

一、后端接口

1. 返回当前交易总数和区块高度

参数

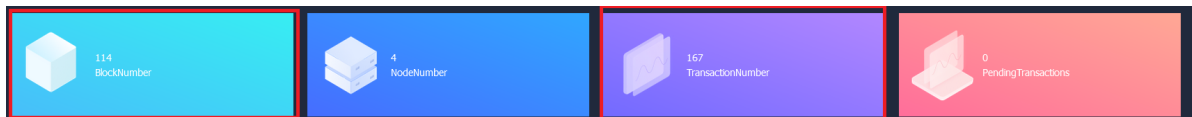
- `groupID`: `unsigned int` - 群组ID(固定为1)

```
1 Request
2 {"jsonrpc": "2.0", "method": "getTotalTransactionCount", "params": [1], "id": 1}
```

返回值

- `object`: - 当前交易总数和区块高度信息，其字段如下：
 - `blockNumber`: `string` - 区块高度
 - `failedTxSum`: `string` - 失败的交易总数
 - `txSum`: `string` - 交易总数

```
1 result
2 {
3     "id": 1,
4     "jsonrpc": "2.0",
5     "result": {
6         "blockNumber": "0x1",      //区块总数
7         "failedTxSum": "0x0",     //失败交易总数
8         "txSum": "0x1"           //交易总数
9     }
10 }
```



2. 返回待打包的交易数量

参数

- `groupID`: `unsigned int` - 群组ID(固定为1)

```
1 request
2 {"jsonrpc": "2.0", "method": "getPendingTxSize", "params": [1], "id": 1}
```

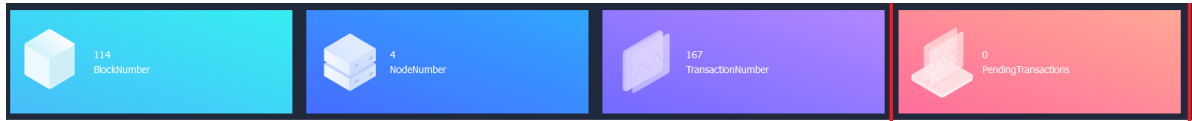
返回值

- `string`: - 待打包的交易数量

```

1 result
2 {
3   "id": 1,
4   "jsonrpc": "2.0",
5   "result": "0x1"    //待打包的交易数量，十六进制字符串
6 }

```



3. 返回当前群组内节点数量

参数

- `groupID`: `unsigned int` - 群组ID(固定为1)

```

1 request
2 {"jsonrpc": "2.0", "method": "getGroupPeers", "params": [1], "id": 1}

```

返回值

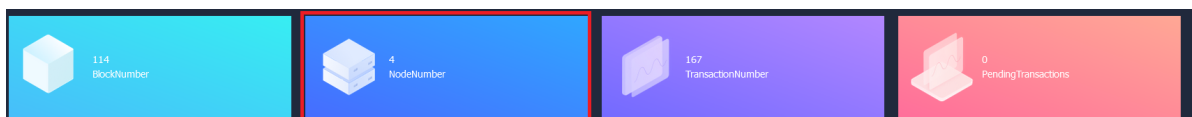
- `array` - 共识节点和观察节点的ID列表

```

1 result
2 {
3   "id": 1,
4   "jsonrpc": "2.0",
5   "result": [
6
7     "0c0bbd25152d40969d3d3cee3431fa28287e07cfff2330df3258782d3008b876d146ddab97e
8     ab42796495bfbb281591febc2a0069dcc7dfe88c8831801c5b5801",
9
10    "037c255c06161711b6234b8c0960a6979ef039374ccc8b723afea2107cba3432dbbc837a71
11    4b7da20111f74d5a24e91925c773a72158fa066f586055379a1772",
12
13    "622af37b2bd29c60ae8f15d467b67c0a7fe5eb3e5c63fdc27a0ee8066707a25afa3aa0eb5a
14    3b802d3a8e5e26de9d5af33806664554241a3de9385d3b448bcd73",
15
16    "10b3a2d4b775ec7f3c2c9e8dc97fa52beb8caab9c34d026db9b95a72ac1d1c1ad551c67c2b
17    7fdc34177857eada75836e69016d1f356c676a6e8b15c45fc9bc34"
18  ]
19 }

