前期热身报告

16340221 王睿泽

实验环境

Oracle VM VirtualBox + Ubuntu 18.04

以太坊的安装

进入 ubuntu 终端命令行：

分别输入：

sudo apt-get installsoftware-properties-common

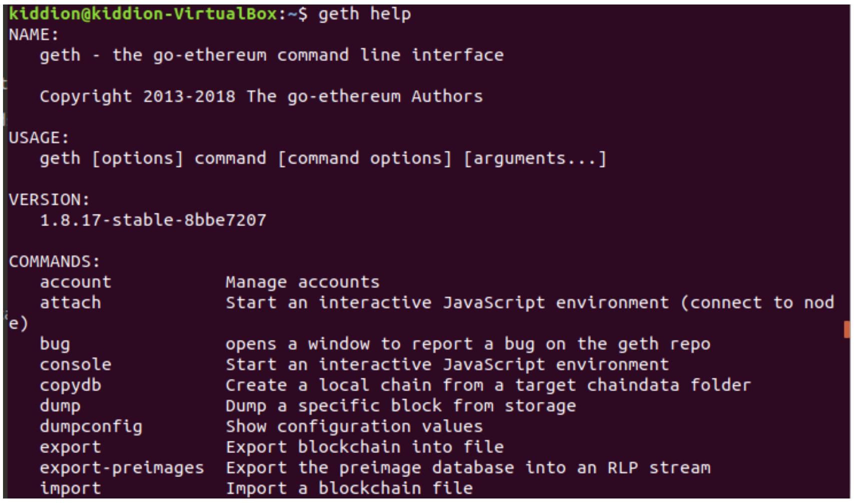
sudo add-apt-repository -yppa:ethereum/ethereum

sudo add-apt-repository -yppa:ethereum/ethereum-dev

sudo apt-get update

sudo apt-get install ethereum

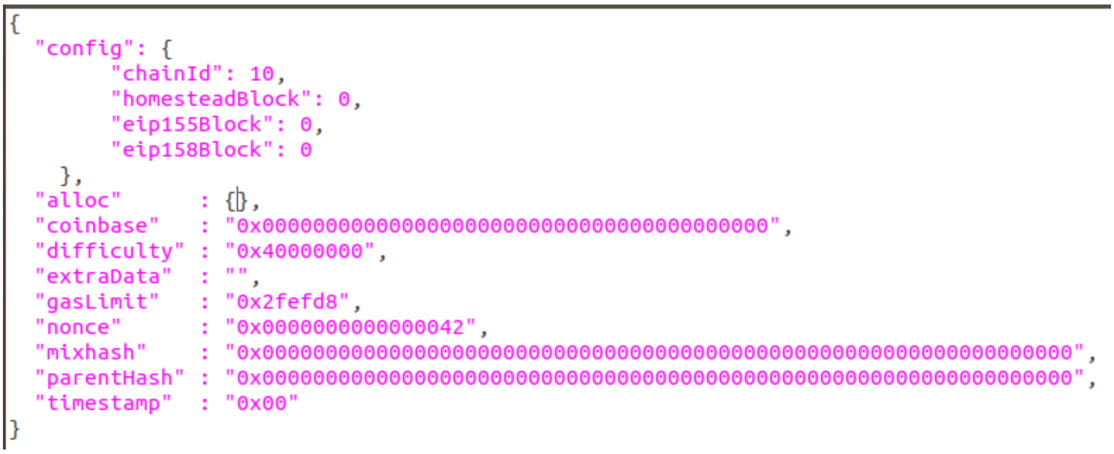
安装完成后输入 geth help 弹出如下图信息即安装成功



```
kiddion@kiddion-VirtualBox:~$ geth help
NAME:
   geth - the go-ethereum command line interface

   Copyright 2013-2018 The go-ethereum Authors

USAGE:
   geth [options] command [command options] [arguments...]

VERSION:
   1.8.17-stable-8bbe7207

COMMANDS:
   account          Manage accounts
   attach           Start an interactive JavaScript environment (connect to nod
e)
   bug              opens a window to report a bug on the geth repo
   console          Start an interactive JavaScript environment
   copydb           Create a local chain from a target chaindata folder
   dump             Dump a specific block from storage
   dumpconfig       Show configuration values
   export           Export blockchain into file
   export-preimages Export the preimage database into an RLP stream
   import           Import a blockchain file
```

私有链创世区块搭建

1. 创建创世文件（genesis.json）



```
{
  "config": {
        "chainId": 10,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
  "alloc"      : {},
  "coinbase"   : "0x0000000000000000000000000000000000000000",
  "difficulty" : "0x40000000",
  "extraData"  : "",
  "gasLimit"   : "0x2fefd8",
  "nonce"      : "0x0000000000000042",
  "mixhash"    : "0x0000000000000000000000000000000000000000000000000000000000000000",
  "parentHash" : "0x0000000000000000000000000000000000000000000000000000000000000000",
  "timestamp"  : "0x00"
}
```

参数解释：

mixhash:与 nonce 配合用于挖矿，由上一个区块的一部分生成的 hash。注意他和 nonce 的设置需要满足以太坊的 Yellow paper, 4.3.4. Block Header Validity, (44)章节所描述的条件。

nonce:nonce 就是一个 64 位随机数，用于挖矿，注意他和 mixhash 的设置需要满足以太坊的 Yellow paper, 4.3.4. Block Header Validity, (44)章节所描述的条件。

