

PDX UTOPIA 区块链协议栈 JSON RPC 参考手册

版本 2.1.0

目 录

1 PDX Utopia 简介.....	1
2 协议和端口设置.....	1
3 区块链网络.....	2
3.1 net_version.....	2
3.2 net_listening.....	2
3.3 net_peerCount.....	3
4 区块链节点.....	4
4.1 eth_protocolVersion.....	4
4.2 eth_syncing.....	5
4.3 eth_coinbase.....	6
4.4 eth_mining.....	6
4.5 eth_gasPrice.....	7
4.6 eth_accounts.....	7
5 区块和交易.....	8
5.1 eth_pendingTransactions.....	9
5.2 eth_getTransactionByHash.....	10
5.3 eth_getTransactionReceipt.....	12
5.4 eth_getTransactionByBlockNumberAndIndex.....	13
5.5 eth_getTransactionByBlockHashAndIndex.....	14
5.6 eth_getBalance.....	14
5.7 eth_getStorageAt.....	15
5.8 eth_getTransactionCount.....	16
5.9 eth_getBlockTransactionCountByHash.....	16
5.10 eth_getBlockTransactionCountByNumber.....	17
5.11 eth_sendTransaction.....	18
5.12 eth_call.....	19
5.13 eth_getBlockByHash.....	20
5.14 eth_blockNumber.....	22
5.15 eth_getCode.....	22
5.16 eth_sign.....	23
5.17 eth_sendRawTransaction.....	24
5.18 eth_getCommitBlockByNormalBlockHash.....	25
5.19 eth_getCommitBlockByNormalBlockNumber.....	26
5.20 eth_getCommitBlockByHash.....	28
5.21 eth_getCommitBlockByNumber.....	29
5.22 eth_blockCommitNumber.....	31
5.23 eth_getBlockByNumber.....	32
5.24 eth_baapQuery.....	34
5.25 eth_currentCommitHash.....	35
6 过滤器.....	35

6.1	eth_newFilter.....	36
6.2	eth_newBlockFilter.....	36
6.3	eth_newPendingTransactionFilter.....	37
6.4	eth_uninstallFilter.....	38
6.5	eth_getFilterChanges.....	38
6.6	eth_getFilterLogs.....	40
6.7	eth_getLogs.....	40
7	数据库.....	41
7.1	db_putString.....	41
7.2	db_getString.....	42
7.3	db_putHex.....	43
7.4	db_getHex.....	44
8	共识和奖惩.....	44
8.1	eth_getLocalMinerRewardAddress.....	45
8.2	eth_getTotalAmountOfHypothecationWithAddress.....	45
8.3	eth_getNodeActivityInfoForAddress.....	46
8.4	eth_getHypothecationSuccessfulAddress.....	47
8.5	eth_getAmountofHypothecation.....	47
8.6	eth_getCurrentQuorumSize.....	48
8.7	eth_getConsensusQuorum.....	49
8.8	eth_getAddressIsInQuorum.....	49
8.9	eth_getNodeStatus.....	50
8.10	eth_getCurrentCommitHeight.....	51
8.11	eth_getNodeDisqualifiedReason.....	51
8.12	eth_getNodeUselessActiveInfoByAddress.....	52
8.13	eth_checkNodeDetail.....	53

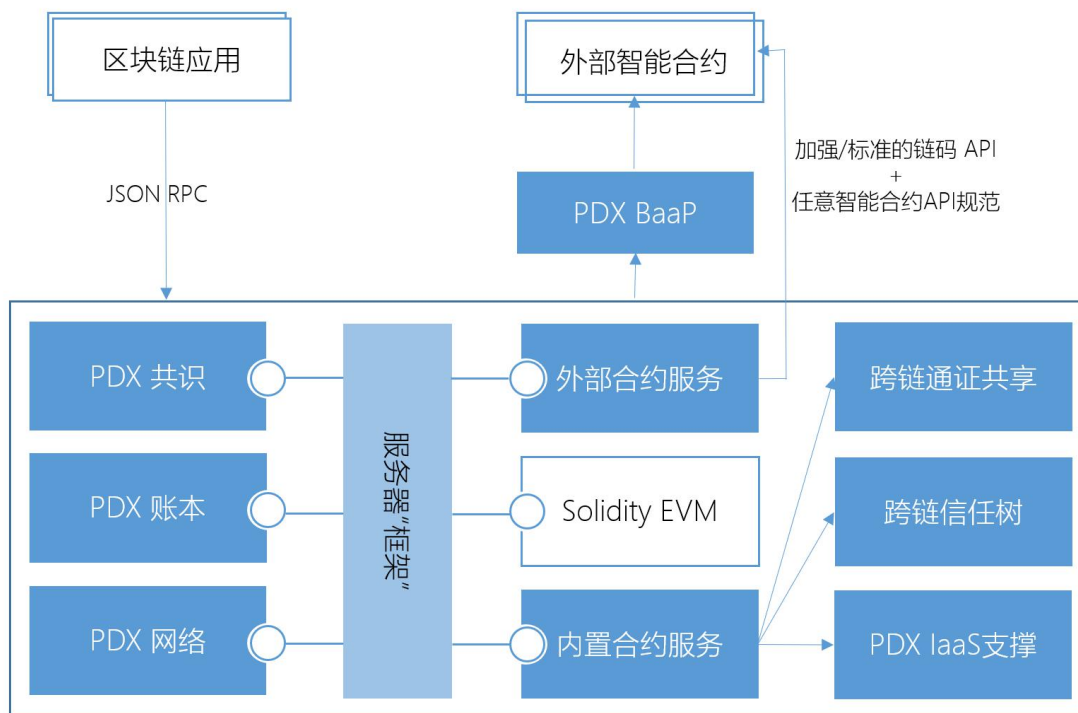
1 PDX Utopia 简介

PDX Utopia 区块链协议栈，基于专利保护的 PDX 共识算法、PDX 账本算法、PDX 智能合约架构，在安全、公平和多数共识的前提下，实现大规模、低延迟、高并发、高吞吐。PDX 共识算法，采用独创的双账本、非阻塞架构，在安全、公平、多数共识的前提下使区块确认的复杂度降到 $O(n)$ 。从算法上确保 PDX Utopia 区块链能够真正做到大规模、低延迟、高并发、高吞吐。PDX 共识的算法规范，可以从以下地址下载：
<https://github.com/PDXbaap/design/blob/master/pdx-consensus.pdf>。

PDX Utopia 区块链协议栈，内生兼容以太坊的 Solidity 合约、以太坊钱包、ERC20、eWASM 和超级账本 chaincode 合约，并可扩展兼容未来的智能合约 API 接口规范。