```

计算result数组的大小即为当前群组内节点数量



4. 根据区块编号返回单个区块信息

参数

- `groupId: unsigned int` - 群组ID(固定为1)
- `blockNumber: string` - 区块高度(十进制字符串或0x开头的十六进制字符串)
- `includeTransactions: bool` - 包含交易标志(true显示交易详细信息, false仅显示交易的hash)

返回值:

`object` - 区块信息, 字段如下:

- `dbHash: string` - 记录交易数据变更的哈希
- `extraData: array` - 附加数据
- `gasLimit: string` - 区块中允许的gas最大值
- `gasUsed: string` - 区块中所有交易消耗的gas
- `hash: string` - 区块哈希
- `logsBloom: string` - log的布隆过滤器值
- `number: string` - 区块高度
- `parentHash: string` - 父区块哈希
- `receiptsRoot: string` - 区块内所有交易回执的merkle根
- `sealer: string` - 共识节点序号
- `sealerList: array` - 共识节点列表
- `signatureList: string` - PBFT共识的签名列表
- `stateRoot: string` - 状态根哈希
- `timestamp: string` - 时间戳, 单位毫秒
- `transactions: array` - 交易列表, 当 `includeTransactions` 为 `false` 时, 显示交易的哈希。
当 `includeTransactions` 为 `true` 时, 显示交易详细信息 (详细字段见[getTransactionByHash](#))
- `transactionsRoot: string` - 区块内所有交易的merkle根

```
1 request
2 {
3     "jsonrpc": "2.0",
4     "method": "getBlockByNumber",
5     "params": [1, "0x2", true], // "0x2"为十六进制字符串格式区块编号, true表示获取
    区块的详细信息
6     "id": 1
7 }
```

```
1 result:
2 params参数为false:
3 {
4     "id": 1,
5     "jsonrpc": "2.0",
6     "result": {
7         "dbHash":
8         "0x0000000000000000000000000000000000000000000000000000000000000000",
9         "extraData": [],
10        "gasLimit": "0x0",
11        "gasUsed": "0x0",
12        "hash":
13        "0xfa639d1454362a8cdfcab1ca1948a5defaf7048b28f67e80780ab1e24e8f8c59",
```

[illegible]

```

1 result:
2 params参数为true:
3 {
4   "id": 1,
5   "jsonrpc": "2.0",
6   "result": {
7     "dbHash":
8     "0x0000000000000000000000000000000000000000000000000000000000000000",
9     "extraData": [],
10    "gasLimit": "0x0",
11    "gasUsed": "0x0",
12    "hash":
13    "0xfa639d1454362a8cdfcab1ca1948a5defaf7048b28f67e80780ab1e24e8f8c59",
14    "logsBloom":
15    "0x0000000000000000000000000000000000000000000000000000000000000000
16    0000000000000000000000000000000000000000000000000000000000000000
17    0000000000000000000000000000000000000000000000000000000000000000
18    0000000000000000000000000000000000000000000000000000000000000000
19    0000000000000000000000000000000000000000000000000000000000000000
20    0000000000000000000000000000000000000000000000000000000000000000
21    0000000000000000000000000000000000000000000000000000000000000000
22    0000000000000000000000000000000000000000000000000000000000000000",
23    "number": "0x1",
24    "parentHash":
25    "0x249f59e00beac8424a7821c4750fdd70c128f4ce795afbab53f345e9fce95d1a",

```

[illegible]

```

49         "v": "0x0"
50     },
51     "to": "0x8c17cf316c1063ab6c89df875e96c9f0f5b2f744",
52     "transactionIndex": "0x0",
53     "value": "0x0"
54 }
55 ],
56 "transactionsRoot":
    "0xb880b08df3b43a9ffc334d7a526522b33e004ef95403d61d76454b6085b9b2f1"
57 }
58 }

```

首页显示最新六个区块的部分信息(区块编号 `number` , 时间戳 `timestamp` , 出块者hash)

时间戳有相应的计算js代码;

出块者hash的获取需要结合 `sealer` 字段和 `sealerList` 字段: `sealerList` 指出了当前全部节点的hash, 且是固定顺序, `sealer` 以十六进制字符串格式指出当前出块者是 `sealerList` 的哪一个(0x0指的是 `sealerList` 数组的第一个, 0x1指的是第二个)

区块	More
Block 114 2023-01-06 19:50:08	Blocker b48c1ae4...
Block 113 2023-01-06 19:50:07	Blocker c442d855...
Block 112 2023-01-06 19:50:00	Blocker f83e6fad5...
Block 111 2023-01-06 19:50:00	Blocker c442d855...
Block 110 2023-01-06 19:49:59	Blocker b48c1ae4...
Block 109 2023-01-06 19:49:58	Blocker 8ef4c8a06...