difficulty:设置当前区块的难度，如果难度过大，cpu 挖矿就很难，这里设置较小难度

alloc 用来预置账号以及账号的以太币数量，因为私有链挖矿比较容易，所以我们不需要预置有币的账号，需要的时候自己创建即可以。

coinbase :矿工的账号，随便填

timestamp:设置创世块的时间戳

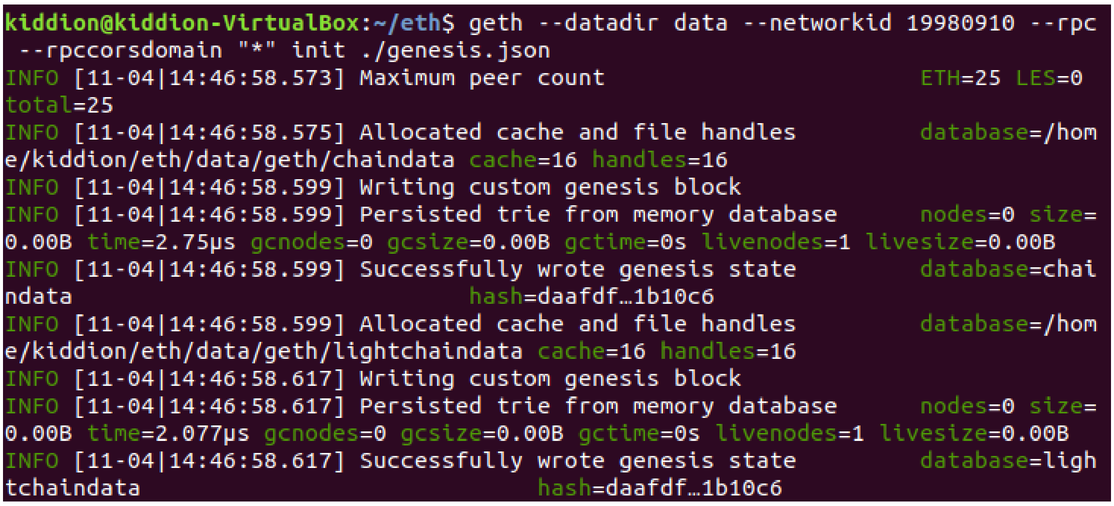
parentHash:上一个区块的 hash 值，因为是创世块，所以这个值是 0

extraData:附加信息，随便填，可以填你的个性信息

gasLimit:该值设置对 GAS 的消耗总量限制，用来限制区块能包含的交易信息总和，因为我们是私有链，所以填最大。

2. **创建数据存放地址并初始化创世块**

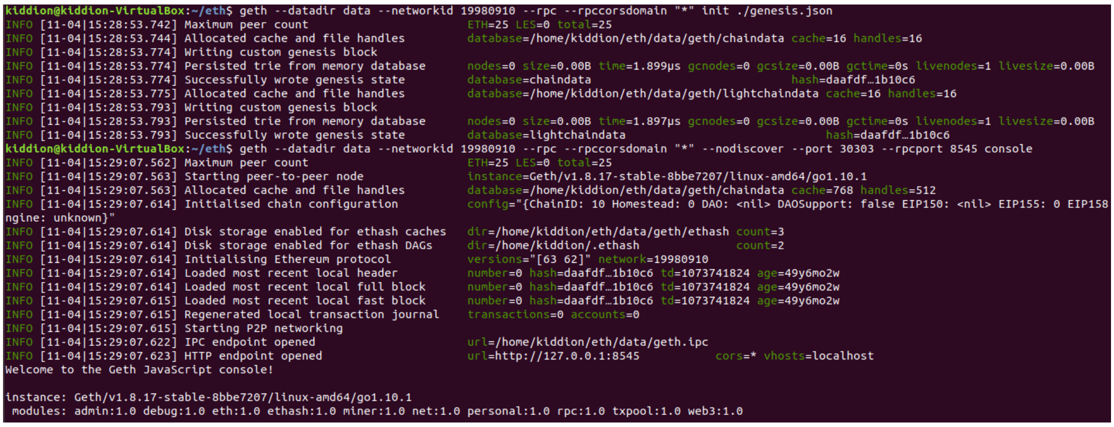
geth --datadir data --networkid 19980910 --rpc --rpccorsdomain "*" init ./genesis.json



```
kiddion@kiddion-VirtualBox:~/eth$ geth --datadir data --networkid 19980910 --rpc
 --rpccorsdomain "*" init ./genesis.json
INFO [11-04|14:46:58.573] Maximum peer count                       ETH=25 LES=0
total=25
INFO [11-04|14:46:58.575] Allocated cache and file handles         database=/hom
e/kiddion/eth/data/geth/chaindata cache=16 handles=16
INFO [11-04|14:46:58.599] Writing custom genesis block
INFO [11-04|14:46:58.599] Persisted trie from memory database      nodes=0 size=
0.00B time=2.75µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-04|14:46:58.599] Successfully wrote genesis state         database=chai
ndata                            hash=daafdf…1b10c6
INFO [11-04|14:46:58.599] Allocated cache and file handles         database=/hom
e/kiddion/eth/data/geth/lightchaindata cache=16 handles=16
INFO [11-04|14:46:58.617] Writing custom genesis block
INFO [11-04|14:46:58.617] Persisted trie from memory database      nodes=0 size=
0.00B time=2.077µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-04|14:46:58.617] Successfully wrote genesis state         database=ligh
tchaindata                            hash=daafdf…1b10c6
```

