600060018603815260200190815260200160002054600a8760405180828051906020019080838 $35 66 02083 1061 168 f578 05182526 02082 0191506 02081 0190506 02083 0392506 1166 c565 \\ b600 1836 020036 101000 \\ a0380 19825 11681845 1168082 1785525 05082 \\ b600 1836 020036 101000 \\ a0380 19825 11681845 \\ b600 1836 020036 101000 \\ a0380 19825 11681845 \\ b600 1836 020036 \\$ 180156117f657506104b0600a866040518082805190602001908083835b6020831061172a57805182526020820191506020810190506020830392506117075602001908152602001600020546000a8760405180828051906020019080838355602083106117ab57805182526020820191506020810190506020830392506117QualContract = web3.eth.contract(abi); web3.eth.estimateGas({data: bytecode})

```
Qual = QualContract.new({
    from: web3.eth.accounts[0],
    data: bytecode,
    gas: '1400000',
    position:"w251111111111",
    txtime:277001
}, function (e, contract){
    console.log(e, contract);
    if(!e){
        if(!contract.address) {
            console.log("Contract transaction send: TransactionHash: " + contract.transactionHash + " waiting to be mined...");
    } else {
        console.log("Contract mined! Address: " + contract.address);
        console.log(contract);
    }
});
```

#### • StoreMap 部署

# 部署StoreMap在移动区块链上

```
n\"},{\"constant\":true,\"inputs\":[],\"name\":\"owner\",\"outputs\":[{\"internalType\":\"address\",\"name\":\"\",\"type\":\"address\"],\"payable\":false,\"stateMutability\":\"view\",\"type\":\"function\"},{\"constant\":true,\"inputs\":[{\"internalType
me\":\"feature\",\"type\":\"int32[]\"},{\"internalType\":\"bytes32\",\"name\":\"function\",\"type\":\"bytes32\",\"internalType\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"bytes32\",\"name\":\"byte
pe'":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"bytes32[]\",\"name\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"type\":\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways\",\"highways
 \":\"bytes32[]\"},{\"internalType\":\"bytes32[]\",\"name\":\"gtypes\",\"type\":\"bytes32[]\"},{\"internalType\":\"bytes32[]
\label{limit} $$ \mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$}\mbox{$\mbox{$\mbox{$}\mbox{$}\mbox{$\mbox{$}\mbox{$\mbox{$}\mbox{$}\mbox{$\mbox{$}\mbox{$}\mbox{$}\mbox{$\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mbox{$}\mb
\":false,\"inputs\":{{\"internalType\":\"uint256\",\"name\":\"gid\",\"type\":\"uint256\"},{\"internalType\":\"int32\",\"name\":\"minzoom\",\"type\":\"int32\"},{\"internalType\":\"bool\",\"name\":\"cost\",\"type\":\"int32\"},{\"internalType\":\"bool\",\"name\":\"bool\",\"type\":\"bool\"},{\"internalType\":\"bool\",\"name\":\"bool\",\"type\":\"bool\"},{\"internalType\":\"bool\",\"name\":\"bool\",\"type\":\"bool\"},\"internalType\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"type\":\"bool\"},\"internalType\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\",\"name\":\"bool\"\",\"name\":\"bool\"\",\"name\":\"bool\"\",\"nam
":\"bytes32\",\"name\":\"highway\",\"type\":\"bytes32\"},{\"internalType\":\"bytes32\",\"name\":\"name\",\"type\":\"bytes32\",
\label{thm:continuous} $$ \space{100} $$ ("internalType":\"bytes32", \name":\"gtype\".\"bytes32"}, {\"internalType\":\"bytes32[]\",\name\":\"path of the continuous of the c
\":\\"function\"},{\"constant\":false,\\"inputs\":[{\\"internalType\\":\\"name\\":\\"name\\":\\"name\\":\\"name\\":\\"name\\":\\"name\\":\"inputs\\":[],\\"payable\\":false,\\"stateMutability\\":\\"nonpayable\\",\\"type\\":\\"internalType\\":\\"add_area_line\\",\\"outputs\\":[],\\"payable\\":false,\\"stateMutability\\":\\"nonpayable\\",\\"type\\":\\"function\\\"}]')
808060200186815260200185815260200184815260200180602001838103835288818151815260200191508051906020019060200280838360005b838110156
100ec5780820151818401526020810190506100d1565b50505050905001838103825284818151815260200191508051906020019060200280838360005b8381\\
103835288818151815260200191508051906020019060200280838360005b8381101561028c578082015181840152602081019050610271565b5050509050
8152602001838360200280828437600081840152601f19601f8201169050808301925050505050505091929192905050610b10565b005b610473600480360
3602081101561045d57600080fd5b81019080803590602001909291905050610d27565b6040518082815260200191505060405180910390f35b6104bf6004
803603604081101561049f57600080fd5b810190808035906020019092919080359060200190929190505050610d45565b005b60606000866000606060606003604
60 ff 16 15 6 10 5 b c 5 76 00 186 60 01815 181 106 105 a 357 fe 5 b 60 200 260 200 1019 060 030 b 908 160 030 b 8152 505 06 105 e 055 5 b 600 086 600 1815 181106 105 c b 57 fe 5 b 600 105 c b 57 
01019060030b908160030b81525050610643565b60008660028151811061062e57 \\ fe5b602002602001019060030b908160030b815250505b806002015494508
2001820160405280156107aa5781602001602082028038833980820191505090505b50945060008091505b838210156109cf57600260008b815260200190815
260200160002060010160008381526020019081526020016000205490506000600160008381526020019081526020016000209506000600584029050828b82\\
908160030b8152505081600001600890549061010000a900460ff16156108d05760018b60038301815181106108b757fe5b602002602001019060030b9081600\\
```

```
857 fe5 b602 0026 02001 01906 0030 b90816 0030 b81525 0505 b816 00401546 00186 0101945 0816 00201548 a85815181106 1097 a57 fe5 b602 02026 0200101818152
5050816001015489858151811061099757 \\ fe5b60200260200101818152505081600301548885815181106109b457 \\ fe5b6020026020010181815250505050818
06001019250506107b3565b826040519080825280602002602001820160405280156109\\ fe5781602001602082028038833980820191505090505b5094506000
809050600092505b84831015610adc57600260008c8152602001908152602001600020600101600084815260200190815260200160002054915060006001600
08481526020019081526020016000209050806004015460001b878380600101945081518110610a6e57fe5b60200260200101818152505060008090505b8160
fffff021916908360030b63fffffff16021790555085600160008b815260200190815260200160002060000160086101000a81548160ff0219169083151502
5b5080600160008c815260200190815260200160002060040181905550505b50505050505050505050528286000526040600020600091509050806
31582092454 f 0a2189 d 48 c 9559559 a 50326 b 36 c 0 d 27 c ad 3 f 0 b c d 72 ad 99 a 4150833625964736 f 6 c 63430005100032"; \\
StoreMapContract = web3.eth.contract(abi);
web3.eth.estimateGas({data: bytecode})
StoreMap = StoreMapContract.new({
  from: web3.eth.accounts[0],
  data: bytecode,
  gas: '1400000'
  position: "w351111111111",
  txtime:278000
 },function (e, contract){
  console.log(e, contract);
  if(!e){
    if(!contract.address) {
      console.log("Contract transaction send: TransactionHash: " + contract.transactionHash + " waiting to be mined...");
      console.log("Contract mined! Address: " + contract.address);
      console.log(contract);
 }
});
```

然后miner.start(),挖着挖着就会返回合约的adress,保存好,这是合约调用时的所需的信息。