

以太坊Web3-API命令解析

1.eth命令集

包含一些跟操作区块链相关的方法.

命令名称	命令说明	备注
defaultAccount	查询默认账户	
defaultBlock	查询默认区块	
syncing/getSyncing	同步节点信息	
isSyncing(callback)	设置回调函数,在同步启动/更新/停止	
coinbase/getCoinbase	矿工账户,挖矿后得到的虚拟币进入该账户	
hashrate/getHashrate	挖矿每秒Hash数,算力值	
gasPrice/getGasPrice	查询Gas的价格	
accounts/getAccounts	查询所有账户	
mining/getMining	是否是在挖矿	
blockNumber/getBlockNumber	返回当前块数	
register(hexString)		尚未实现
unRegister(hexString)		尚未实现
getBalance(address)	查询账户余额	
getStorageAt(address, position)	从制定位置获取存储内容	
getCode(address)	从指定地址获取十六进制编码	
getBlock(hash/number)	获取块信息	
getBlockTransactionCount(hash/number)	获取相关块交易信息	
getUncle(hash/number)	获取侧链信息	
getBlockUncleCount(hash/number)	获取侧链块数	
getTransaction(hash)	查询交易信息	
getTransactionFromBlock(hashOrNumber, indexNumber)	按块查询交易信息	
getTransactionReceipt(hash)	根据事物Hash返回事物接收信息	
getTransactionCount(address)	根据地址查询事物数	
sendTransaction(object)	发送事物	

sendRawTransaction(object)	发送已经签名事物	
sign(object)	签名	
call(object)	执行消息事物,但是不会被记录到区块链中	
estimateGas(object)	评估事物需要的Gas量	
[filter(array (, options))] watch(callback)stopWatching(callback)get()	过滤器	
Contract(abiArray)	创建合约	
contract.myMethod()	合约方法	
contract.myEvent()	创建事件过滤器	
contract.allEvents()	所有的事件	
getCompilers()	查询可用编译器	
compile.lll(string)	编译LLL源代码	
compile.solidity(string)	编译solidity源代码	
compile.serpent(string)	编译serpent源代码	
namereg	返回全局定义对象	
sendIBANTransaction	发送IBANT事物	
[iban]fromAddressfromBanc createIndirectisValidDirectisIndirectchecksuminstitutioncontinentaddresstoString		

#示例代码略:

详解:<https://github.com/ethereum/wiki/wiki/JavaScript-API#web3ethdefaultaccount>

2.admin命令集

包含一些与管理节点相关的方法

命令名称	命令说明	备注
addPeer(url)	增加节点	url为节点信息,需要用双引号
datadir	查询数据文件存储目录	默认目录/home/*/.ethereum
nodeInfo	查询节点信息	
peers	获取远程连接的节点信息	包括网络地址/协议/其他基本信息
setSolc(path)	设置节点使用的Solidity编译器路径	默认为/ usr / bin / solc
startRPC(host, port, cors, apis)	启动一个RPCAPIWeb的HTTP服务用于接收客户端请求	
startWS(host, port, cors, apis)	启动一个RPCAPIWeb的WebSocket服务用于接收客户端请求	
stopRPC()	停止HTTP的RPC服务	
stopWS()	停止WebSocket的RPC服务	

```

#addPeer增加节点
>admin.addPeer("enode://c68470ae95eb374f2d026daf3a81e6c9bbe9d4b66d798936f0dd094a9f3269205111d59cacfd7c4e5ed4111ad0cea6464e224f4814ed19c785d366cbb03c1c10@127.0.0.1:30303")

#datadir查询数据存储目录
> admin.datadir
"/home/suzhou/privatechain/data"

#nodeInfo查看节点信息
>admin.nodeInfo
{
  enode:
"enode://44826a5d6a55f88a18298bca4773fca5749cdc3a5c9f308aa7d810e9b31123f3e7c5fba0b1d70aac5308426f47df2a128a6747040a3815cc7dd167d03be320d@[::]:30303",
  id: "",
  ip: "::",
  listenAddr: "[::]:30303",
  name: "Geth/v1.5.0-unstable/linux/go1.6",
  ports: {
    discovery: 30303,
    listener: 30303
  },
  protocols: {
    eth: {
      difficulty: 17334254859343145000,
      genesis: "0xd4e56740f876aef8c010b86a40d5f56745a118d0906a34e69aec8c0db1cb8fa3",
      head: "0xb83f73fbe6220c111136aefd27b160bf4a34085c65ba89f24246b3162257c36a",
      network: 1
    }
  }
}
}

#startRPC启动RPCWeb服务
> admin.startRPC("127.0.0.1", 8545)
true