3. **开启 geth 私链客户端**

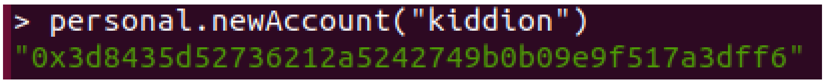
geth --datadir data --networkid 19980910 --rpc --rpccorsdomain "*" --nodiscover --port 30303 --rpcport 8545 console



```
kiddion@kiddion-VirtualBox:~/eth$ geth --datadir data --networkid 19980910 --rpc --rpccorsdomain "*" init ./genesis.json
INFO [11-04|15:28:53.742] Maximum peer count                       ETH=25 LES=0 total=25
INFO [11-04|15:28:53.744] Allocated cache and file handles         database=/home/kiddion/eth/data/geth/chaindata cache=16 handles=16
INFO [11-04|15:28:53.774] Writing custom genesis block
INFO [11-04|15:28:53.774] Persisted trie from memory database      nodes=0 size=0.00B time=1.899µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-04|15:28:53.774] Successfully wrote genesis state         database=chaindata                            hash=daafdf…1b10c6
INFO [11-04|15:28:53.775] Allocated cache and file handles         database=/home/kiddion/eth/data/geth/lightchaindata cache=16 handles=16
INFO [11-04|15:28:53.793] Writing custom genesis block
INFO [11-04|15:28:53.793] Persisted trie from memory database      nodes=0 size=0.00B time=1.897µs gcnodes=0 gcsize=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [11-04|15:28:53.793] Successfully wrote genesis state         database=lightchaindata                            hash=daafdf…1b10c6
kiddion@kiddion-VirtualBox:~/eth$ geth --datadir data --networkid 19980910 --rpc --rpccorsdomain "*" --nodiscover --port 30303 --rpcport 8545 console
INFO [11-04|15:29:07.562] Maximum peer count                       ETH=25 LES=0 total=25
INFO [11-04|15:29:07.563] Starting peer-to-peer node               instance=Geth/v1.8.17-stable-8bbe7207/linux-amd64/go1.10.1
INFO [11-04|15:29:07.563] Allocated cache and file handles         database=/home/kiddion/eth/data/geth/chaindata cache=768 handles=512
INFO [11-04|15:29:07.614] Initialised chain configuration          config="{ChainID: 10 Homestead: 0 DAO: <nil> DAOSupport: false EIP150: <nil> EIP155: 0 EIP158
ngine: unknown}"
INFO [11-04|15:29:07.614] Disk storage enabled for ethash caches   dir=/home/kiddion/eth/data/geth/ethash count=3
INFO [11-04|15:29:07.614] Disk storage enabled for ethash DAGs     dir=/home/kiddion/.ethash              count=2
INFO [11-04|15:29:07.614] Initialising Ethereum protocol           versions="[63 62]" network=19980910
INFO [11-04|15:29:07.614] Loaded most recent local header          number=0 hash=daafdf…1b10c6 td=1073741824 age=49y6mo2w
INFO [11-04|15:29:07.614] Loaded most recent local full block      number=0 hash=daafdf…1b10c6 td=1073741824 age=49y6mo2w
INFO [11-04|15:29:07.615] Loaded most recent local fast block      number=0 hash=daafdf…1b10c6 td=1073741824 age=49y6mo2w
INFO [11-04|15:29:07.615] Regenerated local transaction journal    transactions=0 accounts=0
INFO [11-04|15:29:07.615] Starting P2P networking
INFO [11-04|15:29:07.622] IPC endpoint opened                      url=/home/kiddion/eth/data/geth.ipc
INFO [11-04|15:29:07.623] HTTP endpoint opened                     url=http://127.0.0.1:8545           cors=* vhosts=localhost
Welcome to the Geth JavaScript console!

instance: Geth/v1.8.17-stable-8bbe7207/linux-amd64/go1.10.1
 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
```

4. **创建账号**

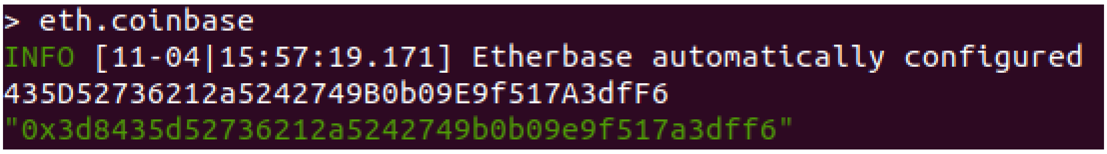
personal.newAccount("kiddion")



```
> personal.newAccount("kiddion")
"0x3d8435d52736212a5242749b0b09e9f517a3dff6"
```

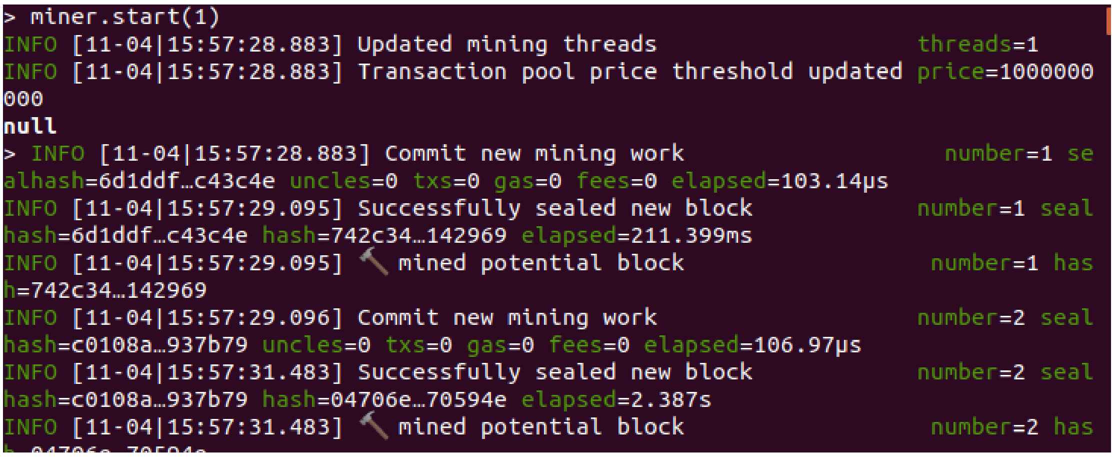
5. **挖矿**

矿工账号



```
> eth.coinbase
INFO [11-04|15:57:19.171] Etherbase automatically configured
435D52736212a5242749B0b09E9f517A3dfF6
"0x3d8435d52736212a5242749b0b09e9f517a3dff6"
```

开始挖矿



```
> miner.start(1)
INFO [11-04|15:57:28.883] Updated mining threads                   threads=1
INFO [11-04|15:57:28.883] Transaction pool price threshold updated price=1000000
000
null
> INFO [11-04|15:57:28.883] Commit new mining work                   number=1 se
alhash=6d1ddf…c43c4e uncles=0 txs=0 gas=0 fees=0 elapsed=103.14µs
INFO [11-04|15:57:29.095] Successfully sealed new block            number=1 seal
hash=6d1ddf…c43c4e hash=742c34…142969 elapsed=211.399ms
INFO [11-04|15:57:29.095] 🔨 mined potential block                   number=1 has
h=742c34…142969
INFO [11-04|15:57:29.096] Commit new mining work                   number=2 seal
hash=c0108a…937b79 uncles=0 txs=0 gas=0 fees=0 elapsed=106.97µs
INFO [11-04|15:57:31.483] Successfully sealed new block            number=2 seal
hash=c0108a…937b79 hash=04706e…70594e elapsed=2.387s
INFO [11-04|15:57:31.483] 🔨 mined potential block                   number=2 has
```

私有链节点的加入

1. **按照创世区块搭建的 2、3 步搭建节点 1**

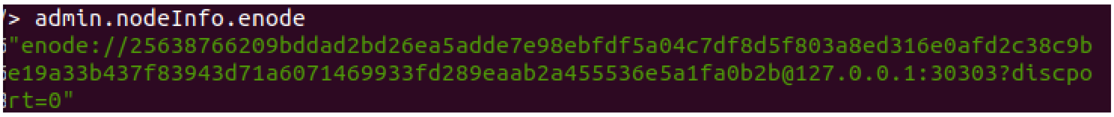
   geth --datadir data1 --networkid 19980910 --rpc --rpccorsdomain "*" init ./genesis.json