PDX Utopia 内置的“跨链通证共享”智能合约，支持性能真正无损的、安全的跨链通证共享，方便形成多链跨链的激励生态。

PDX Utopia 内置的“跨链信任树”和“PDX IaaS 支撑”智能合约，实现跨链的信任背书以及与 PDX SureDigit 可信数字生态平台的融合。



区块链应用，基于 JSON RPC，与 PDX Utopia 交互。本手册是 PDX Utopia 的 JSON RPC API 规范。

有关 PDX Utopia 的其他情况，请访问 <https://www.pdx.ltd/> 或 <https://github.com/pdxbaap>。

2 协议和端口设置

PDX Utopia JSON RPC 协议，兼容并扩展以太坊的 JSON RPC。PDX Utopia JSON RPC 经由 HTTP/S 和 websocket (WS 或 WSS) 传输。JSON RPC 默认关闭，但可以通过启动参数 `--rpc` 打开。HTTP/S 的默认端口是 8545，但可以通过启动参数 `--rpcport+端口配置`。websocket 默认关闭，但可以通过启动参数 `--ws` 打开。websocket 默认端口是 8546，但可以通过 `--wsport+`

端口配置。

3 区块链网络

net_version	查询当前节点的网络 id
net_listening	查询客户端监听网络连接是否活跃
net_peerCount	查询当前客户端链接的节点

3.1 net_version

API 描述

查询当前节点的网络 id。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
int	网络号

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"net_version","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"739"}
```

3.2 net_listening

API 描述

查询客户端监听网络连接是否活跃。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	Boolean 监听中为 true

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"net_listening","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":true}
```

3.3 net_peerCount

API 描述

查询当前客户端链接的节点。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	int 链接的节点数量

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"net_peerCount","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"0x2" }
```

4 区块链节点

eth_protocolVersion	查询当前的协议版本号
eth_syncing	查询同步状态与否
eth_coinbase	查询客户端的账户地址
eth_mining	查询客户端是否在活跃打块
eth_gasPrice	查询当前 gas 的消耗
eth_accounts	查询当前客户端拥有的账户列表

4.1 eth_protocolVersion

API 描述

查询当前的协议版本号。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_protocolVersion","params":[],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": "0x54"
}
```

4.2 eth_syncing

API 描述

查询同步状态与否。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	Boolean 在同步则返回 true
	startingBlock 开始的区块
	currentBlock 当前区块
	highestBlock 最高的区块

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_syncing","params":[],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": {
    startingBlock: '0x384',
    currentBlock: '0x386',
    highestBlock: '0x454'
  }
}
// Or when not syncing
{
  "id":739,
  "jsonrpc": "2.0",
  "result": false
}
```


4.3 eth_coinbase

API 描述

查询客户端的账户地址。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	20bytes 当前账户地址

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_coinbase","params":[],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc": "2.0",  
  "result": "0xc94770007dda54cF92009BFF0dE90c06F603a09f"  
}
```

4.4 eth_mining

API 描述

查询客户端是否在活跃打块。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号, 2.0
id	RPC 请求编号
result	调用结果
	Boolean 挖矿中为 true, 其他为 false

示例代码

以 curl 为例的调用代码如下:

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_mining","params":[],"id":739}'
```

返回结果如下:

```
{"jsonrpc":"2.0","id":739,"result":true}
```

4.5 eth_gasPrice

API 描述

查询当前 gas 的消耗。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象, 字段如下:

jsonrpc	RPC 版本号, 2.0
id	RPC 请求编号
result	调用结果
	int 当前 gas 价格单位是 wei

示例代码

以 curl 为例的调用代码如下:

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_gasPrice","params":[],"id":739}'
```

返回结果如下:

```
{"jsonrpc":"2.0","id":739,"result":"0x09184c72a000" // 10000000000000000}
```

4.6 eth_accounts

API 描述

查询当前客户端拥有的账户列表。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	20Bytes 客户端的地址集合

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": ["0xc94770007dda54cF92009BFF0dE90c06F603a09f"]
}
```

5 区块和交易

eth_pendingTransactions	查询处于 pending 状态的交易列表
eth_getTransactionByHash	通过交易哈希获取交易
eth_getTransactionReceipt	通过交易哈希查询交易的收据
eth_getTransactionByBlockNumberAndIndex	根据区块号和交易序列查询交易
eth_getTransactionByBlockHashAndIndex	根据区块哈希和交易序列查询交易
eth_getBalance	查询当前需要查询的账户余额
eth_getStorageAt	查询给定地址的存储位置返回值
eth_getTransactionCount	查询发送到给定地址的交易量
eth_getBlockTransactionCountByHash	查询给定区块哈希对应的区块号中包含的交易数量
eth_getBlockTransactionCountByNumber	查询给定区块号对应的区块号中包含的交易
eth_sendTransaction	如果数据字段包含代码，创建交易或合约
eth_call	执行一个链上没有的交易
eth_getBlockByHash	通过区块哈希查询区块
eth_blockNumber	查询最近的区块

eth_getCode	查询给定地址的编码
eth_sign	将信息通过以太坊特定的签名算法签名
eth_sendRawTransaction	为已签名的交易创建新的消息调用交易或合约
eth_getCommitBlockByNormalBlockHash	通过普通区块的哈希查找对应的 commit 区块
eth_getCommitBlockByNormalBlockNumber	通过普通区块的区块号查找对应的 commit 区块
eth_getCommitBlockByHash	通过区块哈希查找对应的 commit 区块
eth_getCommitBlockByNumber	通过给定区块号查询 commit 区块
eth_blockCommitNumber	查询当前 commit 区块的区块号
eth_getBlockByNumber	通过给定的区块号查询 normal 区块
eth_baapQuery	chaincode 合约查询
eth_currentCommitHash	查询当前 commit 区块的哈希

5.1 eth_pendingTransactions

API 描述

查询处于 pending 状态的交易列表。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	array pending 交易的列表

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_pendingTransactions","params":[],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc":"2.0",
  "result": [{
    blockHash:
    '0x0000000000000000000000000000000000000000000000000000000000000000',
```

```
    blockNumber: null,
    from: '0x28bdb9c230f4d5e45435e4d006326ec32e46cb31',
    gas: '0x204734',
    gasPrice: '0x4a817c800',
    hash:
'0x8dfa6a59307a490d672494a171feec09db511f05e9c097e098edc2881f9ca4f6',
    input: '0x6080604052600',
    nonce: '0x12',
    to: null,
    transactionIndex: '0x0',
    value: '0x0',
    v: '0x3d',
    r: '0xaabc9ddaaffb2ae0bac4107697547d22d9383667d9e97f5409dd6881ce08f13f',
    s: '0x69e43116be8f842dcd4a0b2f760043737a59534430b762317db21d9ac8c5034'
  },...,{
    blockHash:
'0x0000000000000000000000000000000000000000000000000000000000000000',
    blockNumber: null,
    from: '0x28bdb9c230f4d5e45435e4d006326ec32e487b31',
    gas: '0x205940',
    gasPrice: '0x4a817c800',
    hash:
'0x8e4340ea3983d86e4b6c44249362f716ec9e09849ef9b6e3321140581d2e4dac',
    input: '0xe4b6c4424936',
    nonce: '0x14',
    to: null,
    transactionIndex: '0x0',
    value: '0x0',
    v: '0x3d',
    r: '0x1ec191ef20b0e9628c4397665977cbe7a53a263c04f6f185132b77fa0fd5ca44',
    s: '0x8a58e00c63e05cfeae4f1cf19f05ce82079dc4d5857e2cc281b7797d58b5faf'
  }]
}
```

