Welcome to gdw’s documentation!

Contents:

rpc接口文档

Contents:

基础接口相关

rpc访问钱包登录函数

调用前提：1.在gdw平台数据目录下的config.json下配置好相关的rpc服务器参数 2.启动gdw平台

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "username",
        "password"
    ],
    "id": "1",
    "method": "login"
}
```

请求方法：login

请求参数：rpc 用户名、 rpc 密码

Response Result:

```
{
    "id": "1",
    "result": true
}
```

返回参数:

无:

显示版本号以及客户端的相关信息

调用前提：无

Request:

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

请求方法：about

请求参数：无

Response Result:

```
{
    "id": "2",
    "result": {
        "fc_revision_age": "46 years ago",
        "blockchain_name": "GDW",
        "gdw_revision_age": "64 weeks ago",
        "boost_version": "1.55",
        "openssl_version": "OpenSSL 1.0.1g 7 Apr 2014",
        "compile_date": "compiled on Mar 28 2016 at 11:50:01",
        "build": "win32 64-bit",
        "gdw_revision": "1.2",
        "fc_revision": "0",
        "client_version": "",
        "blockchain_description": "The Future of Banking"
    }
}
```

返回参数:

blockchain_name: blockchain名称

blockchain_description: blockchain描述

client_version: 客户端版本

gdw_revision: gdw版本

fc_revision: fc库版本

fc_revision_age: fc发布时间

compile_date: 编译日期

boost_version: boost版本

openssl_version: openssl 版本

build: 编译平台

获取当前链以及钱包基础信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "2",
    "method": "get_info"
}
```

请求方法：get_info

请求参数：无

Response Result:

```
{
    "id": "2",
    "result": {
        "blockchain_head_block_timestamp": "2016-03-28T07:43:00",
        "blockchain_average_delegate_participation": "1.46017699115044250",
        "wallet_unlocked": true,
        "wallet_next_block_production_time": null,
        "blockchain_share_supply": "10000578600000",
        "network_num_connections": "0",
        "blockchain_next_round_timestamp": "2016-03-29T02:25:40",
        "ntp_time_error": "-0.10100400000000000",
        "blockchain_random_seed": "fca3b1dd8ad99aee1451fcb8426185af01769942",
        "blockchain_next_round_time": "550",
        "wallet_block_production_enabled": false,
        "network_num_connections_max": "200",
        "wallet_last_scanned_block_timestamp": "2016-03-28T07:43:00",
        "wallet_next_block_production_timestamp": null,
        "wallet_unlocked_until_timestamp": "1911-10-30T20:53:40",
        "wallet_open": true,
        "blockchain_head_block_num": "5786",
        "blockchain_confirmation_requirement": "1",
        "network_chain_downloader_blocks_remaining": null,
        "network_chain_downloader_running": false,
        "client_data_dir": "e:/gdwtest/3",
        "blockchain_head_block_age": "66810",
        "blockchain_blocks_left_in_round": "55",
        "wallet_unlocked_until": "999997526",
        "ntp_time": "2016-03-29T02:16:30",
        "wallet_scan_progress": "1.00000000000000000",
        "client_version": "",
        "blockchain_head_block_id": "a8b1ce125f3cbb4e3454a532ca9360bbe9f9dfa4"
    }
}
```

返回参数:

blockchain_head_block_num: 区块号

blockchain_head_block_age: 当前最大块的产生的时长

blockchain_head_block_timestamp: 当前块产生的时间

blockchain_head_block_id: 当前块ID

blockchain_average_delegate_participation: 代理参与率

blockchain_confirmation_requirement: 区块确认周期

blockchain_share_supply: GDW发行量

blockchain_blocks_left_in_round: 本轮出块还有多少剩余

blockchain_next_round_time: 距下一轮出块剩余时间

blockchain_next_round_timestamp: 下一轮出块开始时间

blockchain_random_seed: 当前轮出块

client_data_dir: 本地客户端数据存放路径

client_version: 客户端版本

network_num_connections: 当前连接数

network_num_connections_max: 当前客户端最大连接数

network_chain_downloader_running: 当前是否高速同步模式

network_chain_downloader_blocks_remaining: 当前剩余未同步块数

ntp_time: NTP获取时间戳

ntp_time_error: NTP时间修正参数

wallet_open: 钱包是否打开

wallet_unlocked: 钱包是否解锁

wallet_unlocked_until: 钱包解锁持续秒数

wallet_unlocked_until_timestamp: 钱包下次锁定时间

wallet_last_scanned_block_timestamp: 钱包上一次扫描时间

wallet_scan_progress: 钱包扫描线程数

wallet_block_production_enabled: 钱包是否产块

wallet_next_block_production_time: 钱包内下一轮产块时间

wallet_next_block_production_timestamp: 钱包内下一轮产块时间戳

判断输入的公钥是否合法

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "public_key"
    ],
    "id": "2",
    "method": "validate_address"
}
```

请求方法：validate_address

请求参数：输入的公钥

Response Result:

```
{
    "id": "2",
    "result": {
        "isvalid": true
    }
}
```

返回参数:

isvalid: 是否是正常的公钥

临时设置rpc的登录用户名

调用前提：需要使用rpc_start_server指令让这条指令生效

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "username"
    ],
    "id": "2",
    "method": "rpc_set_username"
}
```

请求方法：rpc_set_username

请求参数：rpc登录用户名

Response Result:

```
{
    "id": "2",
    "result": "null"
}
```

返回参数:

无:

临时设置rpc登录用户密码

调用前提：需要使用rpc_start_server指令让这条指令生效

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "password"
    ],
    "id": "2",
    "method": "rpc_set_password"
}
```

请求方法：rpc_set_password

请求参数：rpc登录密码

Response Result:

```
{
    "id": "3",
    "result": "null"
}
```

返回参数:

无:

临时启动客户端rpc模式

调用前提：需要断开这次rpc的链接，重新进行rpc链接才能看出API执行效果

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "port"
    ],
    "id": "2",
    "method": "rpc_start_server"
}
```

请求方法：rpc_start_server

请求参数：rpc服务端口号

Response Result:

```
{
    "id": "4",
    "result": "null"
}
```

返回参数:

无:

更新NTP时间

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "2",
    "method": "ntp_update_time"
}
```

请求方法：ntp_update_time

请求参数：无

Response Result:

```
{
    "id": "5",
    "result": "null"
}
```

返回参数:

无:

显示当前数据占用的硬盘空间

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "2",
    "method": "disk_usage"
}
```

请求方法：disk_usage

请求参数：无

Response Result:

```
{
    "id": "2",
    "result": {
        "logs": "4085001",
        "mail_client": null,
        "blockchain": "3482597",
        "wallets": {
            ".backups": "3237",
            "op": "402051"
        },
        "network_peers": "289497",
        "dac_state": "7642885",
        "mail_server": null,
        "total": "16171905"
    }
}
```

返回参数:

blockchain: chain数据占用空间

dac_state: 本地数据占用空间

logs: 日志占用空间

mail_client: 保留字段

mail_server: 保留字段

network_peers: 网络连接池占用空间

wallets: 钱包文件夹，下面的项跟你本地钱包有关系

.backups: 备份占用空间

op: 名称为op的钱包占用空间

total: 总计占用空间

获取指定区块的全部详细交易信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "block"
    ],
    "id": "2",
    "method": "blockchain_get_block_transactions"
}
```

请求方法：blockchain_get_block_transactions

请求参数：要查询区块的区块id或者区块编号

Response Result:

```
{
    "id": "2",
    "result": [
        [
            "7e8b7797d42d1ed8f8b1beb3e92f77ae6842c6a3",
            {
                "withdraws": [
                    [
                        "0",
                        "10000001000"
                    ]
                ],
                "imessage_length": "0",
                "deltas": [
                    [
                        "0",
                        [
                            [
                                "0",
                                "-10000001000"
                            ]
                        ]
                    ],
                    [
                        "1",
                        [
                            [
                                "0",
                                "10000000000"
                            ]
                        ]
                    ]
                ],
                "required_fees": {
                    "asset_id": "0",
                    "amount": "0"
                },
                "alt_fees_paid": {
                    "asset_id": "0",
                    "amount": "0"
                },
                "deposits": [
                    [
                        "0",
                        "10000000000"
                    ]
                ],
                "yield": [],
                "trx": {
                    "gdw_inport_asset": {
                        "asset_id": "0",
                        "amount": "10000000000"
                    },
                    "operations": [
                        {
                            "type": "withdraw_op_type",
                            "data": {
                                "claim_input_data": "",
                                "amount": "10000001000",
                                "balance_id": "GQJ2gaD8RiKpiWRcT3AAEj4vr5eAJpXj52"
                            }
                        },
                        {
                            "type": "deposit_op_type",
                            "data": {
                                "amount": "10000000000",
                                "condition": {
                                    "asset_id": "0",
                                    "slate_id": "0",
                                    "type": "withdraw_signature_type",
                                    "data": {
                                        "owner": "G6b7RN33FnT5MHPNwXqwWZBFZ17ArA3wm3"
                                    }
                                }
                            }
                        }
                    ],
                    "signatures": [
"2004f4f30c748da49ddf94468fcfbff02c72f9ac5784d56b9db5efc1d5470182034a8eb3866f42ca84013b218193bf4f
                    ],
                    "expiration": "2016-03-28T08:00:48",
                    "gdw_account": "G6b7RN33FnT5MHPNwXqwWZBFZ17ArA3wm31"
                },
                "signed_keys": [],
                "chain_location": {
                    "trx_num": "0",
                    "block_num": "5533"
                },
                "delegate_vote_deltas": [],
                "balance": [
                    [
                        "0",
                        "1000"
                    ]
                ]
            }
        ]
    ]
}
```

返回参数:

在trx前的数据，没有数据名称: 交易单号

trx: 结构体类型名

signed_keys: 保留字段

deposits: 入账资产

withdraws: 出账资产

yield: 保留参数

deltas: 保留参数

required_fees: 保留参数

alt_fees_paid: 保留参数

balance: 保留参数

delegate_vote_deltas: 保留参数

imessage_length: 备注长度

chain_location: 交易位置结构体

获取指定帐号名称或ID 的账号详细信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account"
    ],
    "id": "2",
    "method": "blockchain_get_account"
}
```