   在打开 geth 客户端时要更改端口号

   geth --datadir data1 --networkid 19980910 --rpc --rpccorsdomain "*" --nodiscover --port 30304 --rpcport 8546 console

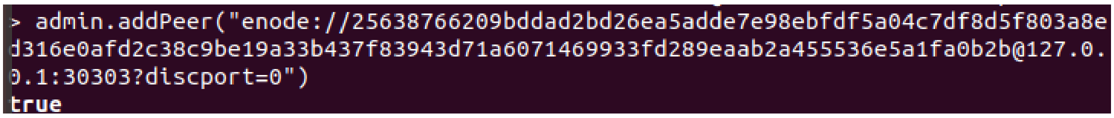
2. **在初始节点（节点 0）中查看 enode**

   admin.nodeInfo.enode



```
> admin.nodeInfo.enode
"enode://25638766209bddad2bd26ea5adde7e98ebfdf5a04c7df8d5f803a8ed316e0afd2c38c9b
e19a33b437f83943d71a6071469933fd289eaab2a455536e5a1fa0b2b@127.0.0.1:30303?discpo
rt=0"
```

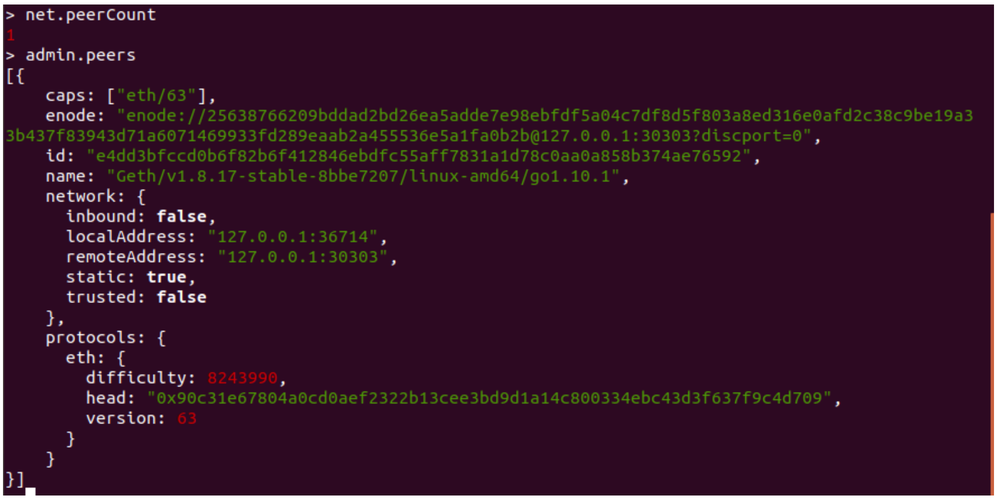
3. **在节点 1 的控制台，加入节点 0**



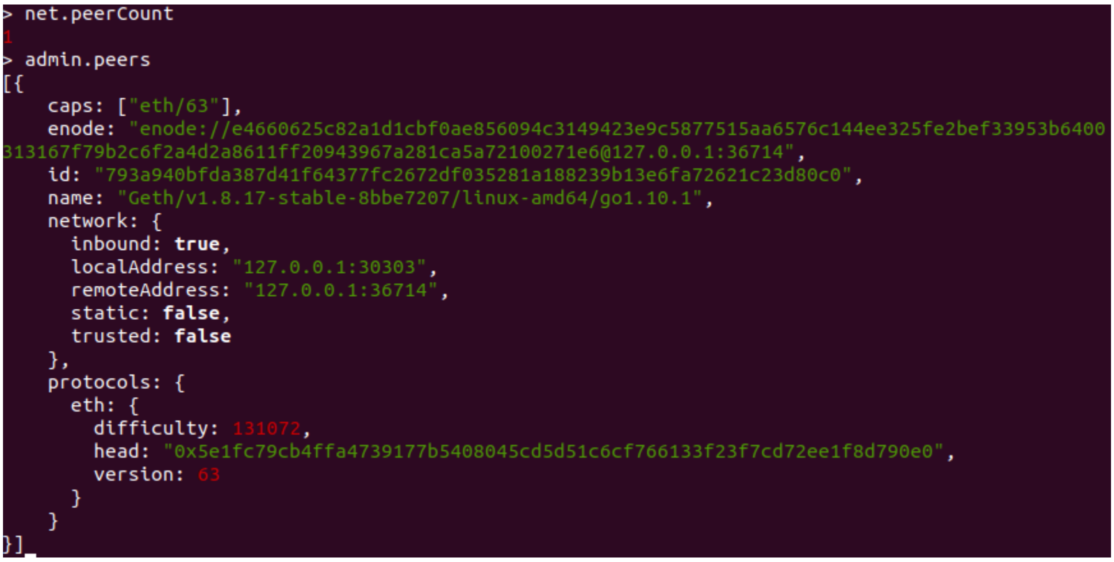
```
> admin.addPeer("enode://25638766209bddad2bd26ea5adde7e98ebfdf5a04c7df8d5f803a8e
d316e0afd2c38c9be19a33b437f83943d71a6071469933fd289eaab2a455536e5a1fa0b2b@127.0.
0.1:30303?discport=0")
true
```