5.2 eth_getTransactionByHash

API 描述

通过交易哈希获取交易。

API 调用

POST/api

参数说明

32Bytes 交易哈希

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
blockHash	包含交易的区块哈希
blockNumber	包含交易的区块号
from	发送者的地址
to	交易接受者
gas	发送者提供的 gas
gasPrice	发送者提供的 gas 价格单位是 wei
hash	交易哈希
input	交易的输入
nonce	交易创建者提供的号码
transactionIndex	交易的序列号
value	转账的金额单位是 wei
v	ECDSA 签名中的 v
r	ECDSA 签名中的 r
s	ECDSA 签名中的 s

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getTransactionByHash","params":
["0x88df016429689c079f3b2f6ad39fa052532c56795b733da78a91ebe6a713944b"],"
id":739}'
```

返回结果如下：

```
{
  "jsonrpc":"2.0",
  "id":739,
  "result":{

    "blockHash":"0x1d59ff54b1eb26b013ce3cb5fc9dab3705b415a67127a003c3e61eb445bb
8df2",
    "blockNumber":"0x5daf3b", // 6139707
    "from":"0xa7d9ddbe1f17865597fbd27ec712455208b6b76d",
    "gas":"0xc350", // 50000
    "gasPrice":"0x4a817c800", // 200000000000

    "hash":"0x88df016429689c079f3b2f6ad39fa052532c56795b733da78a91ebe6a713944b",
    "input":"0x68656c6c6f21",
    "nonce":"0x15", // 21
```

```

    "to": "0xf02c1c8e6114b1dbe8937a39260b5b0a374432bb",
    "transactionIndex": "0x41", // 65
    "value": "0xf3dbb76162000", // 4290000000000000
    "v": "0x25", // 37

    "r": "0x1b5e176d927f8e9ab405058b2d2457392da3e20f328b16ddabcebc33eaac5fea",

    "s": "0x4ba69724e8f69de52f0125ad8b3c5c2cef33019bac3249e2c0a2192766d1721c"
  }
}

```

5.3 eth_getTransactionReceipt

API 描述

通过交易哈希查询交易的收据。

API 调用

POST/api

参数说明

32Bytes 交易哈希

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
transactionHash	交易哈希
transactionIndex	交易序列
blockHash	包含交易的区块哈希
blockNumber	包含交易的区块号
from	发送者的地址
to	交易接受者
cumulativeGasUsed	交易执行需要的全部 gas
gasUsed	特殊交易需要的 gas
contractAddress	合约地址
logs	日志的集合
logsBloom	日志过滤器

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
```

```
{ "jsonrpc": "2.0", "method": "eth_getTransactionReceipt", "params":  
  ["0xb903239f8543d04b5dc1ba6579132b143087c68db1b2168786408fcbce568238"],  
  "id": 739 }
```

返回结果如下：

```
{  
  "id": 739,  
  "jsonrpc": "2.0",  
  "result": {  
    transactionHash:  
      '0xb903239f8543d04b5dc1ba6579132b143087c68db1b2168786408fcbce568238',  
    transactionIndex: '0x1', // 1  
    blockNumber: '0xb', // 11  
    blockHash:  
      '0xc6ef2fc5426d6ad6fd9e2a26abeab0aa2411b7ab17f30a99d3cb96aed1d1055b',  
    cumulativeGasUsed: '0x33bc', // 13244  
    gasUsed: '0x4dc', // 1244  
    contractAddress: '0xb60e8dd61c5d32be8058bb8eb970870f07233155', // or null, if  
    none was created  
    logs: [{  
      // logs as returned by getFilterLogs, etc.  
    }, ...],  
    logsBloom: "0x00...0", // 256 byte bloom filter  
    status: '0x1'  
  }  
}
```

5.4 eth_getTransactionByBlockNumberAndIndex

API 描述

根据区块号和交易序列查询交易。

API 调用

POST/api

参数说明

int/string	区块号，或字符串`"earliest"`、`"latest"`、`"pending"`
int	交易的序列号

API 返回结果

与 eth_getTransactionByHash 返回值（3.2）相同。

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getTransactionByBlockNumberAndIndex","params":  
["0x29c", "0x0"],"id":739}'
```

返回结果如下：

与 eth_getTransactionByHash 返回值（3.2）相同。

5.5 eth_getTransactionByBlockHashAndIndex

API 描述

根据区块哈希和交易序列查询交易。

API 调用

POST/api

参数说明

32Bytes	区块哈希
int	交易索引位置

API 返回结果

与 eth_getTransactionByHash 返回值（3.2）相同。

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getTransactionByBlockHashAndIndex","params":  
["0xc6ef2fc5426d6ad6fd9e2a26abcab0aa2411b7ab17f30a99d3cb96acd1d1055b",  
"0x0"],"id":739}'
```

返回结果如下：

与 eth_getTransactionByHash 返回值（3.2）相同。

5.6 eth_getBalance

API 描述

查询当前需要查询的账户余额。

API 调用

POST/api

参数说明

20Byte	查询账户的地址
--------	---------

int/string 区块号或者字符串 "latest", "earliest", "pending"

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
int	当前余额单位是 wei

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getBalance","params":["0xE05e11D6272Af1A13f1843
26dadF4079Cfa8c31d","latest"],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": "0xc94" // 1207
}
```

5.7 eth_getStorageAt

API 描述

查询给定地址的存储位置返回值。

API 调用

POST/api

参数说明

20Bytes	存储的地址
int	存储的位置
int/string	区块号或字符串 "latest", "earliest" 或者 "pending"

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，存储值的位置值

示例代码

以 curl 为例的调用代码如下：

API 调用

POST/api

参数说明

32Bytes 区块的哈希

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，给定区块包含的交易量

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getBlockTransactionCountByHash","params":
["0xc94770007dda54cf92009BFF0dE90c06F603a09f"],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": "0xc"
}
```