```

3.debug命令集

调试区块信息节点

命令名称	命令说明	备注
backtraceAt(string)	设置日志回溯地址	
blockProfile(file, seconds)	打开块分析,将分析数据写入磁盘	
cpuProfile(file, seconds)	打开CPU分析,将分析数据写入磁盘	
traceBlockByHash(number, [options])	检索与块编号对应的状态并返回一个帐户列表	
gcStats()	返回GC统计信息	
getBlockRlp(number, [options])	通过编号检索并返回RLP编码块	
goTrace(file, seconds)	打开Go运行时跟踪给定的持续时间，并将跟踪数据写入磁盘	
memStats()	返回内存统计信息	
seedHashsign(number, [options])	通过编号获取块的种子哈希值	
setBlockProfileRate(rate)	设置块分析速率	
setHead(number)	通过块编号设置本地链的当前头	
stacks()	返回运行时栈信息	
startCPUProfile(file)	开始CPU分析	
startGoTrace(file)	开始Go追踪	
stopCPUProfile()	停止CPU分析	
stopGoTrace()	停止Go追踪	
traceBlock(tblockRlp, [options])	返回包含在此块中的所有事务的所有调用操作码的完整堆栈跟踪	
traceBlockByNumber(number, [options])	根据块号追踪块堆栈跟踪信息	
traceBlockByHash(hash, [options])	根据Hash追踪块堆栈跟踪信息	
traceBlockFromFile(fileName, [options])	根据文件追踪块堆栈跟踪信息	
traceTransaction(txHash, [options])	追踪事物	
verbosity(level)	设置日志记录冗长度上限	
vmodule(string)	设置日志记录详细模式	
writeBlockProfile(file)	阻止配置文件写入给定的文件	
writeMemProfile(file)	阻止配置文件写入给定的文件	

#示例代码略

详见:<https://github.com/ethereum/go-ethereum/wiki/Management-APIs#debug>

4.miner命令集

包含启动&停止挖矿的一些方法

命令名称	命令说明	备注
setExtra(string)	设置可以包括的额外数据	限制32字节
setGasPrice(number)	设置采矿交易时接受的最低天然气价格	任何低于此限制的交易都不包括在采矿过程中
start(number)	开启n个线程进行挖矿	
stop()	停止挖矿	
getHashrate	获取hash算力值	
setEtherbase(address)	设置挖矿虚拟币接收地址	默认是第0个账户

5.personal命令集

主要包含一些管理账户的方法

命令名称	命令说明	备注
ecRecover(message, signature)	返回用于计算签名的私钥地址	
importRawKey(keydata, passphrase)	将未加密私钥导入密钥存储区,并用密码对其加密	
listAccounts	返回密钥存储区中所有密钥的所有以太坊帐户地址	
lockAccount(address)	锁住账户,锁住后无法交易	
newAccount()	创建新账户	
unlockAccount(address, passphrase, duration)	解锁账户,需要输入密码	
sendTransaction(tx, passphrase)	创建交易	
sign(message, account, [password])	计算以太坊特殊签名	

#ecRecover返回私钥地址

```
> personal.sign("0xdeadbeaf", "0x9b2055d370f73ec7d8a03e965129118dc8f5bf83", "")
"0xa3f20717a250c2b0b729b7e5becbff67fdae7e0699da4de7ca5895b02a170a12d887fd3b17bfdce3481f10bea41f45ba9f709d39ce8325427b57afcfc994cee1b"
```

#sendTracnsaction创建交易

```
> var tx = {from: "0x391694e7e0b0cce554cb130d723a9d27458f9298", to:
"0xafa3f8684e54059998bc3a7b0d2b0da075154d66", value: web3.toWei(1.23, "ether")}
undefined
> personal.sendTransaction(tx, "passphrase")
```