4. **在节点 0 和节点 1 中查看连接节点的数量和列表**

   由下图可知，可以在两个节点中分别看到对方的接入



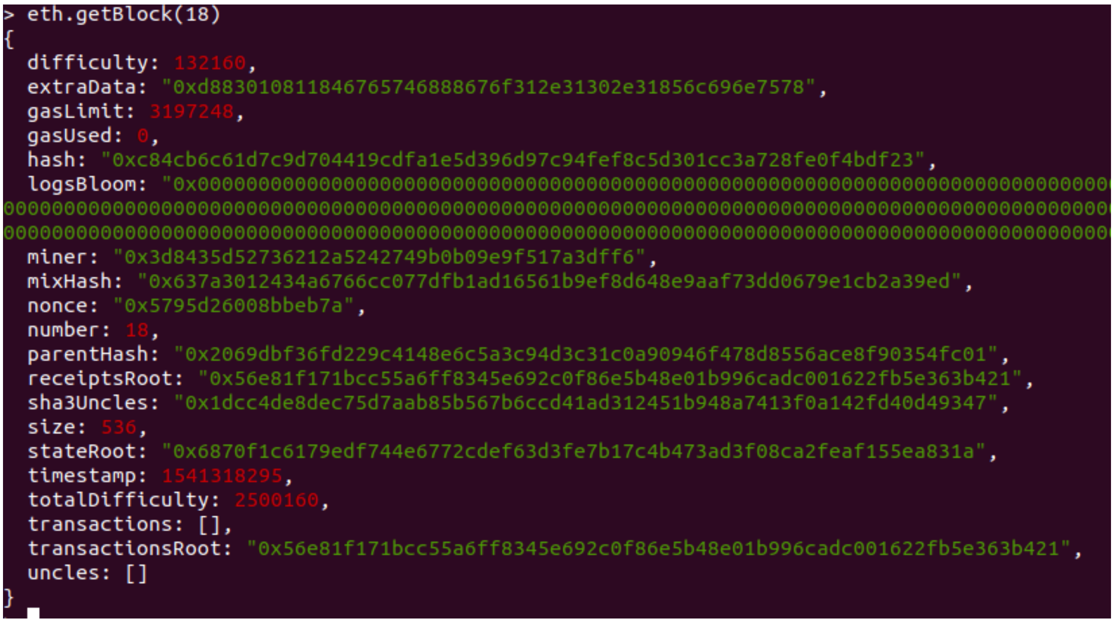
```
> net.peerCount
1
> admin.peers
[{
    caps: ["eth/63"],
    enode: "enode://25638766209bddad2bd26ea5adde7e98ebfdf5a04c7df8d5f803a8ed316e0afd2c38c9be19a3
3b437f83943d71a6071469933fd289eaab2a455536e5a1fa0b2b@127.0.0.1:30303?discport=0",
    id: "e4dd3bfccd0b6f82b6f412846ebdfc55aff7831a1d78c0aa0a858b374ae76592",
    name: "Geth/v1.8.17-stable-8bbe7207/linux-amd64/go1.10.1",
    network: {
      inbound: false,
      localAddress: "127.0.0.1:36714",
      remoteAddress: "127.0.0.1:30303",
      static: true,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 8243990,
        head: "0x90c31e67804a0cd0aef2322b13cee3bd9d1a14c800334ebc43d3f637f9c4d709",
        version: 63
      }
    }
}]
```



```
> net.peerCount
1
> admin.peers
[{
    caps: ["eth/63"],
    enode: "enode://e4660625c82a1d1cbf0ae856094c3149423e9c5877515aa6576c144ee325fe2bef33953b6400
313167f79b2c6f2a4d2a8611ff20943967a281ca5a72100271e6@127.0.0.1:36714",
    id: "793a940bfda387d41f64377fc2672df035281a188239b13e6fa72621c23d80c0",
    name: "Geth/v1.8.17-stable-8bbe7207/linux-amd64/go1.10.1",
    network: {
      inbound: true,
      localAddress: "127.0.0.1:30303",
      remoteAddress: "127.0.0.1:36714",
      static: false,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 131072,
        head: "0x5e1fc79cb4ffa4739177b5408045cd5d51c6cf766133f23f7cd72ee1f8d790e0",
        version: 63
      }
    }
}]
```

对 getBlock 中所得区块的各个字段进行解释

进行 getBlock 操作：eth.getBlock(18)



```
> eth.getBlock(18)
{
  difficulty: 132160,
  extraData: "0xd883010811846765746888676f312e31302e31856c696e7578",
  gasLimit: 3197248,
  gasUsed: 0,
  hash: "0xc84cb6c61d7c9d704419cdfa1e5d396d97c94fef8c5d301cc3a728fe0f4bdf23",
  logsBloom: "0x000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
  miner: "0x3d8435d52736212a5242749b0b09e9f517a3dff6",
  mixHash: "0x637a3012434a6766cc077dfb1ad16561b9ef8d648e9aaf73dd0679e1cb2a39ed",
  nonce: "0x5795d26008bbeb7a",
  number: 18,
  parentHash: "0x2069dbf36fd229c4148e6c5a3c94d3c31c0a90946f478d8556ace8f90354fc01",
  receiptsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 536,
  stateRoot: "0x6870f1c6179edf744e6772cdef63d3fe7b17c4b473ad3f08ca2feaf155ea831a",
  timestamp: 1541318295,
  totalDifficulty: 2500160,
  transactions: [],
  transactionsRoot: "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421",
  uncles: []
}
```

difficulty - BigNumber 类型。当前块的难度，整数

extraData - 字符串。当前块的 extra data 字段

gasLimit - Number，当前区块允许使用的最大 gas

gasUsed - 当前区块累计使用的总的 gas

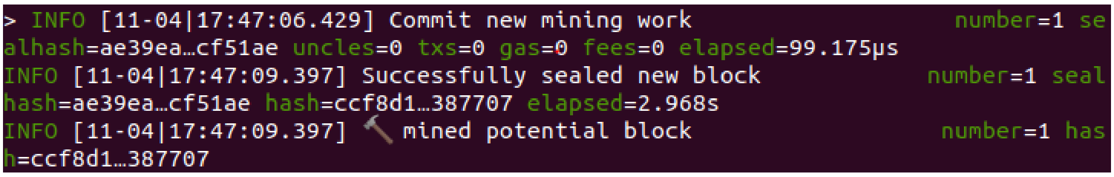
hash - 字符串，区块的哈希串。当这个区块处于 pending 将会返回 null

logsBloom - 字符串，区块日志的布隆过滤器 9。当这个区块处于 pending 将会返回 null

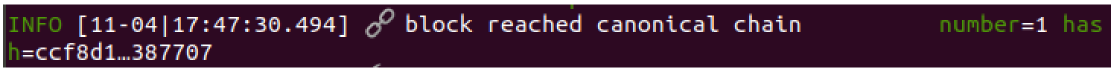
miner - 字符串，20 字节。这个区块获得奖励的矿工
mixhash - 与 nonce 配合用于挖矿，由上一个区块的一部分生成的 hash。
nonce - 字符串，8 字节。POW 生成的哈希。当这个区块处于 pending 将会返回 null
number - 区块号。当这个区块处于 pending 将会返回 null
parentHash - 字符串，32 字节的父区块的哈希值
receiptsRoot - 收据树的根哈希值
sha3Uncles - 字符串，32 字节。叔区块的哈希值。
size - Number。当前这个块的字节大小
stateRoot - 字符串，32 字节。区块的最终状态前缀树的根
timestamp - Number。区块打包时的 unix 时间戳
totalDifficulty - BigNumber 类型。区块链到当前块的总难度，整数
transactions - 数组。交易对象。或者是 32 字节的交易哈希
transactionsRoot - 字符串，32 字节，区块的交易前缀树的根。
uncles - 数组。叔哈希的数组。