5.10 eth_getBlockTransactionCountByNumber

API 描述

查询给定区块号对应的区块号中包含的交易。

API 调用

POST/api

参数说明

int/string 区块号，或字符串`"earliest"`、`"latest"`、`"pending"`

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，给定区块包含的交易量

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getBlockTransactionCountByNumber","params":  
["0xe8"],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc": "2.0",  
  "result": "0xc"  
}
```

5.11 eth_sendTransaction

API 描述

如果数据字段包含代码，创建交易或合约。

API 调用

POST/api

参数说明

from	发送交易的地址
to	交易发送给地址
gas	(默认：90000)提供给交易得以执行的 gas
gasPrice	每种付费 gas 所使用的 gasprice
nonce	nonce 值
data	编译的代码或调用方法签名的散列和编码参数

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，交易的哈希

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_sendTransaction","params":[{"  
  "from": "0xb60e8dd61c5d32be8058bb8eb970870f07233155",  
  "to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567",  
  "gas": "0x76c0", // 30400  
  "gasPrice": "0x9184e72a000", // 10000000000000  
}]}'
```

```
"value": "0x9184e72a", // 2441406250
"data":
"0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970
870f072445675"}], "id": 739}'
```

返回结果如下：

```
{
  "id": 739,
  "jsonrpc": "2.0",
  "result":
"0xe670ec64341771606e55d6b4ca35a1a6b75ec3d5145a99d05921026d1527331"
}
```

5.12 eth_call

API 描述

执行一个链上没有的交易。

API 调用

POST/api

参数说明

from	发送交易的地址
to	交易发送给地址
gas	(默认: 90000)提供给交易得以执行的 gas
gasPrice	每种付费 gas 所使用的 gasprice
value	这个交易的 value 值
data	编译的代码或调用方法签名的散列和编码参数

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，合约执行的值

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_call","params":[{"
  "from": "0xb60e8dd61c5d32be8058bb8eb970870f07233155",
  "to": "0xd46e8dd67c5d32be8058bb8eb970870f07244567",
  "gas": "0x76c0", // 30400
  "gasPrice": "0x9184e72a000", // 1000000000000000

```

```
"value": "0x9184e72a", // 2441406250
"data":
"0xd46e8dd67c5d32be8d46e8dd67c5d32be8058bb8eb970870f072445675058bb8eb970
870f072445675"}}, "id": 739}'
```

返回结果如下：

```
{
  "id": 739,
  "jsonrpc": "2.0",
  "result": "0x"
}
```

5.13 eth_getBlockByHash

API 描述

通过区块哈希查询区块。

API 调用

POST/api

参数说明

32Bytes	区块哈希
Boolean	true, 显示区块中交易的全部细节, false, 仅显示区块中交易的哈希

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，合约执行的值
Hash	区块哈希
parentHash	前一区块的哈希
Nonce	生成区块的工作量证明
logsBloom	日志过滤器
transactionsRoot	交易树的哈希
stateRoot	最终状态树的根哈希
commitNumber	对应的 commit 区块号
receiptsRoot	收据树的根哈希
Miner	区块产生者的地址
Difficulty	区块的难度
totalDifficulty	指定这个区块的整个链的难度
extraData	区块的额外信息
size	区块的大小
asLimit	区块允许的最大的 gas

asUsed	区块中所有交易消耗的 gas
Timestamp	区块被生产出来的时间戳
Transactions	所有交易的集合

示例代码

以 curl 为例的调用代码如下:

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getBlockByHash","params":
["0xe670ec64341771606e55d6b4ca35a1a6b75ee3d5145a99d05921026d1527331",
false],"id":739}'
```

返回结果如下:

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result": {
    "difficulty": "0x400",
    "commitNumber": "0x19",
    "extraData":
      "0xf84db8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f051637be
      14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b801a8080
      80808080808080",
    "gasLimit": "0x4dd30d2",
    "gasUsed": "0x0",
    "hash":
      "0x6fd6d365acaf854d8d1be8fe91e5ae5cda5018ca71a8fd3b212425392e500a0a",
    "logsBloom":
      "0x0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
      000000000000",
    "miner": "0xe05e11d6272af1a13f184326dadf4079cfa8c31d",
    "nonce": "0x0000000000000000",
    "number": "0x103",
    "parentHash":
      "0x6c25b2a403138cb287e6a622f8e6ee100d4f4bfbbbc4dceac4ba185558456be4",
    "receiptsRoot":
      "0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e363b421",
    "rewardAddr": "0xe05e11d6272af1a13f184326dadf4079cfa8c31d",
    "size": "0x2ab",
    "stateRoot":
      "0x46e226056cea1582d326b7ebac9e6db1e72e46c4560328e207f6334685d728b0",
```



```
        "timestamp": "0x5ef1c3c0",
        "totalDifficulty": "0x41c00",
        "transactions": [],
        "transactionsRoot":
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cad001622fb5e363b421"
    }
}
```

5.14 eth_blockNumber

API 描述

查询最近的区块。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，合约执行的值
	int 当前区块号

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": "0xc94" // 1207
}
```

5.15 eth_getCode

API 描述

查询给定地址的编码。

API 调用

POST/api

参数说明

20Byte 地址
int/string 区块号或"latest","earliest","pending"

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc RPC 版本号，2.0
id RPC 请求编号
result 调用结果，合约执行的值
从给定地址的代码

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getCode","params":["0xa94f5374f5e5d3b8e2a8697c1
5331677e6ebf0b"],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result":
    "0x600160008035811a818181146012578301005b601b6001356025565b8060005260206
    000f25b600060078202905091905056"
}
```

5.16 eth_sign

API 描述

将信息通过以太坊特定的签名算法签名。

API 调用

POST/api

参数说明

20Byte 地址
NByte 需要签名的信息

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果，合约执行的值 签名数据

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_sign","params":["0x9b2055d370f73ec7d8a03e9651291  
18dc8f5bf83", "0xdeadbeaf"],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc": "2.0",  
  "result":  
  "0xa3f20717a250c2b0b729b7e5becbfff67fdae7e0699da4de7ca5895b02a170a12d887fd3  
b17bfdce3481f10bea41f45ba9f709d39ce8325427b57afcf994cee1b"  
}
```

