

BHP 算力公链

API 接口

V1.0rc

目录

1. 概述.....	6
1.1. 命令列表.....	7
1.2. GET 请求示例.....	8
1.3. POST 请求示例.....	8
1.4. 测试工具.....	9
2. 接口说明.....	9
2.1. 创建一个新地址.....	9
2.1.1. 方法.....	9
2.1.2. 作用.....	9
2.1.3. 备注.....	10
2.1.4. 调用示例.....	10
2.2. 查询钱包内所有地址.....	10
2.2.1. 方法.....	10
2.2.2. 作用.....	10
2.2.3. 备注.....	10
2.2.4. 调用示例.....	10
2.3. 导出私钥.....	11
2.3.1. 方法.....	11
2.3.2. 作用.....	11
2.3.3. 备注.....	11
2.3.4. 参数说明.....	12
2.3.5. 调用示例.....	12
2.4. 地址验证.....	12
2.4.1. 方法.....	12
2.4.2. 作用.....	12
2.4.3. 参数说明.....	12
2.4.4. 调用示例.....	12
2.5. 查询账户资产信息.....	13
2.5.1. 方法.....	13
2.5.2. 作用.....	13
2.5.3. 参数说明.....	13
2.5.4. 调用示例.....	13
2.6. 查询资产信息.....	14
2.6.1. 方法.....	14
2.6.2. 作用.....	14
2.6.3. 参数说明.....	14
2.6.4. 调用示例.....	15
2.7. 查询余额信息.....	16
2.7.1. 方法.....	16
2.7.2. 作用.....	16
2.7.3. 备注.....	16
2.7.4. 参数说明.....	16
2.7.5. 调用示例.....	16

2.8. 获取主链区块数量.....	17
2.8.1. 方法.....	17
2.8.2. 作用.....	17
2.8.3. 调用示例.....	17
2.9. 获取主链最大高度区块散列.....	18
2.9.1. 方法.....	18
2.9.2. 作用.....	18
2.9.3. 调用示例.....	18
2.10. 获取指定区块散列.....	18
2.10.1. 方法.....	18
2.10.2. 作用.....	18
2.10.3. 参数说明.....	19
2.10.4. 调用示例.....	19
2.11. 获取指定区块信息.....	19
2.11.1. 方法.....	19
2.11.2. 作用.....	19
2.11.3. 参数说明.....	19
2.11.4. 调用示例.....	20
2.12. 获取连接的节点列表.....	22
2.12.1. 方法.....	22
2.12.2. 作用.....	22
2.12.3. 调用示例.....	22
2.13. 获取当前节点连接数.....	24
2.13.1. 方法.....	24
2.13.2. 作用.....	24
2.13.3. 调用示例.....	24
2.14. 查询节点版本信息.....	24
2.14.1. 方法.....	24
2.14.2. 作用.....	24
2.14.3. 调用示例.....	25
2.15. 获取共识节点信息及投票情况.....	25
2.15.1. 方法.....	25
2.15.2. 作用.....	25
2.15.3. 调用示例.....	25
2.16. 获取合约日志.....	26
2.16.1. 方法.....	26
2.16.2. 作用.....	26
2.16.3. 备注.....	27
2.16.4. 参数说明.....	27
2.16.5. 调用示例.....	27
2.17. 查询合约信息.....	28
2.17.1. 方法.....	28
2.17.2. 作用.....	28
2.17.3. 参数说明.....	28

2.17.4. 调用示例.....	28
2.18. 以散列值调用智能合约.....	30
2.18.1. 方法.....	30
2.18.2. 备注.....	30
2.18.3. 参数说明.....	30
2.18.4. 调用示例.....	31
2.19. 广播交易.....	33
2.19.1. 方法.....	33
2.19.2. 作用.....	33
2.19.3. 参数说明.....	33
2.19.4. 调用示例.....	33
2.20. 获取交易列表.....	34
2.20.1. 方法.....	34
2.20.2. 作用.....	34
2.20.3. 调用示例.....	34
2.21. 获取交易信息.....	34
2.21.1. 方法.....	34
2.21.2. 作用.....	35
2.21.3. 参数说明.....	35
2.21.4. 调用示例.....	35
2.22. 获取未花费交易输出信息.....	37
2.22.1. 方法.....	37
2.22.2. 作用.....	37
2.22.3. 参数说明.....	37
2.22.4. 调用示例.....	37
2.23. 获取指定的存储值.....	38
2.23.1. 方法.....	38
2.23.2. 作用.....	38
2.23.3. 参数说明.....	38
2.23.4. 调用示例.....	38
2.24. 获取系统手续费.....	39
2.24.1. 方法.....	39
2.24.2. 作用.....	39
2.24.3. 参数说明.....	39
2.24.4. 调用示例.....	39
2.25. 向指定地址转账.....	39
2.25.1. 方法.....	39
2.25.2. 备注.....	40
2.25.3. 参数说明.....	40
2.25.4. 调用示例.....	40
2.26. 批量转账.....	42
2.26.1. 方法.....	42
2.26.2. 作用.....	42
2.26.3. 备注.....	42

2.26.4. 参数说明.....	42
2.26.5. 调用示例.....	42

1. 概述

每个 BHP 节点都可选的提供了一套 API 接口，用于从节点获取区块链数据，使得开发区块链应用变得十分方便。接口通过 JSON-RPC 的方式提供，底层使用 HTTP/HTTPS 协议进行通讯。要启动一个提供 RPC 服务的节点，可运行以下命令：

```
dotnet bhp-cli.dll /rpc
```

若要通过 HTTPS 的方式访问 RPC 服务器，需要在启动节点前修改配置文件 config.json，并设置域名、证书和密码：

```
{
  "ApplicationConfiguration": {
    "Paths": {
      "Chain": "Chain_{0}",
      "Index": "Index_{0}"
    },
    "P2P": {
      "Port": 10665, //本地 P2P 监听端口
      "WsPort": 10666 //本地 WebSocket 监听端口
    },
    "RPC": {
      "BindAddress": "127.0.0.1",
      "Port": 10667, //RPC 服务端口
      "SslCert": "", //HTTPS 证书
      "SslCertPassword": ""
    },
    "UnlockWallet": {
      "Path": "C:/BHP/20181018/test-cli/bhp-cli-4/4.json", //钱包路径
      "Password": "123456", //钱包密码
      "StartConsensus": false, //普通节点不需要启动
      "IsActive": true //打开钱包
    }
  }
}
```