5. 根据区块编号返回单个区块内部所有解码后的交易信息

参数

- `groupID: unsigned int` - 群组ID(固定为1)
- `blockNumber: string` - 区块高度(十进制字符串或0x开头的十六进制字符串)
- `includeTransactions: bool` - 包含交易标志(由于解码解码需要详细的交易信息, 因此这里必须固定为true)

```

1  request
2  {
3      "jsonrpc": "2.0",
4      "method": "getBlockByNumber_sp",
5      "params": [1, "0x2", true],          //"0x2"为十六进制字符串格式区块编号, true表示获取
      区块的详细信息
6      "id": 1
7  }

```

由于存在有四种类型的消息, 因为result有四种json格式:

第一种: Store spectrum usage info on blockchain

返回值:

- `txCount`: 区块内交易总数

- `transactions`：交易列表
 - `method`：交易类型
 - `txHash`：本交易的Hash
 - `apAddr`：AP基站的Address
 - `apid`：AP基站的id
 - `apLocation`：AP基站的经纬度
 - `apChannelList`：AP基站的可用信道列表
 - `apPower`：AP基站的功率

```

1  {
2  {
3  "txCount": 1,    //此区块内所含交易数量
4  "transactions": [
5    {
6      "method": "Store spectrum usage info on blockchain",
7      "txHash":
8      "0xba42461b3d9890a6b78aa3268e55f9219a9f2ff602a27c7368c5fd2b5b2c5bce",
9      "apAddr": "3632303030303039325f35353331323932393738",
10     "apid": "620000092_55",
11     "apLocation": "Lat:113.740030, Lng:40.385655",
12     "apChannelList": "[47]",
13     "apPower": 23
14   }
15 ]
16 }
```

第二种：Buyer:Purchase Request

返回值：

- `txCount`：区块内交易总数
- `transactions`：交易列表
 - `method`：交易类型
 - `txHash`：本交易的Hash
 - `buyerAPAddr`：信道购买方AP的Address
 - `buyerAPid`：信道购买方AP的id
 - `pricesList`：购买方AP认定的潜在出售的AP的Address和id

```

1  {
2    "txCount": 1,
3    "transactions": [
4      {
5        "method": "Buyer:Purchase Request",
6        "txHash":
7          "0xe6c5221cc97df6ff5764c7de5fa71afd6f486c42da7664a93613a6c1b8ee0666",
8        "buyerAPAddr": "3632303030303039315f35353730303139373933",
9        "buyerAPid": "620000091_55",
10       "pricesList": "SellerAPAddr:3632303030303039335f35353739353234373334
11       || sellerAPid:620000093_55"
12     }
13   ]
14 }

```

第三种: Seller:Willingness to sell

返回值:

- txCount: 区块内交易总数
- transactions: 交易列表
 - method: 交易类型
 - txHash: 本交易的Hash
 - buyerAPAddr: 信道购买方AP的地址
 - buyerAPid: 信道购买方AP的id
 - sellerAPAddr: 信道出售方AP的地址
 - sellerAPid: 信道出售方AP的id
 - switchChannels: 信道出售方意图出售的信道

```

1  {
2    "txCount": 1,
3    "transactions": [
4      {
5        "method": "Seller:Willingness to sell",
6        "txHash":
7          "0x1b338313b4291276e29149f392828c0009519cc8cb42a711c1824763b5f8d9f4",
8        "buyerAPAddr": "3632303030303039315f35353730303139373933",
9        "buyerAPid": "620000091_55",
10       "sellerAPAddr": "3632303030303039335f35353739353234373334",
11       "sellerAPid": "620000093_55",
12       "switchChannels": [
13         38
14       ]
15     }
16   ]
17 }