5.17 eth_sendRawTransaction

API 描述

为已签名的交易创建新的消息调用交易或合约。

API 调用

POST/api

参数说明

签名之后的交易数据

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果 32Bytes 交易哈希

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_sendRawTransaction",
```

```
"params":["0xf89581b78504a817c8008347e7c49493439fd79141e5d86ca182b6caa07f16a1c87ec380ad080112290a037075741203706478120768656c6c6f31311a140a0c626161702d74782d747970651204657865638205e9a08bb3a1f5638ca7635c73cc61a33f646e6646aaa-b70198c9b9035225285a96ae8a002a88b29b7b344328332dfec764dbc3b6a0ed0f1f1fad64ae3e70a9777784ae"],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result":
    "0xc670ec64341771606e55d6b4ca35a1a6b75ec3d5145a99d05921026d1527331"
}
```

5.18 eth_getCommitBlockByNormalBlockHash

API 描述

通过普通区块的哈希查找对应的 commit 区块。

API 调用

POST/api

参数说明

32byte 普通区块哈希

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	Hash 区块哈希
	Number 区块号
	parentHash 前一区块哈希
	Size 区块大小
	Timestamp 时间戳
	extraData 额外信息

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data '{"jsonrpc":"2.0",
"method":"eth_getCommitBlockByNormalBlockHash","params":["0x44fec6160987f55
0c35c126d7801af444b10264e8dfa8f53038e3d7a3cd2049a"],"id":739}'
```

返回结果如下：

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result": {
    "extraData":
      "0xf902deb8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f051637
      be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b801ab8
      41aab9f76885b3a6c45476df1ba2543e0590842d4a9fe8c3a4e25994662efac042372228e33
      37f58338980c9b9f51159d8be39e234827d874c12c80e3827d17ca80180808080b9024df9
      024a81faf9014aa06dc385877edf995d3bc66b323a8779f019e3c1f22c7e6f9c784b4402cfd
      0df04a022c934bdb2d2293b4c2ca8a19f4a4b820d104b30a92afbad318897d5549e6f28a02
      1b2c00d8ad4ab6c2fc3b6e439f31f8b763d577e3fcc44dedc148ad5d3be4502a0523bb3fa5
      7052de5efd7b18773b3794efaaec992213af9a6eb23ad38ce3a4b0a09f24fdf2c1c79777326
      ac7f9c7775d4b540f7ef4489409ee1e36d5b897635680a03b9bfb95e7432b8401c7ebfad4c
      0adb11d2e79fede064a322502947a418e4acfa028b60039874cdf3a4137564a9e63a2a0a544
      285ac672d9600284b6d17a48424ea00b61fac91cdb470c0fcbdaefdd5f04d725f173d184a9
      65f52e7bf7460d7b8e8da0839c2b5311937e3cd5e95743501fd31919cbb9258c9823a8d3b
      5767103e1040fa0abc2dc7f383236c1ef6b4508bf9f02e832fd15fd8a591d0f66ac753e469d
      8bd1f8aaf8a8b8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f0516
      37be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b80c0
      a045b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9cb84189f4
      bc004cb5783902a90de143b8831c396022421d9cd72e821bf2914668647352d191c67c6de
      825a25faf4e684eb5de4eeacb2003a11325921e64bb468226e200c0c0c080808019b8403
      161313439306337666233616637343035353930643732653165363961343566383136323
      7356536326663383430316564666664636566363732393766636435c3c0808080808080",
    "hash":
      "0x3f6df788f0693baa2c5682c6830dfcf978ce7fe7aad77fc3c6d0134cd20bc6a5",
    "number": "0x1a",
    "parentHash":
      "0x45b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9c",
    "size": "0x53d",
    "timestamp": "0x5ef1c3b5"
  }
}
```

5.19 eth_getCommitBlockByNormalBlockNumber

API 描述

通过普通区块的区块号查找对应的 commit 区块。

API 调用

POST/api

参数说明

int 区块号

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
Hash	区块哈希
Number	区块号
parentHash	前一区块哈希
Size	区块大小
Timestamp	时间戳
extraData	额外信息

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data '{
  "jsonrpc": "2.0",
  "method": "eth_getCommitBlockByNormalBlockHash",
  "params": ["0x15"/"/21], "id": 739}]'
```

返回结果如下：

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result": {
    "extraData":
      "0xf902deb8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f051637
      be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b801ab8
      41aab9f76885b3a6c45476df1ba2543e0590842d4a9fe8c3a4c25994662efac042372228e33
      37f58338980c9b9f51159d8be39e234827d874c12c80e3827d17ca80180808080b9024df9
      024a81faf9014aa06dc385877edf995d3bc66b323a8779f019e3c1f22c7e6f9c784b4402cfd
      0df04a022c934bdb2d2293b4c2ca8a19f4a4b820d104b30a92afbad318897d5549e6f28a02
      1b2c00d8ad4ab6c2fc3b6e439f31f8b763d577e3fcc44dedc148ad5d3be4502a0523bb3fa5
      7052de5efd7b18773b3794efaaec992213af9a6eb23ad38ce3a4b0a09f24fdf2c1c79777326
      ac7f9c7775d4b540f7ef4489409ee1e36d5b897635680a03b9bfb95e7432b8401c7ebfad4c
      0adb11d2e79fede064a322502947a418e4acfa028b60039874cdf3a4137564a9e63a2a0a544
      285ac672d9600284b6d17a48424ea00b61fac91cdb470c0fcbdaefdd5f04d725f173d184a9
      65f52e7bf7460d7b8e8da0839c2b5311937e3ed5e95743501fd31919cbb9258c9823a8d3b
      5767103e1040fa0abc2dc7f383236c1ef6b4508bf9f02e832fd15fd8a591d0f66ac753e469d
      8bd1f8aaf8a8b8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f0516
      37be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b80c0
      a045b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9cb84189f4
      bc004cb5783902a90de143b8831c396022421d9cd72e821bf2914668647352d191c67c6de
      825a25faf4e684eb5de4eeacb2003a11325921e64bb468226e200c0c0c080808019b8403"
```

```
161313439306337666233616637343035353930643732653165363961343566383136323
7356536326663383430316564666664636566363732393766636435c3c08080808080",
    "hash":
    "0x3f6df788f0693baa2c5682c6830dfcf978ce7fe7aad77fc3c6d0134cd20bc6a5",
    "number": "0x1a",
    "parentHash":
    "0x45b7d827400ec553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9c",
    "size": "0x53d",
    "timestamp": "0x5ef1c3b5"
  }
}
```

5.20 eth_getCommitBlockByHash

API 描述

通过区块哈希查找对应的 commit 区块。

API 调用

POST/api

参数说明

32byte commit 区块哈希

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	Hash 区块哈希
	Number 区块号
	parentHash 前一区块哈希
	Size 区块大小
	Timestamp 时间戳
	extraData 额外信息

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getCommitBlockByHash","params":["0x3f6df788f069
3baa2c5682c6830dfcf978ce7fe7aad77fc3c6d0134cd20bc6a5"],"id":739}'
```