JSON-RPC 服务器启动后，会监听 TCP 端口，默认端口如下。

	主网（Main Net）	测试网（Test Net）
JSON-RPC HTTP	20557	10557

1.1. 命令列表

方法	参数	说明	备注
dumpprivkey	<address>	导出指定地址的私钥	需要打开钱包
getaccountstate	<address>	根据账户地址，查询账户资产信息	
getapplicationlog	<txid> [verbose=0]	根据指定的交易 ID 获取合约日志	需要启动日志记录
getassetstate	<asset_id>	根据指定的资产编号，查询资产信息	
getbalance	<asset_id>	根据指定的资产编号，返回钱包中对应资产的余额信息	需要打开钱包
getbestblockhash		获取主链中高度最大的区块的散列	
getblock	<hash> [verbose=0]	根据指定的散列值，返回对应的区块信息	
getblock	<index> [verbose=0]	根据指定的索引，返回对应的区块信息	
getblockcount		获取主链中区块的数量	
getblockhash	<index>	根据指定的索引，返回对应区块的散列值	
getblocksysfee	<index>	根据指定的索引，返回截止到该区块前的系统手续费	
getconnectioncount		获取节点当前的连接数	
getcontractstate	<script_hash>	根据合约脚本散列，查询合约信息	
getnewaddress		创建一个新的地址	需要打开钱包
getrawmempool		获取内存中未确认的交易列表	
getrawtransaction	<txid> [verbose=0]	根据指定的散列值，返回对应的交易信息	
getstorage	<script_hash> <key>	根据合约脚本散列和存储的 key，返回存储的 value	
gettxout	<txid> <n>	根据指定的散列和索引，返回对应的交易输出（零钱）信息	
getpeers		获得该节点当前已连接/未连接的节点列表	
getversion		获取查询节点的版本信息	
getvalidators		查看当前共识节点的信息	
invoke	<script_hash> <params>	使用给定的参数以散列值调用智能合约	

invokefunction	<script_hash> <operation> <params>	以指定的脚本散列值调用智能合约，传入操作及参数	
invokescript	<script>	通过虚拟机运行脚本并返回结果	
listaddress		列出当前钱包内的所有地址	需要打开钱包
sendrawtransaction	<hex>	广播交易	
sendfrom	<asset_id> <from><to> <value> [fee=0]	从指定地址，向指定地址转账	需要打开钱包
sendtoaddress	<asset_id> <address> <value> [fee=0]	向指定地址转账	需要打开钱包
sendmany	<outputs_array> [fee=0] [change_address]	批量转账命令	需要打开钱包
submitblock	<hex>	提交新的区块	需要成为共识节点
validateaddress	<address>	验证地址是否是正确的 BHP 地址	

1.2. GET 请求示例

一次典型的 JSON-RPC GET 请求格式如下：

下面以获取主链中区块的数量方法为例。

请求 URL：

[http://somewebsite.com:20557?jsonrpc=2.0&method=getblockcount¶ms=\[\]&id=1](http://somewebsite.com:20557?jsonrpc=2.0&method=getblockcount¶ms=[]&id=1)

发送请求后，将会得到如下的响应：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": 909129
}
```

1.3. POST 请求示例

一次典型的 JSON-RPC Post 请求的格式如下：

下面以获取主链中区块的数量方法为例。

请求 URL: exp.bhpa.io:20557

请求 Body:

```
{
  "jsonrpc": "2.0",
  "method": "getblockcount",
  "params": [],
}
```



```
"id": 1
}
```

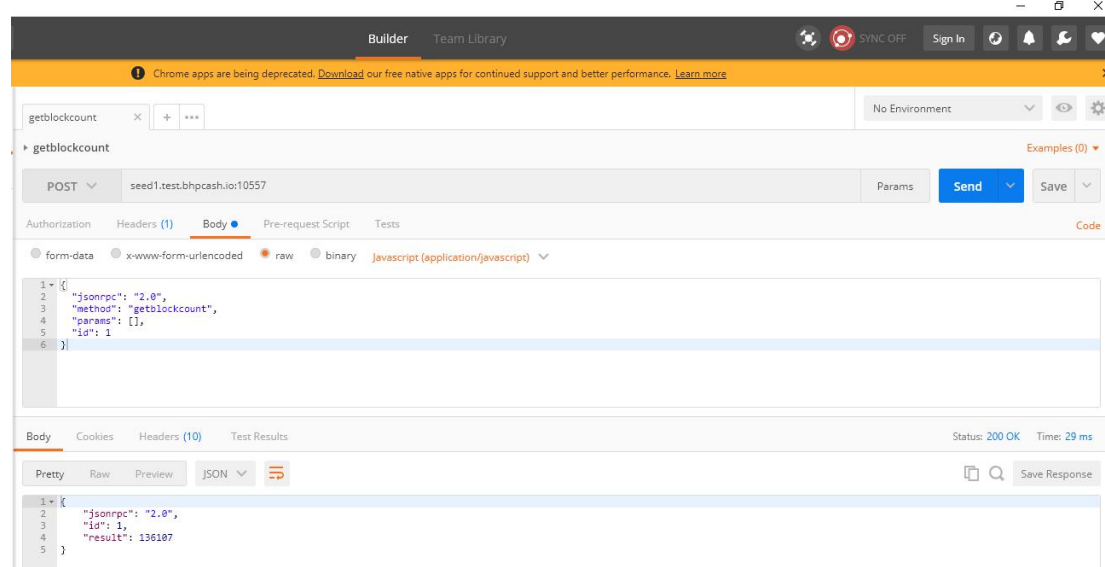
发送请求后，将会得到如下的响应：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": 909122
}
```

注：当使用离线同步包同步区块时，程序可能无法响应 API 请求，建议将区块同步到最新高度后再使用 API，否则返回的结果可能不是最新的。

1.4. 测试工具

你可以用 Chrome 扩展程序中的 Postman 来方便地进行测试（安装 Chrome 扩展程序需要科学上网），下面是测试截图：



2. 接口说明

2.1. 创建一个新地址

2.1.1. 方法

getnewaddress

2.1.2. 作用

创建一个新的地址

2.1.3. 备注

执行此命令前需要在 Bhp-CLI 节点中打开钱包

2.1.4. 调用示例

请求正文：

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

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": "AVHcdW3FGKbPWGHNhkPjgVgi4GGndiCxdo"
}
```

响应说明：返回新创建的地址

2.2. 查询钱包内所有地址

2.2.1. 方法

listaddress

2.2.2. 作用

列出当前钱包内的所有地址

2.2.3. 备注

执行此命令前需要在 Bhp-CLI 节点中打开钱包

2.2.4. 调用示例

请求正文：