```

第四种: Trading confirm

返回值:

- txCount: 区块内交易总数

- `transactions`: 交易列表
 - `method`: 交易类型
 - `txHash`: 本交易的Hash
 - `buyerAPAddr`: 信道购买方AP的地址
 - `buyerAPid`: 信道购买方AP的id
 - `sellerAPAddr`: 信道出售方AP的地址
 - `sellerAPid`: 信道出售方AP的id
 - `switchChannels`: 信道出售方确认被出售的信道

```

1  {
2    "txCount": 1,
3    "transactions": [
4      {
5        "method": "Trading confirm",
6        "txHash":
7        "0xcc5967d8ac0c18b9e7ab2d8f9bcd6a7b561c6ebd75f97aa334ce5c125bc43c36",
8        "buyerAPAddr": "3632303030303039315f35353730303139373933",
9        "buyerAPid": "62000091_55",
10       "sellerAPAddr": "3632303030303039335f35353739353234373334",
11       "sellerAPid": "62000093_55",
12       "switchChannels": [
13         38
14       ]
15     }
16   ]
17 }
```

第五种: Update trading results to Coexistence manager

返回值:

- `txCount`: 区块链内交易总数
- `transactions`: 交易列表
 - `method`: 交易类型
 - `txHash`: 本交易的Hash
 - `apAddr`: 需要进行信息更新的AP基站的Address

```

1  {
2    "txCount": 4,
3    "transactions": [
4      {
5        "method": "Update trading results to Coexistence manager",
6        "txHash":
7        "0x65f6ea22466c3a3d8b0fadfb5cc0c6dae419345350cf10a00a6c29871c708d8e",
8        "apAddr": "3632303030303039315f35353730303139373933"
9      },
10     {
11       "method": "Update trading results to Coexistence manager",
12       "txHash":
13       "0xc6affdf7056142c359428caffa2b0a096c421ae0be6fa2d542b73f43ab14c887",
14       "apAddr": "3632303030303039315f35353730303139373933"
15     },
16   ]
17 }
```

```

14      {
15          "method": "Update trading results to Coexistence manager",
16          "txHash":
17          "0x4b77Fc37b48ed4111b225adf77c0b495b5d7816feab2219053c90292643718b5",
18      },
19      {
20          "method": "Update trading results to Coexistence manager",
21          "txHash":
22          "0x1764cb6d29884df9173127d0dd9677eca7560e30c976088c2b9d4d7296e5042b",
23          "apAddr": "3632303030303039315f35353730303139373933"
24      }
25  ]
26  }

```

点击区块可以查看具体的解码的交易信息(结合使用4/5json消息)

红框中的信息即为5号json消息回复中解码的区块内交易，其余皆为使用4号json消息获取的信息。

transactionHash	blockHeight	from	to
0x41348b5e0e7fa3f05a016c6c6a6709b0cc662811c614770449a9d993ec3ec9	114	0x41348b5e0e7fa3f05a016c6c6a6709b0cc662811c614770449a9d993ec3ec9	0x8cd46b96a29f6e4b42ae194dfeff9a54aef...

transactionInfo transactionReceipt

blockHash: 0x534cddf0ec94bcb1af81e8d0ce25f4ca8977b46dfac34d240c7aa1404d383
 blockNumber: 114
 gas: 0x1c9c380
 from: 0xae1c26b9d2b224ebdef2ae649c8e5c9459f456
 transactionIndex: 0x0
 to: 0x8cd46b96a29f6e4b42ae194dfeff9a54aef6fd
 nonceRaw:
 value: 0x0
 hash: 0x41348b5e0e7fa3f05a016c6c6a6709b0cc662811c614770449a9d993ec3ec9
 gasPrice: 0x1c9c380
 input:

function	methodId	Buyer:Purchase Request	txHash	buyerAPAddr	buyerAPId	pricesList
data	method					
Buyer:Purchase Request			0x41348b5e0e7fa3f05a016c6c6a6709b0cc662811c614770449a9d993ec3ec9	3532303430303231345f	520400214_55	SellerAPAddr:36323030303039339f55353433353839303638 SellerAPId:620000093_55

6. 获取现有的所有区块

参数

无

```
1 | {"jsonrpc": "2.0", "method": "getBlockByNumber_all", "params": [], "id": 1}
```