返回结果如下：

```
{
```

```
"jsonrpc": "2.0",
"id": 739,
"result": {
  "extraData":
    "0xf902deb8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f051637
    be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b801ab8
    41aab9f76885b3a6c45476df1ba2543c0590842d4a9fe8c3a4c25994662efac042372228e33
    37f58338980c9b9f51159d8be39e234827d874c12c80e3827d17ca80180808080b9024df9
    024a81fa9014aa06dc385877edf995d3bc66b323a8779f019e3c1f22c7e6f9c784b4402cfd
    0df04a022c934bdb2d2293b4c2ca8a19f4a4b820d104b30a92afbad318897d5549e6f28a02
    1b2c00d8ad4ab6c2fc3b6e439f31f8b763d577e3fcc44dedc148ad5d3be4502a0523bb3fa5
    7052de5efd7b18773b3794efaaec992213af9a6eb23ad38ce3a4b0a09f24fdf2c1c79777326
    ac7f9c7775d4b540f7ef4489409ce1e36d5b897635680a03b9bfb95e7432b8401c7ebfad4c
    0adb11d2e79fede064a322502947a418e4acfa028b60039874cdf3a4137564a9e63a2a0a544
    285ac672d9600284b6d17a48424ea00b61fac91cdb470c0fcbdaefdd5f04d725f173d184a9
    65f52e7bf7460d7b8e8da0839c2b5311937e3ced5e95743501fd31919cbb9258c9823a8d3b
    5767103e1040fa0abc2dc7f383236c1ef6b4508bf9f02e832fd15fd8a591d0f66ac753e469d
    8bd1f8aaf8a8b8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f0516
    37be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b80c0
    a045b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9cb84189f4
    bc004cb5783902a90de143b8831c396022421d9cd72e821bf2914668647352d191c67c6de
    825a25faf4e684eb5de4eeacb2003a11325921e64bb468226c200c0c0c080808019b8403
    161313439306337666233616637343035353930643732653165363961343566383136323
    7356536326663383430316564666664636566363732393766636435c3c0808080808080",
  "hash":
    "0x3f6df788f0693baa2c5682c6830dfcf978ce7fe7aad77fc3c6d0134cd20bc6a5",
  "number": "0x1a",
  "parentHash":
    "0x45b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9c",
  "size": "0x53d",
  "timestamp": "0x5ef1c3b5"
}
```

5.21 eth_getCommitBlockByNumber

API 描述

通过给定区块号查询 commit 区块。

API 调用

POST/api

参数说明

int64

区块号

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0	
id	RPC 请求编号	
result	调用结果	
	Hash	区块哈希
	Number	区块号
	parentHash	前一区块哈希
	Size	区块大小
	Timestamp	时间戳
	extraData	额外信息

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getCommitBlockByNumber","params":["0x56"],"id":739}'
```

返回结果如下：

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result": {
    "extraData":
      "0xf902deb8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f051637
      be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b801ab8
      41aab9f76885b3a6c45476df1ba2543ce590842d4a9fe8c3a4e25994662efac042372228e33
      37f58338980c9b9f51159d8be39e234827d874c12c80e3827d17ca801808080b9024df9
      024a81faf9014aa06dc385877edf995d3bc66b323a8779f019e3c1f22c7e6f9c784b4402cfd
      0df04a022c934bdb2d2293b4c2ca8a19f4a4b820d104b30a92afbad318897d5549e6f28a02
      1b2c00d8ad4ab6c2fc3b6c439f31f8b763d577e3fcc44dedc148ad5d3be4502a0523bb3fa5
      7052de5efd7b18773b3794efaaec992213af9a6eb23ad38ce3a4b0a09f24fdf2c1c79777326
      ac7f9c7775d4b540f7ef4489409ee1e36d5b897635680a03b9bfb95e7432b8401c7ebfad4c
      0adb11d2e79fede064a322502947a418e4acfa028b60039874cdf3a4137564a9e63a2a0a544
      285ac672d9600284b6d17a48424ea00b61fac91cdb470c0fcbdaefdd5f04d725f173d184a9
      65f52e7bf7460d7b8e8da0839c2b5311937e3ced5e95743501fd31919cbb9258c9823a8d3b
      5767103e1040fa0abc2dc7f383236c1ef6b4508bf9f02e832fd15fd8a591d0f66ac753c469d
      8bd1f8aaf8a8b8400e1e1acf776209f6e10a203539bd596d8c22382e105abd806bc0a7f0516
      37be14af0ce187b425b9acb8b92131a797b4420be8443161324269b053d81869e600b80c0
      a045b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9cb84189f4
      bc004cb5783902a90de143b8831c396022421d9cd72e821bf2914668647352d191c67c6de
      825a25faf4e684eb5de4eeacb2003a11325921e64bb468226e200c0c0c080808019b8403
      161313439306337666233616637343035353930643732653165363961343566383136323
```

```

7356536326663383430316564666664636566363732393766636435c3c08080808080",
    "hash":
    "0x3f6df788f0693baa2c5682c6830dfcf978ce7fe7aad77fc3c6d0134cd20bc6a5",
    "number": "0x1a",
    "parentHash":
    "0x45b7d827400ee553c9f66ddc1308c52715cd7201e6530385024013b22ffdd9c",
    "size": "0x53d",
    "timestamp": "0x5ef1c3b5"
  }
}

```

5.22 eth_blockCommitNumber

API 描述

查询当前 commit 区块的区块号。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	uint64 区块号

示例代码

以 curl 为例的调用代码如下：

```

$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_blockCommitNumber","params":[ ],"id":739}'

```

返回结果如下：

```

{
  "jsonrpc": "2.0",
  "id": 739,
  "result": "0x289" //649
}

```

5.23 eth_getBlockByNumber

API 描述

通过给定的区块号查询 normal 区块。

API 调用

POST/api

参数说明

int64	区块号
bool	true,显示区块中交易的全部细节, false,仅显示区块中交易的哈希

API 返回结果

返回结果为一个 JSON 对象, 字段如下:

jsonrpc	RPC 版本号, 2.0
id	RPC 请求编号
result	调用结果
Hash	区块哈希
parentHash	前一区块的哈希
Nonce	生成区块的工作量证明
logsBloom	日志过滤器
transactionsRoot	交易树的哈希
stateRoot	最终状态树的根哈希
commitNumber	对应的 commit 区块号
receiptsRoot	收据树的根哈希
Miner	区块产生者的地址
Difficulty	区块的难度
totalDifficulty	指定这个区块的整个链的难度
extraData	区块的额外信息
size	区块的大小
asLimit	区块允许的最大的 gas
asUsed	区块中所有交易消耗的 gas
Timestamp	区块被生产出来的时间戳
Transactions	所有交易的集合

示例代码

以 curl 为例的调用代码如下:

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["0x56",false],"id":739
}'
```

返回结果如下:

[illegible]

5.24 eth_baapQuery

API 描述

chaincode 合约查询。

API 调用

POST/api

参数说明

params chaincode Transaction 对象 protobuf 序列化的 bytecode 的 HEX 字符串
chaincode protobuf 消息结构如下:

```
message Transaction {
    uint32 type = 1; // 1
    bytes payload = 2; // Invocation
}
message Invocation {
    string fcn = 1; // 合约方法
    repeated bytes args = 2; // 方法的参数
    map<string, bytes> meta = 3; // { "baap-tx-type" ->
        "query", "to" -> chaincode 智能合约地址 }
}
```

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	Bytes 查询的结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_baapQuery","params":
:["0x080112510a0367657412037064781a150a0c626161702d74782d74797065120571756
572791a2e0a02746f12283933343339666437393134316535643836636131383262366361
6130376631366131633837656333"],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc": "2.0",
  "result": "0x68656c6c6f3131"
}
```

5.25 eth_currentCommitHash

API 描述

查询当前 commit 区块的哈希。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	string 区块哈希(字符串形式)

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_currentCommitHash","params":[],"id":739}'
```

返回结果如下：

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result":
    "0x181880d5255b290dbd20a5f81cf249234c315f84fc119f297fb6a1d4d2d9f423"
}
```

6 过滤器

eth_newFilter	创建基于过滤器选项的过滤器对象，在状态发生变化时获取通知
eth_newBlockFilter	在节点中创建一个过滤器，在新区块到达的时候发送通知
eth_newPendingTransactionFilter	在节点中创建一个过滤器，当产生挂起交易时进行通知
eth_uninstallFilter	通过给定的 id 卸载一个过滤器
eth_getFilterChanges	过滤器的投票方法，该方法返回上次投票后发生的一系列日志
eth_getFilterLogs	给定 id 匹配的日志列表
eth_getLogs	获取日志的集合

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_newBlockFilter","params":[],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc": "2.0",  
  "result": "0x1"  
}
```

6.3 eth_newPendingTransactionFilter

API 描述

在节点中创建一个过滤器，当产生挂起交易时进行通知。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_newPendingTransactionFilter","params":[],"id":739}'
```

返回结果如下：

```
{
```



```
{
  "id": 739,
  "jsonrpc": "2.0",
  "result": "0x1"
}
```

6.4 eth_uninstallFilter

API 描述

通过给定的 id 卸载一个过滤器。

API 调用

POST/api

参数说明

过滤器 id

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_uninstallFilter","params":["0xb" //11],"id":739}'
```

返回结果如下：

```
{
  "id": 739,
  "jsonrpc": "2.0",
  "result": true
}
```

6.5 eth_getFilterChanges

API 描述

滤波器的投票方法，该方法返回上次投票后发生的一系列日志。

API 调用

POST/api

参数说明

过滤器 id

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
removed	"true" 日志已被移除
logIndex	日志的序号
transactionIndex	交易的序列
transactionHash	交易哈希
blockHash	区块的哈希
blockNumber	区块的号码
address	日志起源的地址
data	包含日志中非索引的参数
topics	索引日志参数的参考

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getFilterChanges","params":["0x16"],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc":"2.0",  
  "result": [{  
    "logIndex": "0x1", // 1  
    "blockNumber": "0x1b4", // 436  
    "blockHash":  
    "0x8216c5785ac562ff41e2dcfdf5785ac562ff41e2dcfdf829c5a142f1fccd7d",  
    "transactionHash":  
    "0xdf829c5a142f1fccd7d8216c5785ac562ff41e2dcfdf5785ac562ff41e2dcf",  
    "transactionIndex": "0x0", // 0  
    "address": "0x16c5785ac562ff41e2dcfdf829c5a142f1fccd7d",  
  
    "data": "0x0000000000000000000000000000000000000000000000000000000000000000",  
    "topics":  
    ["0x59eb90bc63057b6515673c3ecf9438e5058bca0f92585014eced636878c9a5"]  
  }, {  
    ...  
  }]  
}
```

6.6 eth_getFilterLogs

API 描述

给定 id 匹配的日志列表。

API 调用

POST/api

参数说明

过滤器 id

API 返回结果

与 eth_getFilterChanges 返回值 (4.5) 相同。

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getFilterLogs","params":["0x16"],"id":739}'
```

返回结果如下：

与 eth_getFilterChanges 返回值 (4.5) 相同。

6.7 eth_getLogs

API 描述

获取日志的集合。

API 调用

POST/api

参数说明

fromBlock	(default: "latest") 区块号，或 "latest"-最新产生的区块或 "pending", "earliest"-还没有被打包的交易
toBlock	(default: "latest") 区块号，或 "latest"-最新产生的区块或 "pending", "earliest"-还没有被打包的交易
address	合约地址或者开始搜寻日志的一系列地址
topics	检索的条件
Blockhash	是一种新的滤波选项，它限制了使用 32 字节散列返回单块的日志

API 返回结果

与 eth_getFilterChanges 返回值 (4.5) 相同。

示例代码

参数说明

string	数据库名字
string	key 名称
string	存储的字符串

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
Bool	是否成功

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"db_putString","params":["testDB","myKey","myString"],  
"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc":"2.0",  
  "result": true  
}
```

7.2 db_getString

API 描述

获取本地数据库中的字符串。

API 调用

POST/api

参数说明

string	数据库名字
string	key 名称

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

string

查询的结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"db_getString","params":["testDB","myKey"],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc":"2.0",  
  "result": "myString"  
}
```

7.3 db_putHex

API 描述

存储本地数据库中的二进制数据。

API 调用

POST/api

参数说明

string	数据库名字
string	key 名称
data	序列化数据

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
bool	已经序列化返回 true

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"db_putHex","params":["testDB","myKey",  
"0x68656c6c6620776f726c64"],"id":739}'
```

返回结果如下：

```
{  
  "id":739,  
  "jsonrpc":"2.0",
```

```
"result": true
}
```

7.4 db_getHex

API 描述

获取本地数据库中的二进制数据。

API 调用

POST/api

参数说明

string	数据库名字
string	key 名称

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"db_getHex","params":["testDB","myKey"],"id":739}'
```

返回结果如下：