```
{
```

```
    "jsonrpc": "2.0",
    "method": "listaddress",
    "params": [],
    "id": 1
  }
  响应正文:
  {
    "jsonrpc": "2.0",
    "id": 1,
    "result": [
      {
        "address": "ASL3KCvJasA7QzpYGePp25pWuQCj4dd9Sy",
        "haskey": true,
        "label": null,
        "watchonly": false
      },
      {
        "address": "AV2Ai7PXcNbjTSeKgWqsDEjLaEAJZpytru",
        "haskey": true,
        "label": null,
        "watchonly": false
      }
    ]
  }
```

响应说明:

- 1) address: 钱包内的地址
- 2) watchonly: 该地址是否为监视地址

2.3. 导出私钥

2.3.1. 方法

dumpprivkey

2.3.2. 作用

导出指定地址的私钥

2.3.3. 备注

执行此命令前需要在打开钱包

2.3.4. 参数说明

address: 要导出私钥的地址，该地址需为标准地址。

2.3.5. 调用示例

请求正文:

```
{
  "jsonrpc": "2.0",
  "method": "dumpprivkey",
  "params": ["ASMGGHQPzZqxFB2yKmzvfv82jtKVnjhp1ES"],
  "id": 1
}
```

响应正文:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": "L3FdGAisCmV*****9XM65cvjYQ1"
}
```

响应说明: 返回该标准地址的私钥

2.4. 地址验证

2.4.1. 方法

validateaddress

2.4.2. 作用

验证地址是否是正确的 BHP 地址

2.4.3. 参数说明

address: 地址

2.4.4. 调用示例

请求正文:

```
{
```

```
    "jsonrpc": "2.0",
    "method": "validateaddress",
    "params": ["AQVh2pG732YvtNaxEGkQUei3YA4cvo7d2i"],
    "id": 1
  }
  响应正文:
  {
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
      "address": "AQVh2pG732YvtNaxEGkQUei3YA4cvo7d2i",
      "invalid": true
    }
  }
```

2.5. 查询账户资产信息

2.5.1. 方法

getaccountstate

2.5.2. 作用

根据账户地址，查询账户资产信息

2.5.3. 参数说明

address: 账户地址，以 A 开头的 34 位长度的字符串

2.5.4. 调用示例

```
请求正文:
{
  "jsonrpc": "2.0",
  "method": "getaccountstate",
  "params": ["AJBENSwajTzQtwyJFkiJSv7MAaaMc7DsRz"],
  "id": 1
}
响应正文:
{
  "jsonrpc": "2.0",
```

```
"id": 1,
"result": {
  "version": 0,
  "script_hash": "0x1179716da2e9523d153a35fb3ad10c561b1e5b1a",
  "frozen": false,
  "votes": [],
  "balances": [
    {
      "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
      "value": "94"
    }
  ]
}
```

响应说明：

- 1) script_hash: 合约脚本散列，在 BHP 中所有账户都是合约账户。
- 2) frozen: 该账户是否冻结。
- 3) votes: 查询该地址用于投票的 BHP
- 4) balance: 该地址的资产余额。
- 5) asset: 资产 ID
- 6) value: 资产金额

2.6. 查询资产信息

2.6.1. 方法

getassetstate

2.6.2. 作用

根据指定的资产编号，查询资产信息

2.6.3. 参数说明

asset_id: 资产 ID（资产标识符），即该资产在注册时的交易 ID。

资产 ID 可以通过 CLI 命令 中的 list asset 命令查询，也可以在区块链浏览器中查询。

2.6.4. 调用示例

请求正文:

```
{
  "jsonrpc": "2.0",
  "method": "getassetstate",
  "params":
  ["0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854"],
  "id": 1
}
```

响应正文:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "version": 0,
    "id":
    "0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
    "type": "GoverningToken",
    "name": [
      {
        "lang": "zh-CN",
        "name": "算力币"
      },
      {
        "lang": "en",
        "name": "BHP"
      }
    ],
    "amount": "100000000",
    "available": "100000000",
    "precision": 8,
    "owner": "00",
    "admin": "Abf2qMs1pzQb8kYk9RuxtUb9jtRKJVuBJt",
    "issuer": "Abf2qMs1pzQb8kYk9RuxtUb9jtRKJVuBJt",
    "expiration": 4000000,
    "frozen": false
  }
}
```

2.7. 查询余额信息

2.7.1. 方法

getbalance

2.7.2. 作用

根据指定的资产编号，返回钱包中对应资产的余额信息。

2.7.3. 备注

- 1) 执行此命令前需要在 Bhp-CLI 节点中打开钱包
- 2) 全局资产查询：当区块未完全同步时，返回的资产余额可能不是最新的，请确保使用该 API 时区块已经同步到最新高度。
- 3) 合约资产查询：**a.** 当未同步到发布合约的区块时，执行该 API 会报错，只有当区块同步到发布该合约资产的区块时，才会返回正确的结果。**b.** 当区块未完全同步时，返回的资产余额可能不是最新的，请确保使用该 API 时区块已经同步到最新高度。

2.7.4. 参数说明

asset_id: 资产 ID（资产标识符），如果是全局资产，此处为注册资产时的 Register Transaction 或 Publish Transaction 的交易 ID，如果是合约内部资产，此处为合约的 Script Hash。资产 ID 可以通过 CLI 命令中的 list asset 命令查询，也可以在区块链浏览器中查询。

2.7.5. 调用示例

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getbalance",
  "params":
  ["025d82f7b00a9ff1cfe709abe3c4741a105d067178e645bc3ebad9bc79af47d4"],
  "id": 1}
```

响应正文：

```
{
  "jsonrpc": "2.0",
```



```
"id": 1,  
"result": {  
  "balance": "1.01",  
  "confirmed": "1.01"  
}
```

响应说明：

- 1) **balance**: 钱包中该资产的真实余额。
- 2) **confirmed**: 钱包中该资产的已确认的金额，只有已确认的金额可以用来转账。
- 3) **balance** 和 **confirmed** 二者可能会不相等，仅在从钱包中转出一笔钱，而且有找零未确认时时，**confirmed** 值会小于 **balance**。当这笔交易确定后，二者会变得相等。

2.8. 获取主链区块数量

2.8.1. 方法

getblockcount

2.8.2. 作用

获取主链中区块的数量

2.8.3. 调用示例

请求正文：

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

响应正文：

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "result": 991991  
}
```

2.9. 获取主链最大高度区块散列

2.9.1. 方法

getbestblockhash

2.9.2. 作用

获取主链中高度最大的区块的散列

2.9.3. 调用示例

请求正文：

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

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": "0x773dd2dae4a9c9275290f89b56e67d7363ea4826dfd4fc13cc01cf73a44b0d0e"
}
```