6.txpool命令集

包含一些查看交易内存池的方法

命令名称	命令说明	备注
content	查询交易池细节,pending待提交事物,queued	
inspect	检查交易信息,列出待处理交易的文本摘要	
status	查看交易池状态数	

```
#content查看交易明细
> txpool.content
{
  pending: {
    0x0216d5032f356960cd3749c31ab34eef21b3395: {
      0x24d407e5a0b506e1cb2fae163100b5de01f5193c: {
        34: [{
          blockHash:"0x0000000000000000000000000000000000000000000000000000000000000000",
            blockNumber: null,
            from: "0x24d407e5a0b506e1cb2fae163100b5de01f5193c",
            gas: "0x44c72",
            gasPrice: "0x4a817c800",
            hash: "0xb5b8b853af32226755a65ba0602f7ed0e8be2211516153b75e9ed640a7d359fe",
            input: "...",
            nonce: "0x22",
            to: "0x7320785200f74861b69c49e4ab32399a71b34f1a",
            transactionIndex: null,
            value: "0x0"
          }]
        }
      },
      queued: {
        0x976a3fc5d6f7d259ebfb4cc2ae75115475e9867c: {
          3: [{
            blockHash:
              "0x0000000000000000000000000000000000000000000000000000000000000000",
                blockNumber: null,
                from: "0x976a3fc5d6f7d259ebfb4cc2ae75115475e9867c",
                gas: "0x15f90",
                gasPrice: "0x4a817c800",
                hash: "0x57b30c59fc39a50e1cba90e3099286dfa5aaf60294a629240b5bbec6e2e66576",
                input: "0x",
                nonce: "0x3",
                to: "0x346fb27de7e7370008f5da379f74dd49f5f2f80f",
                transactionIndex: null,
                value: "0x1f161421c8e0000"
              }]
            }]
          }
        }
      }
    }
  }
}

#inspect检查交易内容
```

```

> txpool.inspect
{
  pending: {
    0x26588a9301b0428d95e6fc3a5024fce8bec12d51: {
      31813: ["0x3375ee30428b2a71c428afa5e89e427905f95f7e: 0 wei + 500000 ×
200000000000 gas"]
    },
    0x2a65aca4d5fc5b5c859090a6c34d164135398226: {
      563670: ["0x6dfef5bc94b031407ffe71ae8076ca0fbf190963: 1047859050000000000 wei +
90000 × 200000000000 gas"]
    },
    0xea674fdde714fd979de3edf0f56aa9716b898ec8: {
      70148: ["0xe39c55ead9f997f7fa20ebe40fb4649943d7db66: 1000767667434026200 wei +
90000 × 200000000000 gas"]
    }
  },
  queued: {
    0x0f6000de1578619320aba5e392706b131fb1de6f: {
      6: ["0x8383534d0bcd0186d326c993031311c0ac0d9b2d: 9000000000000000000 wei + 21000
× 200000000000 gas"]
    },
    0x976a3fc5d6f7d259ebfb4cc2ae75115475e9867c: {
      3: ["0x346fb27de7e7370008f5da379f74dd49f5f2f80f: 14000000000000000000 wei + 90000
× 200000000000 gas"]
    },
    0x9b11bf0459b0c4b2f87f8cebca4cfc26f294b63a: {
      7: ["0x6368f3f8c2b42435d6c136757382e4a59436a681: 17900000000000000000 wei +
90000 × 200000000000 gas"]
    }
  }
}
#status查看交易池状态
> txpool.status
{
  pending: 0,
  queued: 0
}

```

7.web3命令集

包含了以上对象，还包含一些单位换算的方法

命令名称	命令说明	备注
version.api	查询web3的版本信息	
version.node	查询节点路径	节点不是指区块链节点
version.network	查询启动端口地址	
version.ethereum	查看以太坊编号	
sha3(String,options)	用 Keccak-256 SHA3算法获取Hash值	
toHex(StringOrNumber)	转16进制	
toAscii(hexString)	转为Ascii编码	
fromAscii(textString,[padding])	Ascii转为十六进制数	
toDecimal(hexString)	16进制转为10进制实数	
fromDecimal(number)	实数转十六进制数	
fromWei(numberStringOrBigNumber,unit)	维转以太坊的单位,维/以太等	
toWei(numberStringOrBigNumber,unit)	转成维单位数字	
toBigNumber(numberOrHexString)	转大数	
isAddress(hexString)	检查给定的字符串是否为一个地址	

8.net命令集

包含以下查看p2p网络状态,连接节点数的方法

命令名称	命令说明	备注
listening/getListening	当前节点是否正常监听网络	只读属性
peerCount/getPeerCount	查询连接节点的数目	只读属性

```
#listening检查监听网络是否正常
> net.listening
true

#peerCount查询连接点的数目
> net.peerCount
0
```

9.附录

上述API主要参看以下接口地址:

api-wiki地址:<https://github.com/ethereum/wiki/wiki/JavaScript-API>

APIS:https://github.com/ethereum/go-ethereum/wiki/Management-APIs#admin_startrpc