```
{
  "id":739,
  "jsonrpc":"2.0",
  "result": "0x68656c6c6f20776f726c64"
}
```

8 共识和奖惩

eth_getLocalMinerRewardAddress	获取本节点奖励地址
eth_getTotalAmountOfHypothecationWithAddress	查询相关地址质押的全部金额
eth_getNodeActivityInfoForAddress	查询给定地址的活跃度信息
eth_getHypothecationSuccessfulAddress	获取质押成功的地址
eth_getAmountOfHypothecation	查询当前进入共识委员会需要的质押金为多少
eth_getCurrentQuorumSize	查询共识委员会数量

eth_getConsensusQuorum	查询当前共识委员会成员
eth_getAddressIsInQuorum	查询给定地址是否在共识委员会中
eth_getNodeStatus	查询给定地址的节点状态。
eth_getCurrentCommitHeight	查询当前 commit 块的高度
eth_getNodeDisqualifiedReason	查询节点被剔除共识委员会原因
eth_getNodeUselessActiveInfoByAddress	查询给定地址的节点的无效活跃度信息(主要用于尚未加入到共识委员会但是已经加入到网络当中的节点的活跃度情况)
eth_checkNodeDetail	查询给定高度下的节点活跃度详情。

8.1 eth_getLocalMinerRewardAddress

API 描述

获取本节点奖励地址。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getLocalMinerRewardAddress","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"0x7efe6e4aec16d0157c319c1f1a785aec81c7ae75"}
```

8.2 eth_getTotalAmountOfHypothecationWithAddress

API 描述

查询相关地址质押的全部金额。

API 调用

POST/api

参数说明

20Byte 查询地址

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getTotalAmountOfHypothecationWithAddress",
"params":["0xE05e11D6272Af1A13f184326dadF4079Cfa8c31d"],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"3000000000000000"}
```

8.3 eth_getNodeActivityInfoForAddress

API 描述

查询给定地址的活跃度信息。

API 调用

POST/api

参数说明

20Byte 查询地址

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_getNodeActivityInfoForAddress","params":
["0xE05e11D6272Af1A13f184326dadF4079Cfa8c31d"],"id":739}'
```

返回结果如下：

```
{"jsonrpc": "2.0", "id": 739, "result": "30"}
```

8.4 eth_getHypothecationSuccessfulAddress

API 描述

获取质押成功的地址。

API 调用

POST/api

参数说明

20Byte 查询地址

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":  
"eth_getHypothecationSuccessfulAddress","params":[],"id":739}'
```

返回结果如下：

```
{  
  "jsonrpc": "2.0",  
  "id": 739,  
  "result":  
  ["0x41b84E81Ba73ab6910ac9b2038d6F18C15a05367","0x0297545eCb2686062fF91D1  
87bf21CDdE63ca28B"]  
}
```

8.5 eth_getAmountofHypothecation

API 描述

查询当前进入共识委员会需要的质押金为多少。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getAmountofHypothecation","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"3000000000000"}
```

8.6 eth_getCurrentQuorumSize

API 描述

查询共识委员会数量。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getCurrentQuorumSize","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"30"}
```

8.7 eth_getConsensusQuorum

API 描述

查询当前共识委员会成员。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getConsensusQuorum","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":["0xb28098Faa03770F37036C6b5b89d84733edf7EA  
9"]}
```

8.8 eth_getAddressIsInQuorum

API 描述

查询给定地址是否在共识委员会中。

API 调用

POST/api

参数说明

20Byte	查询地址
--------	------

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getAddressIsInQuorum","params":  
["0xb28098Faa03770F37036C6b5b89d84733edf7EA9"],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":true}
```

8.9 eth_getNodeStatus

API 描述

查询给定地址的节点状态。

API 调用

POST/api

参数说明

20Byte 查询地址

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
int	<div>1 没质押也不在线</div> <div>2 质押了但不在线</div> <div>3 质押了但是活跃度不够</div> <div>4 质押了活跃度也够了只是还没生效</div> <div>5 在共识委员会中</div> <div>6 已退出共识委员会可以退出质押</div> <div>7 需要被惩罚</div> <div>8 已被惩罚</div>

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getNodeStatus","params":  
["0xb28098Faa03770F37036C6b5b89d84733edf7EA9"],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":1}
```

8.10 eth_getCurrentCommitHeight

API 描述

查询当前 commit 块的高度。

API 调用

POST/api

参数说明

无。

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getCurrentCommitHeight","params":[],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":105}
```

8.11 eth_getNodeDisqualifiedReason

API 描述

查询节点被剔除共识委员会原因。

API 调用

POST/api

参数说明

20Byte	查询地址
--------	------

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果
	String 节点被剔除原因
	Activity is not enough to be removed.Should hypothecation again

活跃度不足，需重新质押
Exit on his own initiative.Should hypothecation again
主动退出，需重新质押
Multiple signatures.Should hypothecation again
多签剔除，需重新质押
Exception storage penalty.Should not hypothecation again
异常存储剔除，无需重新质押

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":"eth_getNodeDisqualifiedReason",  
"params":["0xE05e11D6272Af1A13f184326dadF4079Cfa8c31d"],"id":739}'
```

返回结果如下：

```
{"jsonrpc":"2.0","id":739,"result":"Activity is not enough to be removed.Should  
hypothecation again"}
```

8.12 eth_getNodeUselessActiveInfoByAddress

API 描述

查询给定地址的节点的无效活跃度信息(主要用于尚未加入到共识委员会但是已经加入到网络当中的节点的活跃度情况)。

API 调用

POST/api

参数说明

20byte 查询地址

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data  
'{"jsonrpc":"2.0","method":  
"eth_getNodeUselessActiveInfoByAddress","params":["0x44fee6160987f550c35c126d7  
801af444b10264e8dfa8f53038e3d7a3cd2049a"],"id":739}'
```

返回结果如下：

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result": 13
}
```

8.13 eth_checkNodeDetail

API 描述

查询给定高度下的节点活跃度详情。

API 调用

POST/api

参数说明

int64 区块号

API 返回结果

返回结果为一个 JSON 对象，字段如下：

jsonrpc	RPC 版本号，2.0
id	RPC 请求编号
result	调用结果

示例代码

以 curl 为例的调用代码如下：

```
$ curl -H "Content-Type:application/json" -X POST --data
'{"jsonrpc":"2.0","method":"eth_checkNodeDetail",
  "params":["0x20"],"id":739}'
```

返回结果如下：

```
{
  "jsonrpc": "2.0",
  "id": 739,
  "result": {
    "0x86082FA9d3c14D00a8627aF13CfA893e80B39101": {
      "address": "0x86082fa9d3c14d00a8627af13cfa893e80b39101",
      "num_assertions_accepted": 31,
      "num_assertions_total": 31,
      "num_blocks_accepted": 0,
      "num_blocks_accepted_total": 0,
      "num_blocks_failed": 0,
      "num_blocks_failed_total": 0,
      "activeness_index": 31,
      "contribution_index": 620,
      "capabilities_index": 100,
      "qualification_index": 16260,
      "prequalified_at": 0,
      "qualified_at": 0,
      "disqualified_at": 0,
      "disqualified_reason": "",
      "can_be_master": 0,
      "PunishedHeight": 0,
      "useless_assertions": 31
    }
  }
}
```