响应说明：result 为主链中高度最大的区块的散列

2.10. 获取指定区块散列

2.10.1. 方法

getblockhash

2.10.2. 作用

根据指定的索引，返回对应区块的散列值

2.10.3. 参数说明

index: 区块索引

2.10.4. 调用示例

请求正文:

```
{
  "jsonrpc": "2.0",
  "method": "getblockhash",
  "params": [10000],
  "id": 1
}
```

响应正文:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result":
"0x4c1e879872344349067c3b1a30781eeb4f9040d3795db7922f513f6f9660b9b2"
}
```

2.11. 获取指定区块信息

2.11.1. 方法

getblock

2.11.2. 作用

- 1) 根据指定的散列值，返回对应的区块信息
- 2) 根据指定的索引，返回对应的区块信息

2.11.3. 参数说明

- 1) hash: 区块散列值（根据散列值获取时）。
- 2) index: 区块索引（区块高度） = 区块数 - 1（根据索引获取时）。
- 3) verbose: 可选参数，verbose 默认值为 0，verbose 为 0 时返回的是区块的序列化后的信息，用 16 进制字符串表示，如果从中获取详细信息需要调用 SDK 来进行反序列化。verbose 为 1 时返回的是对应区块的详细信息，用 Json 格式字符串表示。

2.11.4. 调用示例

以散列值获取为例，索引值获取相同。

示例 1 返回区块序列化后的信息：

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getblock",
  "params":
  ["773dd2dae4a9c9275290f89b56e67d7363ea4826dfd4fc13cc01cf73a44b0d0e"],
  "id": 1
}
```

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result":
  "000000002deadfa82cbc4682f5800ec72a8d8bd6afa469af5b2de83a51d28795a893222816f8081
bf1054136cca420f807f844a958b2dea482dfc99d2538ef9c77d13320f9263659d4220f00878f40bd
841c552a59e75d652b5d3827bf04c165bbe9ef95cca4bf5501fd450140b514d8562ad3badac0e097
a502a43c58e23c75029dad8ccdb3b1ce221067d73d5612950e38c7565d6b166ef62894399a6f152
c38a1bdb8c7d3715f75f20c1c7340e443f55108c5eefd99f954e06b21e97a4f0cf64dbd4e52426c27f
7046cd880d6a7b1a507131c39afa48b9cac16411d6f84ec2f0b5d9977e5f1e3ce760a127b31409b8
a52714b37a3b0970a19b4fb2669d2aa41ea85e05e68dfb03a197d505282dd53846ca58b1457504c
65759a9ceb8f84f5148dec71727e9c743e986092728174401862c08611338be8e352b9110b2bb6d
11ce0485286d857162deb417f1cb920d6727f8e6edbe1b7fce8d9b122523d5b45cfd02ab1ca002a5
8e28c8903ad764a84409dfcbda69cef1164936212e8e5d91965c8a976dc8dbcb5ea7d2f2d2f0105d
adb902924559fede016a1f76a2c7ab0ff89a6446b0c19c88375906c8b9eccb61bc1f1552102486fd1
5702c4490a26703112a5cc1d0923fd697a33406bd5a1c00e0013b09a7021024c7b7fb6c310fccf1ba
33b082519d82964ea93868d676662d4a59ad548df0e7d2102aaec38470f6aad0042c6e877cfd808
7d2676b0f516fddd362801b9bd3936399e2103b209fd4f53a7170ea4444e0cb0a6bb6a53c2bd016
926989cf85f9b0fba17a70c2103b8d9d5771d8f513aa0869b9cc8d50986403b78c6da36890638c3d
46a5adce04a2102ca0e27697b9c248f6f16e085fd0061e26f44da85b58ee835c110caa5ec3ba55421
02df48f60e8f3e01c48ff40b9b7f1310d7a8b2a193188befe1c2e3df740e89509357ae010000878f40
bd00000000"
}
```

示例 2 返回 JSON 格式：

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getblock",
```

```
    "params":
[["773dd2dae4a9c9275290f89b56e67d7363ea4826dfd4fc13cc01cf73a44b0d0e", 1],
    "id": 1
  ]
  响应正文:
  {
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
      "hash":
"0x773dd2dae4a9c9275290f89b56e67d7363ea4826dfd4fc13cc01cf73a44b0d0e",
      "size": 686,
      "version": 0,
      "previousblockhash":
"0x282293a89587d2513ae82d5baf69a4afd68b8d2ac70e80f58246bc2ca8dfea2d",
      "merkleroot":
"0xce52b4ed7fc69fe044d355bedcf55e070f5b8e661e3e4380cd513aef6224de42",
      "time": 1496721145,
      "index": 991956,
      "nonce": "2a551c84bd408f87",
      "nextconsensus": "APyEx5f4Zm4oCHwFWiSTaph1fPBxZacYVR",
      "script": {
        "invocation":
"40b514d8562ad3badac0e097a502a43c58e23c75029dad8ccdb3b1ce221067d73d5612950e38c7
565d6b166ef62894399a6f152c38a1bdb8c7d3715f75f20c1c7340e443f55108c5eefd99f954e06b2
1e97a4f0cf64dbd4e52426c27f7046cd880d6a7b1a507131c39afa48b9cac16411d6f84ec2f0b5d99
77e5f1e3ce760a127b31409b8a52714b37a3b0970a19b4fb2669d2aa41ea85e05e68dfb03a197d5
05282dd53846ca58b1457504c65759a9ceb8f84f5148dec71727e9c743e986092728174401862c0
8611338be8e352b9110b2bb6d11ce0485286d857162deb417f1cb920d6727f8e6edbe1b7fce8d9b
122523d5b45cfd02ab1ca002a58e28c8903ad764a84409dfcbda69cef1164936212e8e5d91965c8a
976dc8dbcb5ea7d2f2d2f0105dadb902924559fed016a1f76a2c7ab0ff89a6446b0c19c88375906c
8b9eccb61bc1",
        "verification":
"552102486fd15702c4490a26703112a5cc1d0923fd697a33406bd5a1c00e0013b09a7021024c7b7
fb6c310fccf1ba33b082519d82964ea93868d676662d4a59ad548df0e7d2102aaec38470f6aad004
2c6e877cfd8087d2676b0f516fddd362801b9bd3936399e2103b209fd4f53a7170ea4444e0cb0a6b
b6a53c2bd016926989cf85f9b0fba17a70c2103b8d9d5771d8f513aa0869b9cc8d50986403b78c6d
a36890638c3d46a5adce04a2102ca0e27697b9c248f6f16e085fd0061e26f44da85b58ee835c110c
aa5ec3ba5542102df48f60e8f3e01c48ff40b9b7f1310d7a8b2a193188befe1c2e3df740e89509357
ae"
      },
      "tx": [
        {
          "txid":
```

```

"0x2033d1779cef38259dc9df82a4deb258a944f807f820a4cc364105f11b08f816",
    "size": 10,
    "type": "MinerTransaction",
    "version": 0,
    "attributes": [],
    "vin": [],
    "vout": [],
    "sys_fee": "0",
    "net_fee": "0",
    "scripts": [],
    "nonce": 3175124871
  }
],
"confirmations": 20,
"nextblockhash":
"0x0b08e2eed05c70f27293521c47f7f60dfc29f9f299ae9909a8552a4a87db7a2"
}
}