对日志输出进行解释

挖矿



```
> INFO [11-04|17:47:06.429] Commit new mining work                    number=1 se
alhash=ae39ea…cf51ae uncles=0 txs=0 gas=0 fees=0 elapsed=99.175µs
INFO [11-04|17:47:09.397] Successfully sealed new block             number=1 seal
hash=ae39ea…cf51ae hash=ccf8d1…387707 elapsed=2.968s
INFO [11-04|17:47:09.397] 🔨 mined potential block                  number=1 has
h=ccf8d1…387707
```

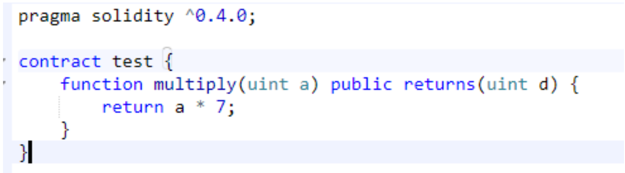


```
INFO [11-04|17:47:30.494] 🔗 block reached canonical chain          number=1 has
h=ccf8d1…387707
```

Commit new mining work 表明发出申请挖掘下一个块
Successfully sealed new block 密封成功
mined potential block 挖掘潜在的块
block reached canonical chain 块到达标准链

编写简单的智能合约，在 remix 下进行调试，并部署在链上进行调用

1. **编写合约**
   在这里就按照网上的方法编写了一个比较简单的合约

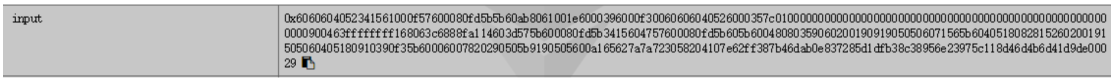


```
pragma solidity ^0.4.0;

contract test {
    function multiply(uint a) public returns(uint d) {
        return a * 7;
    }
}
```

2. **进行编译，获取 abi 和 bytecode**

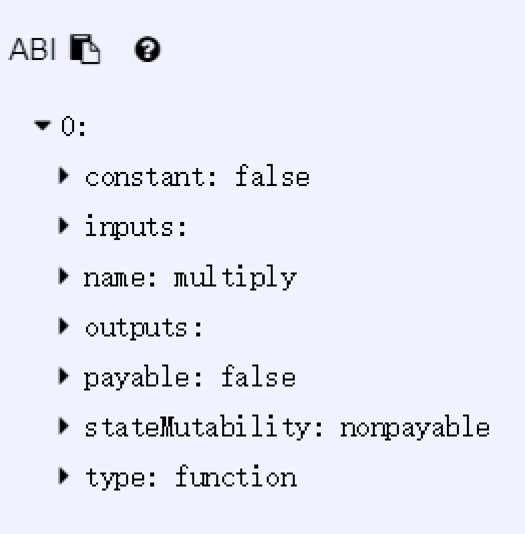
   在 remix 上进行测试编译

   bytecode:



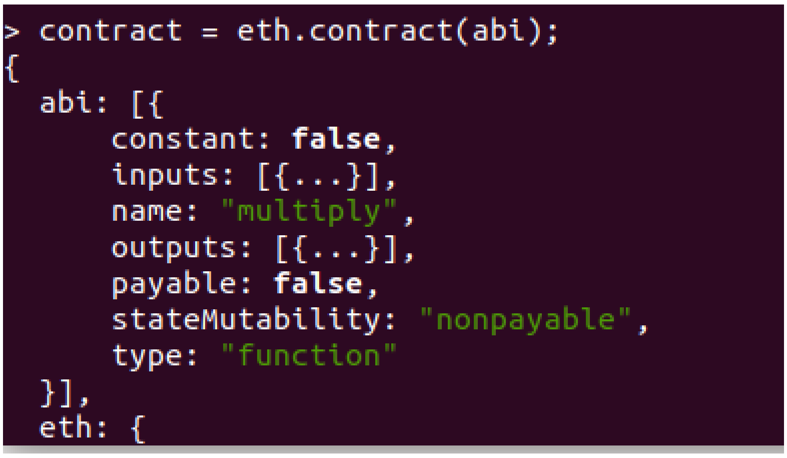
```
input    0x6060604052341561000f57600080fd5b5b60ab8061001e6000396000f30060606040526000357c010000000000000000000000000000000000000000000000000000
         0000900463ffffffff168063c6888fa114603d575b600080fd5b3415604757600080fd5b605b6004808035906020019091905050606071565b6040518082815260200191
         505060405180910390f35b60006007820290505b9190505600a165627a7a723058204107e62ff387b46dab0e837285d1dfb38c38956e23975c118d46d4b6d41d9de000
         29
```

   abi:

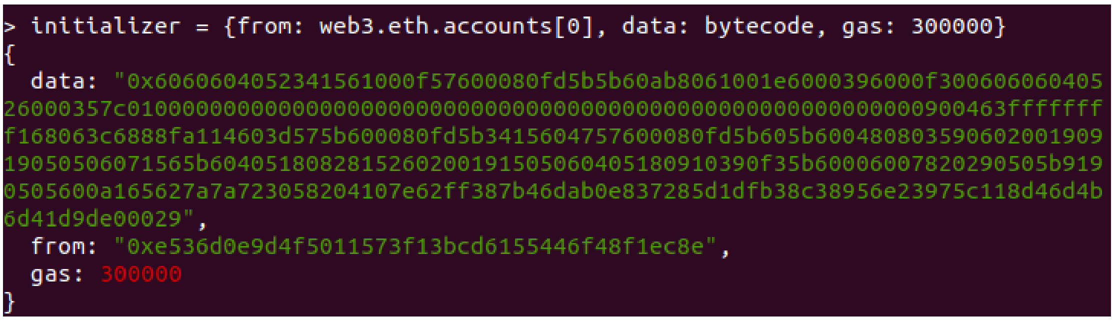


```
ABI
▼ 0:
  ▶ constant: false
  ▶ inputs:
  ▶ name: multiply
  ▶ outputs:
  ▶ payable: false
  ▶ stateMutability: nonpayable
  ▶ type: function
```