返回值:

按顺返回所有区块，每个区块按照4.中的 true 方式json消息返回的回复

在blockInfo页面分页显示所有区块:

BlockHeight	TransactionCount	ConsensusNodeHash	CreateTime
Block114	1	b48c1ae6487241cb90b92562156ba07a3fffa0462b7b57c0673ae44972979886806eaf253b1dd5666170d52571de39361c71d46159fe...	2023-01-06 19:50:08
Block113	1	c442d855675e839ad69af11023e8411ecff9ca0e414c9d4a4229e33cbee190896e0cb0182110ca79747b6c0ee54f2a1feb2fb20b0e5...	2023-01-06 19:50:07
Block112	4	f83e6fad51af6dce000bc7a812b18d9672a16d4c17ba5318c734e8239c678e28c37546979bc24574b2209acd0520f077a55a630620ce6...	2023-01-06 19:50:00
Block111	4	c442d855675e839ad69af11023e8411ecff9ca0e414c9d4a4229e33cbee190896e0cb0182110ca79747b6c0ee54f2a1feb2fb20b0e5...	2023-01-06 19:50:00
Block110	1	b48c1ae6487241cb90b92562156ba07a3fffa0462b7b57c0673ae44972979886806eaf253b1dd5666170d52571de39361c71d46159fe...	2023-01-06 19:49:59
Block109	1	8ef4c8a06a3e172e14975768936a72ddcf3630b7ff59134acc852ad5ca889f431b783cfb726c355500c597cce254b9b9d8b7b4c2736b8e6...	2023-01-06 19:49:58
Block108	1	b48c1ae6487241cb90b92562156ba07a3fffa0462b7b57c0673ae44972979886806eaf253b1dd5666170d52571de39361c71d46159fe...	2023-01-06 19:49:58
Block107	1	c442d855675e839ad69af11023e8411ecff9ca0e414c9d4a4229e33cbee190896e0cb0182110ca79747b6c0ee54f2a1feb2fb20b0e5...	2023-01-06 19:49:55
Block106	1	b48c1ae6487241cb90b92562156ba07a3fffa0462b7b57c0673ae44972979886806eaf253b1dd5666170d52571de39361c71d46159fe...	2023-01-06 19:49:55
Block105	1	8ef4c8a06a3e172e14975768936a72ddcf3630b7ff59134acc852ad5ca889f431b783cfb726c355500c597cce254b9b9d8b7b4c2736b8e6...	2023-01-06 19:49:54

7. 根据用户输入的区块编号获取区块

①捕获用户输入的十进制数值型区块编号 ②将其转化为十六进制字符串 ③分别使用json消息4与json消息5获取区块的信息和具体解码的交易信息。

Please input Block Number or Tx Hash

区块

Block 114

Blocker b48c1ae4...

text.total 113 text.tao

inputText.blockInput

BlockHeight

TransactionCount

输入区块编号后，跳转至：

transactionHash

blockHeight

from

to

0x41348b5e0e7ffa3f05a016c6c6af6709fbc662811c614770449a9d993ec3ec9

114

0xfefc726b9d2b224ebdef2ae649c8e5c9459ff...

0x8cd46b96a29f6e4b42aef194dfeff9a54afe...

transactionInfo

transactionReceipt

blockHash: 0xe534cdf0ec94bcb1af81e8d0ce25ffa088978b46dfac34240c7aa41404d383

blockNumber: 114

gas: 0xc1c9c380

from: 0xfefc726b9d2b224ebdef2ae649c8e5c9459ff456

transactionIndex: 0x0

to: 0x8cd46b96a29f6e4b42aef194dfeff9a54afe8fd

nonceRaw:

value: 0x0

hash: 0x41348b5e0e7ffa3f05a016c6c6af6709fbc662811c614770449a9d993ec3ec9

gasPrice: 0xc1c9c380

input:

function

methodId

data

Buyer:Purchase Request

method

txHash

buyerAPAddr

buyerAPId

pricesList

Buyer:Purchase Request

0x41348b5e0e7ffa3f05a016c6c6af6709fbc662811c614770449a9d993ec3ec9

3532303430303231345f

520400214_55

SellerAPAddr:3632303030303039335f35353433353839303638 || SellerAPId:620000093_55