```

2.12. 获取连接的节点列表

2.12.1. 方法

getpeers

2.12.2. 作用

获得该节点当前已连接/未连接的节点列表

2.12.3. 调用示例

请求正文:

```

{
  "jsonrpc": "2.0",
  "method": "getpeers",
  "params": [],
  "id": 1
}

```

响应正文:

```

{
  "jsonrpc": "2.0",

```

```

"id": 1,
"result": {
  "unconnected": [
    {
      "address": "::ffff:70.73.16.236",
      "port": 10333
    },
    {
      "address": "::ffff:82.95.77.148",
      "port": 10333
    },
    {
      "address": "::ffff:49.50.215.166",
      "port": 10333
    },
    .....
  ],
  "bad": [
    {
      "address": "::ffff:139.219.226.107",
      "port": 10333
    }
  ],
  "connected": [
    {
      "address": "::ffff:139.219.106.33",
      "port": 10333
    },
    {
      "address": "::ffff:47.88.53.224",
      "port": 10333
    },
    .....
  ]
}
}

```

响应说明：

- 1) **unconnected**: 当前未连接到的节点
- 2) **bad**: 不再连接（拉黑）的节点
- 3) **connected**: 当前已连接到的节点

2.13. 获取当前节点连接数

2.13.1. 方法

getConnectioncount

2.13.2. 作用

获取节点当前的连接数

2.13.3. 调用示例

请求正文：

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

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": 10
}
```

2.14. 查询节点版本信息

2.14.1. 方法

getversion

2.14.2. 作用

获取查询节点的版本信息

2.14.3. 调用示例

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getversion",
  "params": [],
  "id": 3
}
```

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 3,
  "result": {
    "port": 0,
    "nonce": 1546258664,
    "useragent": "/BHP:1.0.0/"
  }
}
```

2.15. 获取共识节点信息及投票情况

2.15.1. 方法

getvalidators

2.15.2. 作用

获取当前 BHP 共识节点的信息及投票情况

2.15.3. 调用示例

请求正文：

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

响应正文：

```
{
```

```

    "jsonrpc": "2.0",
    "id": 1,
    "result": [
      {
        "publickey":
"02486fd15702c4490a26703112a5cc1d0923fd697a33406bd5a1c00e0013b09a70",
        "votes": "46632420",
        "active": true
      },
      {
        "publickey":
"024c7b7fb6c310fccf1ba33b082519d82964ea93868d676662d4a59ad548df0e7d",
        "votes": "46632420",
        "active": true
      },
      {
        "publickey":
"025bdf3f181f53e9696227843950deb72dcd374ded17c057159513c3d0abe20b64",
        "votes": "0",
        "active": false
      },
      .....
    ]
  }

```

响应说明：

- 1) **publickey**: 候选人公钥
- 2) **votes**: 候选人所得票数
- 3) **active: true** 表示该候选人已经成为共识节点，反之，**false** 表示该候选人未成为共识节点

2.16. 获取合约日志

2.16.1. 方法

`getapplicationlog`

2.16.2. 作用

根据指定的交易 ID 获取合约日志

2.16.3. 备注

调用此方法前需要运行命令 `dotnet bhp-cli.dll --log` 启动日志记录，完整的合约日志会记录到 `ApplicationLogs` 目录。

2.16.4. 参数说明

txid: 交易 ID

2.16.5. 调用示例

请求正文:

```
{
  "jsonrpc": "2.0",
  "method": "getapplicationlog",
  "params":
["0x0d03ad35eb8b0bb2e43e18896d22cd2a77fe54fc0b00794fb295bcf96257d0e3"],
  "id": 1
}
```

响应正文:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "txid":
"0x0d03ad35eb8b0bb2e43e18896d22cd2a77fe54fc0b00794fb295bcf96257d0e3",
    "vmstate": "HALT, BREAK",
    "gas_consumed": "2.932",
    "stack": [],
    "notifications": [
      {
        "contract": "0xac116d4b8d4ca55e6b6d4ecce2192039b51cccc5",
        "state": {
          "type": "Array",
          "value": [
            {
              "type": "ByteArray",
              "value": "7472616e73666572"
            },
            {
              "type": "ByteArray",
              "value":