3. **实例化合约并进行布署**

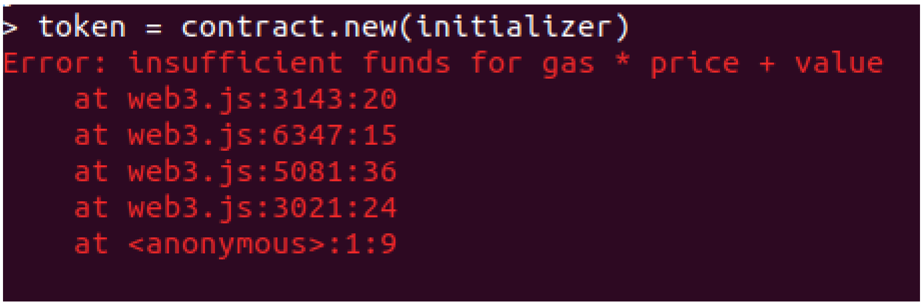


```
> contract = eth.contract(abi);
{
  abi: [{
      constant: false,
      inputs: [{...}],
      name: "multiply",
      outputs: [{...}],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
  }],
  eth: {
```

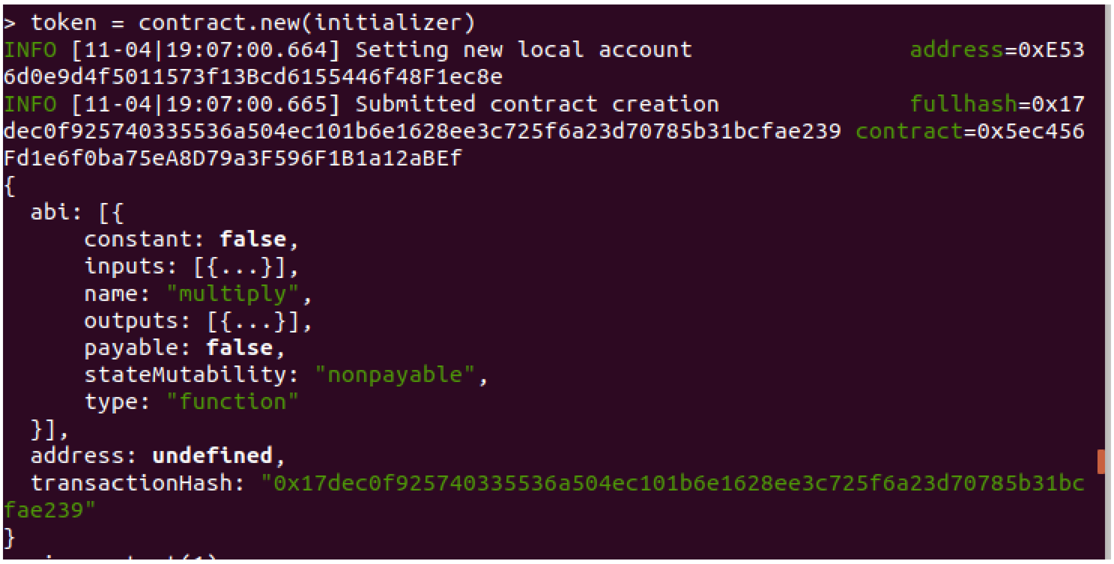


```
> initializer = {from: web3.eth.accounts[0], data: bytecode, gas: 300000}
{
  data: "0x6060604052341561000f57600080fd5b5b60ab8061001e6000396000f300606060405
26000357c0100000000000000000000000000000000000000000000000000000000900463ffffff
ff168063c6888fa114603d575b600080fd5b3415604757600080fd5b605b600480803590602001909
19050506071565b6040518082815260200191505060405180910390f35b60006007820290505b919
0505600a165627a7a723058204107e62ff387b46dab0e837285d1dfb38c38956e23975c118d46d4b
6d41d9de00029",
  from: "0xe536d0e9d4f5011573f13bcd6155446f48f1ec8e",
  gas: 300000
}
```

   如果余额不足会产生下面的报错，可以先进行挖矿

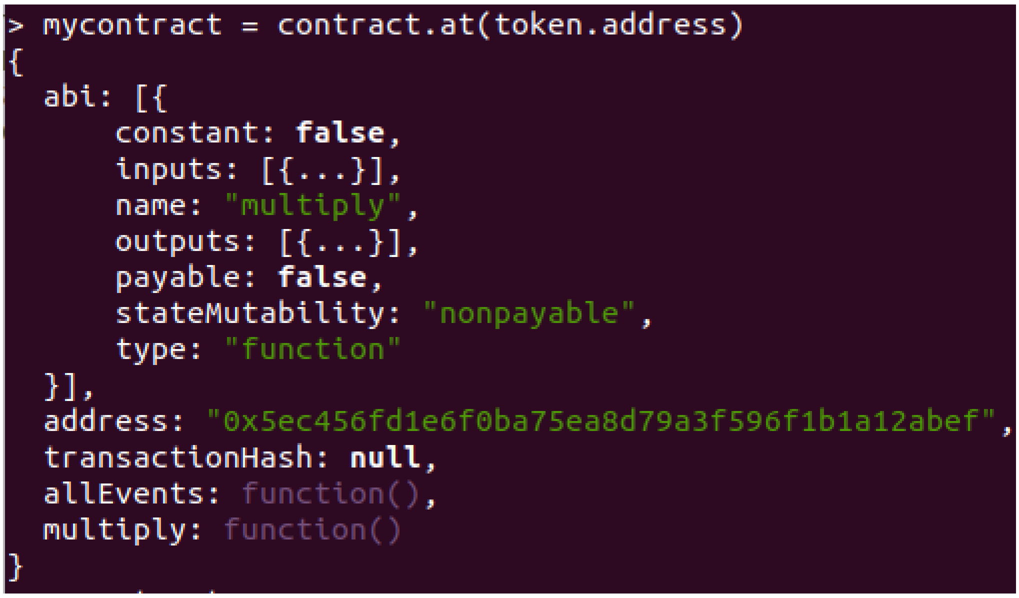


```
> token = contract.new(initializer)
Error: insufficient funds for gas * price + value
    at web3.js:3143:20
    at web3.js:6347:15
    at web3.js:5081:36
    at web3.js:3021:24
    at <anonymous>:1:9
```