```

```
{
  "jsonrpc": "2.0",
```

```
    "method": "getcontractstate",
    "params": [{"dc675afc61a7c0f7b3d2682bf6e1d8ed865a0e5f"}],
    "id": 1
  }
}
响应正文:
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "version": 0,
    "hash": "0xdc675afc61a7c0f7b3d2682bf6e1d8ed865a0e5f",
    "script":
"5fc56b6c766b00527ac46c766b51527ac46107576f6f6c6f6e676c766b52527ac403574e476c766b
53527ac4006c766b54527ac4210354ae498221046c666efebbaee9bd0eb4823469c98e748494a92
a71f346b1a6616c766b55527ac46c766b00c3066465706c6f79876c766b56527ac46c766b56c3641
6006c766b55c36165f2026c766b57527ac462d8016c766b55c36165d801616c766b00c30b746f74
616c537570706c79876c766b58527ac46c766b58c36440006168164e656f2e53746f726167652e4
76574436f6e7465787406737570706c79617c680f4e656f2e53746f726167652e4765746c766b575
27ac46270016c766b00c3046e616d65876c766b59527ac46c766b59c36412006c766b52c36c766b
57527ac46247016c766b00c30673796d626f6c876c766b5a527ac46c766b5ac36412006c766b53c3
6c766b57527ac4621c016c766b00c308646563696d616c73876c766b5b527ac46c766b5bc364120
06c766b54c36c766b57527ac462ef006c766b00c30962616c616e63654f66876c766b5c527ac46c7
66b5cc36440006168164e656f2e53746f726167652e476574436f6e746578746c766b51c351c3617
c680f4e656f2e53746f726167652e4765746c766b57527ac46293006c766b51c300c36168184e656
f2e52756e74696d652e436865636b5769746e657373009c6c766b5d527ac46c766b5dc3640e0000
6c766b57527ac46255006c766b00c3087472616e73666572876c766b5e527ac46c766b5ec3642c0
06c766b51c300c36c766b51c351c36c766b51c352c36165d40361527265c9016c766b57527ac462
0e00006c766b57527ac46203006c766b57c3616c756653c56b6c766b00527ac4616168164e656f2
e53746f726167652e476574436f6e746578746c766b00c3617c680f4e656f2e53746f726167652e4
765746165700351936c766b51527ac46168164e656f2e53746f726167652e476574436f6e746578
746c766b00c36c766b51c361651103615272680f4e656f2e53746f726167652e507574616168164
e656f2e53746f726167652e476574436f6e7465787406737570706c79617c680f4e656f2e53746f7
26167652e4765746165f40251936c766b52527ac46168164e656f2e53746f726167652e47657443
6f6e7465787406737570706c796c766b52c361659302615272680f4e656f2e53746f726167652e50
757461616c756653c56b6c766b00527ac461516c766b51527ac46168164e656f2e53746f7261676
52e476574436f6e746578746c766b00c36c766b51c361654002615272680f4e656f2e53746f72616
7652e507574616168164e656f2e53746f726167652e476574436f6e7465787406737570706c796c
766b51c361650202615272680f4e656f2e53746f726167652e50757461516c766b52527ac462030
06c766b52c3616c756659c56b6c766b00527ac46c766b51527ac46c766b52527ac4616168164e65
6f2e53746f726167652e476574436f6e746578746c766b00c3617c680f4e656f2e53746f72616765
2e4765746c766b53527ac46168164e656f2e53746f726167652e476574436f6e746578746c766b5
1c3617c680f4e656f2e53746f726167652e4765746c766b54527ac46c766b53c3616576016c766b5
2c3946c766b55527ac46c766b54c3616560016c766b52c3936c766b56527ac46c766b55c300a264
0d006c766b52c300a2620400006c766b57527ac46c766b57c364ec00616168164e656f2e53746f72
```

```

6167652e476574436f6e746578746c766b00c36c766b55c36165d800615272680f4e656f2e53746f
726167652e507574616168164e656f2e53746f726167652e476574436f6e746578746c766b51c36
c766b56c361659c00615272680f4e656f2e53746f726167652e5075746155c57600135472616e73
666572205375636365737366756cc476516c766b00c3c476526c766b51c3c476536c766b52c3c47
6546168184e656f2e426c6f636b636861696e2e476574486569676874c46168124e656f2e52756e
74696d652e4e6f7469667961516c766b58527ac4620e00006c766b58527ac46203006c766b58c36
16c756653c56b6c766b00527ac4616c766b00c36c766b51527ac46c766b51c36c766b52527ac462
03006c766b52c3616c756653c56b6c766b00527ac461516c766b00c36a527a527ac46c766b51c36c
766b52527ac46203006c766b52c3616c7566",
    "parameters": [
        "ByteArray"
    ],
    "returntype": "ByteArray",
    "name": "Woolong",
    "code_version": "0.9.2",
    "author": "llwlvlll",
    "email": "llwlvlll@gmail.com",
    "description": "GO BHP!!!",
    "properties": {
        "storage": true,
        "dynamic_invoke": false
    }
}
}

```

2.18. 以散列值调用智能合约

2.18.1. 方法

- 1) **invoke**: 使用给定的参数以散列值调用智能合约，并返回结果
- 2) **invokefunction**: 使用给定的操作和参数，以散列值调用智能合约之后返回结果
- 3) **invokescript**: 通过虚拟机传递脚本之后返回结果

2.18.2. 备注

此方法用于测试你的虚拟机脚本，如同在区块链上运行一样。这个 RPC 调用对区块链没有任何影响

2.18.3. 参数说明

1. invoke

- a. scripthash: 智能合约脚本散列值
 - b. params: 传递给智能合约的参数
- 2. invokefunction**
- a. scripthash: 智能合约脚本散列值
 - b. operation: 操作名称（字符串）
 - c. params: 传递给智能合约操作的参数
- 3. invokescript**
- a. script: 一个由虚拟机运行的脚本，与 InvocationTransaction 中携带的脚本相同

2.18.4. 调用示例

响应正文类似，因此之后示例只列出请求正文。

示例 1 invoke:

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "invoke",
  "params": [
    "dc675afc61a7c0f7b3d2682bf6e1d8ed865a0e5f",
    [
      {
        "type": "String",
        "value": "name"
      },
      {
        "type": "Boolean",
        "value": false
      }
    ]
  ],
  "id": 1
}
```

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "script": "00046e616d65675f0e5a86edd8e1f62b68d2b3f7c0a761fc5a67dc",
    "state": "HALT, BREAK",
    "gas_consumed": "2.489",
    "stack": [
      {
        "type": "ByteArray",
        "value": "576f6f6c6f6e67"
```

```

    }
  ],
  "tx":
"d1011b00046e616d65675f0e5a86edd8e1f62b68d2b3f7c0a761fc5a67dc000000000000000000
00000"
}
}

```

示例 2 invokefunction 含参:

请求正文:

```

{
  "jsonrpc": "2.0",
  "method": "invokefunction",
  "params": [
    "af7c7328eee5a275a3bcaee2bf0cf662b5e739be",
    "balanceOf",
    [
      {
        "type": "Hash160",
        "value": "91b83e96f2a7c4fdf0c1688441ec61986c7cae26"
      }
    ]
  ],
  "id": 3
}

```

示例 3 invokefunction 无参:

请求正文:

```

{
  "jsonrpc": "2.0",
  "method": "invokefunction",
  "params": [
    "af7c7328eee5a275a3bcaee2bf0cf662b5e739be",
    "decimals"
  ],
  "id": 3
}

```

示例 4 invokescript:

```

{
  "jsonrpc": "2.0",
  "method": "invokescript",
  "params": ["00046e616d656724058e5e1b6008847cd662728549088a9ee82191"],
  "id": 3
}

```


2.19. 广播交易

2.19.1. 方法

sendrawtransaction

2.19.2. 作用

广播交易

2.19.3. 参数说明

hex: 在程序中构造的已签名的交易序列化后的 16 进制字符串

2.19.4. 调用示例

请求正文:

```
{
  "jsonrpc": "2.0",
  "method": "sendrawtransaction",
  "params":
  ["80000001195876cb34364dc38b730077156c6bc3a7fc570044a66fbfeeea56f71327e8ab0000029
b7cffdaa674beae0f930ebe6085af9093e5fe56b34a5c220ccdcf6efc336fc500c65eaf440000000f9a
23e06f74cf86b8827a9108ec2e0f89ad956c9b7cffdaa674beae0f930ebe6085af9093e5fe56b34a5c
220ccdcf6efc336fc50092e14b5e00000030aab52ad93f6ce17ca07fa88fc191828c58cb7101414091
5467ecd359684b2dc358024ca750609591aa731a0b309c7fb3cab5cd0836ad3992aa0a24da431f43
b68883ea5651d548feb6bd3c8e16376e6e426f91f84c58232205572f35c7819267e721335948d385
fae5be66e7ba8c748ac15467dcca0693692dac"],
  "id": 1
}
```