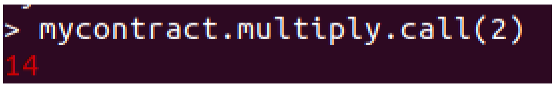


```
> token = contract.new(initializer)
INFO [11-04|19:07:00.664] Setting new local account                address=0xE53
6d0e9d4f5011573f13Bcd6155446f48F1ec8e
INFO [11-04|19:07:00.665] Submitted contract creation              fullhash=0x17
dec0f925740335536a504ec101b6e1628ee3c725f6a23d70785b31bcfae239 contract=0x5ec456
Fd1e6f0ba75eA8D79a3F596F1B1a12aBEf
{
  abi: [{
      constant: false,
      inputs: [{...}],
      name: "multiply",
      outputs: [{...}],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
  }],
  address: undefined,
  transactionHash: "0x17dec0f925740335536a504ec101b6e1628ee3c725f6a23d70785b31bc
fae239"
}
```

4. 通过合约地址，实例化自己的合约，并进行调用



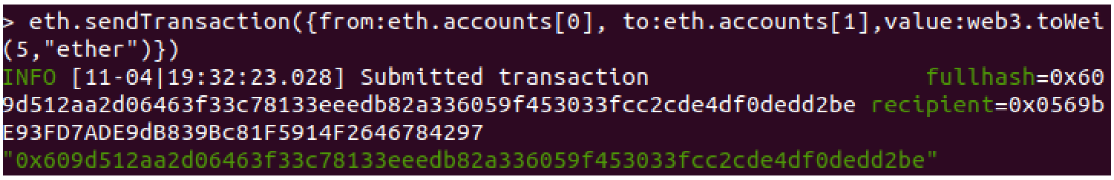
```
> mycontract = contract.at(token.address)
{
  abi: [{
      constant: false,
      inputs: [{...}],
      name: "multiply",
      outputs: [{...}],
      payable: false,
      stateMutability: "nonpayable",
      type: "function"
  }],
  address: "0x5ec456fd1e6f0ba75ea8d79a3f596f1b1a12abef",
  transactionHash: null,
  allEvents: function(),
  multiply: function()
}
```



```
> mycontract.multiply.call(2)
14
```

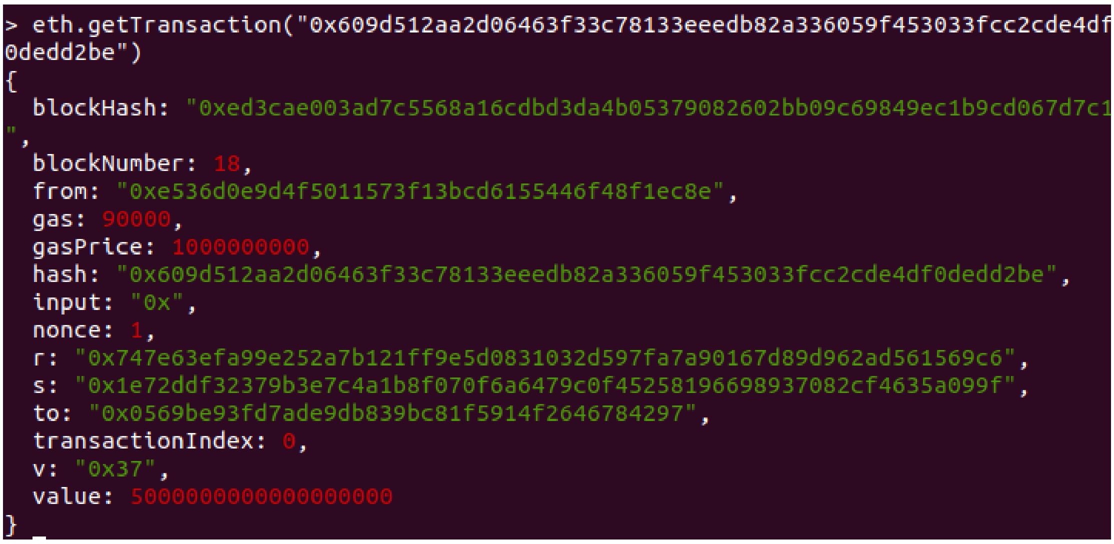
对交易的字段进行解释

首先发起一笔转账



```
> eth.sendTransaction({from:eth.accounts[0], to:eth.accounts[1],value:web3.toWei
(5,"ether")})
INFO [11-04|19:32:23.028] Submitted transaction                    fullhash=0x60
9d512aa2d06463f33c78133eeedb82a336059f453033fcc2cde4df0dedd2be recipient=0x0569b
E93FD7ADE9dB839Bc81F5914F2646784297
"0x609d512aa2d06463f33c78133eeedb82a336059f453033fcc2cde4df0dedd2be"
```

查询交易



```
> eth.getTransaction("0x609d512aa2d06463f33c78133eeedb82a336059f453033fcc2cde4df
0dedd2be")
{
  blockHash: "0xed3cae003ad7c5568a16cdbd3da4b05379082602bb09c69849ec1b9cd067d7c1
",
  blockNumber: 18,
  from: "0xe536d0e9d4f5011573f13bcd6155446f48f1ec8e",
  gas: 90000,
  gasPrice: 1000000000,
  hash: "0x609d512aa2d06463f33c78133eeedb82a336059f453033fcc2cde4df0dedd2be",
  input: "0x",
  nonce: 1,
  r: "0x747e63efa99e252a7b121ff9e5d0831032d597fa7a90167d89d962ad561569c6",
  s: "0x1e72ddf32379b3e7c4a1b8f070f6a6479c0f452581966989370 82cf4635a099f",
  to: "0x0569be93fd7ade9db839bc81f5914f2646784297",
  transactionIndex: 0,
  v: "0x37",
  value: 5000000000000000000
}
```

blockHash:交易区块的哈希

blockNumber:交易区块的块号

from:交易发起者的地址

gas:交易发起者提供的 gas

gasPrice: 交易发起者配置的 gas 价格

hash:交易的哈希值

input:交易附带的数据

nonce: 交易的发起者在之前进行过的交易数量

r:交易签名的数据

s:交易签名的数据

to:交易接受者的地址

transactionIndexs:

v: 交易签名的数据

value:交易的价值