响应正文:

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

响应说明:

- 1) 当 result 为 true 时表明当前交易广播成功
- 2) 当 result 为 false 时表示当前交易广播失败, 原因可能有双重花费、签名不完整等
- 3) 本示例中广播了一个已经确认的交易, 因为双重花费所以广播失败

2.20. 获取交易列表

2.20.1. 方法

getrawmempool

2.20.2. 作用

获取内存中未确认的交易列表

2.20.3. 调用示例

请求正文：

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

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": [
    "0x9786cce0dddb524c40ddb5e31a41ed1f6b5c8a683c122f627ca4a007a7cf4e",
    "0xb488ad25eb474f89d5ca3f985cc047ca96bc7373a6d3da8c0f192722896c1cd7",
    "0xf86f6f2c08bf766ebe59dc84bc3b8829f1053f0a01deb26bf7960d99fa86cd6"
  ]
}
```

响应说明：这些是节点收到的未确定的交易，即零确认的交易

2.21. 获取交易信息

2.21.1. 方法

getrawtransaction

2.21.2. 作用

根据指定的散列值，返回对应的交易信息

2.21.3. 参数说明

- 1) txid: 交易 ID
- 2) verbose: 可选参数，verbose 默认值为 0，verbose 为 0 时返回的是交易的序列化后的信息，用 16 进制字符串表示，如果从中获取详细信息需要调用 SDK 来进行反序列化。verbose 为 1 时返回的是对应交易的详细信息，用 Json 格式字符串表示

2.21.4. 调用示例

示例 1 返回区块序列化后的信息：

示例请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getrawtransaction",
  "params":
  ["f4250dab094c38d8265acc15c366dc508d2e14bf5699e12d9df26577ed74d657"],
  "id": 1
}
```

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result":
  "80000001195876cb34364dc38b730077156c6bc3a7fc570044a66fbfeeea56f71327e8ab0000029
  b7cffdaa674beae0f930ebe6085af9093e5fe56b34a5c220ccdcf6efc336fc500c65eaf44000000f9a
  23e06f74cf86b8827a9108ec2e0f89ad956c9b7cffdaa674beae0f930ebe6085af9093e5fe56b34a5c
  220ccdcf6efc336fc50092e14b5e00000030aab52ad93f6ce17ca07fa88fc191828c58cb7101414091
  5467ecd359684b2dc358024ca750609591aa731a0b309c7fb3cab5cd0836ad3992aa0a24da431f43
  b68883ea5651d548feb6bd3c8e16376e6e426f91f84c58232205572f35c7819267e721335948d385
  fae5be66e7ba8c748ac15467dcca0693692dac"
}
```

示例 2 返回 JSON 格式：

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getrawtransaction",
  "params":
```

```
[{"f4250dab094c38d8265acc15c366dc508d2e14bf5699e12d9df26577ed74d657", 1},
  {
    "id": 1
  }
]
响应正文:
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "txid":
"0xf4250dab094c38d8265acc15c366dc508d2e14bf5699e12d9df26577ed74d657",
    "size": 262,
    "type": "ContractTransaction",
    "version": 0,
    "attributes": [],
    "vin": [
      {
        "txid":
"0xab82713f756eae6bf6fa6440057fca7c36b6c157700738bc34d3634cb765819",
        "vout": 0
      }
    ],
    "vout": [
      {
        "n": 0,
        "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": "2950",
        "address": "AHCNSDkh2Xs66SzmyKGdoDKY752uyeXDrt"
      },
      {
        "n": 1,
        "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": "4050",
        "address": "ALDCagdWUVV4wYoEzCcJ4dtHqtWhsNEEaR"
      }
    ],
    "sys_fee": "0",
    "net_fee": "0",
    "scripts": [
      {
        "invocation":
"40915467ecd359684b2dc358024ca750609591aa731a0b309c7fb3cab5cd0836ad3992aa0a24da431f43b68883ea5651d548feb6bd3c8e16376e6e426f91f84c58",

```

```

        "verification":
"2205572f35c7819267e721335948d385fae5be66e7ba8c748ac15467dcca0693692dac"
    }
],
    "blockhash":
"ox9c814276156d33f5dbd4e1bd4e279bb4da4ca73ea7b7f9f0833231854648a72c",
    "confirmations": 144,
    "blocktime": 1496719422
}
}

```

2.22. 获取未花费交易输出信息

2.22.1. 方法

gettxout

2.22.2. 作用

根据指定的散列和索引，返回对应的 `unspent` 交易输出（零钱）信息。如果交易输出已经花费，返回结果为空

2.22.3. 参数说明

- 1) `txid`: 交易 ID
- 2) `n`: 要获取的交易输出在该交易中的索引（从 0 开始）

2.22.4. 调用示例

请求正文：

```

{
    "jsonrpc": "2.0",
    "method": "gettxout",
    "params":
["f4250dab094c38d8265acc15c366dc508d2e14bf5699e12d9df26577ed74d657", 0],
    "id": 1
}

```

响应正文：

```

{
    "jsonrpc": "2.0",
    "id": 1,

```

```
"result": {  
  "n": 0,  
  "asset": "0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",  
  "value": "2950",  
  "address": "AHCNSDkh2Xs66SzmyKGdoDKY752uyeXDrt"  
}  
}
```

2.23. 获取指定的存储值

2.23.1. 方法

getstorage

2.23.2. 作用

根据合约脚本散列和存储的 key，返回存储的 value

2.23.3. 参数说明

- 1) script_hash: 合约脚本散列。
- 2) key: 存储区的键。（需要转化为 hex string）

2.23.4. 调用示例

请求正文：

```
{  
  "jsonrpc": "2.0",  
  "method": "getstorage",  
  "params": ["03febccf81ac85e3d795bc5cbd4e84e907812aa3", "5065746572"],  
  "id": 15  
}
```

响应正文：

```
{  
  "jsonrpc": "2.0",  
  "id": 15,  
  "result": "4c696e"  
}
```

2.24. 获取系统手续费

2.24.1. 方法

getblocksysfee

2.24.2. 作用

根据指定的索引，返回截止到该区块前的系统手续费

2.24.3. 参数说明

index: 区块索引

2.24.4. 调用示例

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "getblocksysfee",
  "params": [1005434],
  "id": 1
}
```

响应正文：

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": "195500"
}
```

响应说明：result 为截止到该区块前的系统手续费，单位为 BhpGas

2.25. 向指定地址转账

2.25.1. 方法

- 1) sendfrom: 从指定地址，向指定地址转账
- 2) sendtoaddress: 向指定地址转账

2.25.2. 备注

执行此命令前需要在 Bhp-CLI 节点中打开钱包

2.25.3. 参数说明

- 1) **asset_id**: 资产 ID (资产标识符)，即该资产在注册时的 **RegistTransaction** 的交易 ID。其余资产 ID 可以通过 CLI 命令中的 **list asset** 命令查询，也可以在区块链浏览器中查询。
- 2) **from**: 转账地址 (**sendfrom** 使用)。
- 3) **to**: 收款地址 (**sendfrom** 使用)。
- 4) **address**: 收款地址 (**sendtoaddress** 使用)。
- 5) **value**: 转账金额。
- 6) **fee**: 手续费，可选参数，默认为 0。

2.25.4. 调用示例

响应正文和响应说明与 **sendfrom** 类似，因此后面的示例只列出请求正文

示例 1 **sendfrom**:

请求正文:

```
{
  "jsonrpc": "2.0",
  "method": "sendfrom",
  "params":
  ["602c79718b16e442de58778e148d0b1084e3b2dffd5de6b7b16cee7969282de7","AWg3L6W68bFfSS13Tf4rt8CRdG2ktaAjGb","AWg3L6W68bFfSS13Tf4rt8CRdG2ktaAjGb",1],
  "id": 1
}
```

响应正文:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "txid":
    "0x60170ad03627ce45c7dd56ecec3b3b26eab0845aa8b2cbbeecaefc5771b9eb1",
    "size": 262,
    "type": "ContractTransaction",
    "version": 0,
    "attributes": [],
    "vin": [
      {
        "txid":
```



```

"0xd2188c1bd454ac883d79826e5c677deedb91cc61ec6d819df48ff4a963873adb",
    "vout": 1
  }
],
"vout": [
  {
    "n": 0,
    "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
    "value": "1",
    "address": "AWg3L6W68bFfSS13Tf4rt8CRdG2ktaAjGb"
  },
  {
    "n": 1,
    "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
    "value": "17.4798197",
    "address": "AWg3L6W68bFfSS13Tf4rt8CRdG2ktaAjGb"
  }
],
"sys_fee": "0",
"net_fee": "0",
"scripts": [
  {
    "invocation":
"40a8d40e1652d7ad0c7bb59ef8217237037824af54ee5e46f2fd096c44dd46ef27fa7255010e2a8a
2166af8a904e13b96bd3ac82e791633685824c35e7f2731e79",
    "verification":
"2102883118351f8f47107c83ab634dc7e4ffe29d274e7d3dcf70159c8935ff769bebac"
  }
]
}
}

```

响应说明：

- 1) 返回如上的交易详情说明交易发送成功，否则交易发送失败
- 2) 如果签名不完整会返回待签名的交易
- 3) 如果余额不足会返回错误信息

示例 2 sendtoaddress:

请求正文：

```

{
  "jsonrpc": "2.0",
  "method": "sendtoaddress",
  "params":
["c56f33fc6ecfd0c225c4ab356fee59390af8560be0e930faebe74a6daff7c9b","AK4if54jXjSiJBs6jkf

```

```
ZjxAastauJtjise",1],
  "id": 1
}
```

2.26. 批量转账

2.26.1. 方法

sendmany

2.26.2. 作用

批量转账命令，并且可以指定找零地址

2.26.3. 备注

执行此命令前需要在 Bhp-CLI 节点中打开钱包

2.26.4. 参数说明

<outputs_array> [fee=0] [change_address]

outputs_array: 数组，数组中的每个元素的数据结构如下：

{"asset": \<asset>,"value": \<value>,"address": \<address>}

- 1) asset: 资产 ID（资产标识符），即该资产在注册时的 RegistTransaction 的交易 ID。其余资产 ID 可以通过 [CLI 命令](../cli.md) 中的 `list asset` 命令查询，也可以在区块链浏览器中查询
- 2) value: 转账金额
- 3) address: 收款地址
- 4) fee: 手续费，可选参数，默认为 0
- 5) change_address: 找零地址，可选参数，默认为钱包中第一个标准地址

2.26.5. 调用示例

请求正文：

```
{
  "jsonrpc": "2.0",
  "method": "sendmany",
  "params": [
    [
      {
```

```

        "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": 1,
        "address": "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
    },
    {
        "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": 1,
        "address": "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
    }
]
},
"id": 1
}

```

请求正文（包含手续费和找零地址）：

```

{
    "jsonrpc": "2.0",
    "method": "sendmany",
    "params": [
        [
            {
                "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
                "value": 1,
                "address": "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
            },
            {
                "asset":
"0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
                "value": 1,
                "address": "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
            }
        ],
        0,
        "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
    ],
    "id": 1
}

```

响应正文：

```

{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {

```

```

    "txid":
    "0x55ba819b50f5821298328f3bf9bb17e088afc900cf2ad7dbfc03d49940b5cf30",
    "size": 322,
    "type": "ContractTransaction",
    "version": 0,
    "attributes": [],
    "vin": [
      {
        "txid":
        "0x06de043b9b914f04633c580ab02d89ba55556f775118a292adb6803208857c91",
        "vout": 1
      }
    ],
    "vout": [
      {
        "n": 0,
        "asset":
        "0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": "1",
        "address": "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
      },
      {
        "n": 1,
        "asset":
        "0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": "1",
        "address": "AbRTHXb9zqdqn5sVh4EYpQHGGZ536FgwCx2"
      },
      {
        "n": 2,
        "asset":
        "0x13f76fabfe19f3ec7fd54d63179a156bafc44afc53a7f07a7a15f6724c0aa854",
        "value": "495",
        "address": "AK5q8peiC4QKwuZHWX5Dkqhmar1TAGvZBS"
      }
    ],
    "sys_fee": "0",
    "net_fee": "0",
    "scripts": [
      {
        "invocation":
        "406e545e30a6b39f71a7a40f1d4937939b9e1ca38851449842a2e2318bd499afd9c89f0c9665892
        3e3e435ee91192e9dbf101d81a240fa7c953ac0c322d2f2b980",
        "verification":

```

"2103cf5ba6a9135f8eaeda771658564a855c1328af6b6808635496a4f51e3d29ac3eac"

}

]

}

}

响应说明：

- 1) 返回如上的交易详情说明交易发送成功，否则交易发送失败
- 2) JSON 格式不正确，会返回 Parse error
- 3) 如果签名不完整会返回待签名的交易
- 4) 如果余额不足会返回错误信息