请求方法：blockchain_get_account

请求参数：要查询的账户名称或者要查询的账户id、账户地址以及账户公钥

Response Result:

```
{
    "id": "2",
    "result": {
        "owner_key": "G8KkneVfc6iaR9grBpRij5wqhPK5XwxS7ohNhJ5tH8DEoL77j32",
        "name": "gdw1",
        "registration_date": "2015-05-31T16:00:00",
        "last_update": "2015-05-31T16:00:00",
        "delegate_info": {
            "pay_balance": "20090309",
            "total_burned": "0",
            "blocks_missed": "810",
            "pay_rate": "100",
            "votes_for": "20090309",
            "blocks_produced": "58",
            "last_block_num_produced": "5782",
            "total_paid": "20090309",
            "signing_key_history": [
                [
                    "0",
                    "G8KkneVfc6iaR9grBpRij5wqhPK5XwxS7ohNhJ5tH8DEoL77j32"
                ]
            ],
            "next_secret_hash": "7e7f255f175ffaf52397084f6ca747674088f176"
        },
        "public_data": null,
        "id": "2",
        "active_key_history": [
            [
                "2015-05-31T16:00:00",
                "G8KkneVfc6iaR9grBpRij5wqhPK5XwxS7ohNhJ5tH8DEoL77j32"
            ]
        ]
    }
}
```

返回参数:

id: 账户ID

name: 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

delegate_info: 代理信息结构体

meta_data: 保留字段

查询blockchain的信息和设定参数

调用前提：无

Request:

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

请求方法：blockchain_get_info

请求参数：无

Response Result:

```
{
    "id": "1",
    "result": {
        "db_version": "201",
        "symbol_size_min": "3",
        "block_interval": "10",
        "max_delegate_pay_issued_per_block": "507977",
        "memo_size_max": "51",
        "symbol_size_max": "8",
        "statistics_enabled": false,
        "max_pending_queue_size": "1000",
        "genesis_timestamp": "2015-05-31T16:00:00",
        "delegate_num": "99",
        "name_size_max": "63",
        "symbol": "GDW",
        "short_symbol_asset_reg_fee": "50000000000",
        "data_size_max": "65536",
        "address_prefix": "GDW",
        "max_trx_per_second": "100",
        "asset_shares_max": "1000000000000000",
        "long_symbol_asset_reg_fee": "50000000",
        "blockchain_id": "94beaf23764844cae3e1f368273d0dddc8bc06c575c0cf6578bcb1f2012e7694",
        "name": "GDW",
        "max_delegate_reg_fee": "122181818",
        "relay_fee": "1000"
    }
}
```

返回参数:

blockchain_id: 创世快id号

name: blockchain的名称

symbol: 资产符号

address_prefix: 地址前缀

db_version: blockchain的数据版本

genesis_timestamp: 创世快的创建时间

block_interval: 出块时间间隔

delegate_num: 代理总数

max_delegate_pay_issued_per_block: 代理最大可领取GDW数量

max_delegate_reg_fee: 注册为代理所需费用

name_size_max: 注册用户名称最大长度限制

memo_size_max: 备注长度限制

data_size_max: 用户公开数据最大长度

symbol_size_max: 资产符号最大长度

symbol_size_min: 资产符号最小长度

asset_shares_max: 资产发行数量最大限制

short_symbol_asset_reg_fee: 最短注册符号注册费用

long_symbol_asset_reg_fee: 最长符号注册费用

statistics_enabled: 是否打开统计模式

relay_fee: 所需手续费

max_pending_queue_size: 区队列最大长度

max_trx_per_second: 最大每秒交易数

返回当前基础资产分配的快照

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "filename"
    ],
    "id": "2",
    "method": "blockchain_generate_snapshot"
}
```

请求方法：blockchain_generate_snapshot

请求参数：导出资产文档存放的位置

Response Result:

```
{
    "id": "2",
    "result": null
}
```

返回参数:

无:

传回是否本地区块链已经和网络同步的信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "2",
    "method": "blockchain_is_synced"
}
```

请求方法：blockchain_is_synced

请求参数：无

Response Result:

```
{
    "id": "2",
    "result": true
}
```

返回参数:

result: 是否同步中(已经同步的区块时间在两天以内，就不属于同步中)

查询当前总共有多少块

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "2",
    "method": "blockchain_get_block_count"
}
```

请求方法：blockchain_get_block_count

请求参数：无

Response Result:

```
{
    "id": "2",
    "result": "5534"
}
```

返回参数:

result: 当前区块拥有块数

获取指定的balance_id下的余额记录

调用前提：balanceid可从转账的时候从生成的交易单的withdraw_op_type中查看

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "balance_id"
    ],
    "id": "2",
    "method": "blockchain_get_balance"
}
```

请求方法：blockchain_get_balance

请求参数：要查询的余额id

Response Result:

```
{
    "id": "2",
    "result": {
        "deposit_date": "2016-06-20T05:32:35",
        "snapshot_info": {
            "original_address": "GL34D5qcbMU56mRRByLFVvDVVEvZe9sVn6",
            "original_balance": "1000000000000"
        },
        "last_update": "2016-06-20T06:21:30",
        "meta_data": "null",
        "balance": "196009",
        "condition": {
            "asset_id": "0",
            "slate_id": "0",
            "type": "withdraw_signature_type",
            "data": {
                "owner": "GL34D5qcbMU56mRRByLFVvDVVEvZe9sVn6"
            }
        }
    }
}
```

返回参数:

condition: 金额信息结构体

balance: 余额数量

snapshot_info: 创世资产结构体

deposit_date: 余额增加时间

last_update: 最近balance更新时间

meta_data: 保留字段

列出被指定公钥拥有的balance

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "public_key"
    ],
    "id": "2",
    "method": "blockchain_list_key_balances"
}
```

请求方法：blockchain_list_key_balances

请求参数：要查询余额的公钥

Response Result:

```
{
    "id": "2",
    "result": [
        [
            "GNXWy8jSAkuWRoxLDKU7JL7CT6k7kP7Afc",
            {
                "deposit_date": "1970-01-01T00:04:50",
                "snapshot_info": {
                    "original_address": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy",
                    "original_balance": "1000000000000"
                },
                "last_update": "2016-03-28T06:58:50",
                "meta_data": null,
                "balance": "1000000200000",
                "condition": {
                    "asset_id": "0",
                    "slate_id": "0",
                    "type": "withdraw_signature_type",
                    "data": {
                        "owner": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
                    }
                }
            }
        ]
    ]
}
```

返回参数:

在所有数据前的数据，没有数据名称的字段: balanceid

condition: 金额信息结构体

balance: 余额数量

snapshot_info: 创世资产结构体

deposit_date: 余额增加时间

last_update: 最近balance更新时间

meta_data: 保留字段

以签署区块的顺序传回目前回合的现任受托人信息

调用前提：内容较多，可能用普通工具没有办法接收全部返回。

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "start_num",
        "limit_num"
    ],
    "id": "2",
    "method": "blockchain_list_active_delegates"
}
```

请求方法：blockchain_list_active_delegates

请求参数：起始顺序号、 返回数量

Response Result:

```
{
    "id": "2",
    "result": [
        {
            "owner_key": "G8XfYBVvdxrSFbW2Ehtzhf1vAQNq21nTiMpyE6QGvoydtY2aXcZ",
            "name": "gdw42",
            "registration_date": "2015-05-31T16:00:00",
            "last_update": "2015-05-31T16:00:00",
            "delegate_info": {
                "pay_balance": "19586615",
                "total_burned": "0",
                "blocks_missed": "809",
                "pay_rate": "100",
                "votes_for": "19586615",
                "blocks_produced": "57",
                "last_block_num_produced": "5737",
                "total_paid": "19586615",
                "signing_key_history": [
                    [
                        "0",
                        "G8XfYBVvdxrSFbW2Ehtzhf1vAQNq21nTiMpyE6QGvoydtY2aXcZ"
                    ]
                ],
                "next_secret_hash": "7bf4dcb7b7e7887053638b5ac9d637930a91a535"
            },
            "public_data": null,
            "id": "43",
            "active_key_history": [
                [
                    "2015-05-31T16:00:00",
                    "G8XfYBVvdxrSFbW2Ehtzhf1vAQNq21nTiMpyE6QGvoydtY2aXcZ"
                ]
            ]
        }
    ]
}
```

返回参数:

id: 账户ID

name: 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

delegate_info: 代理信息结构体

传回已注册的帐号名称, 从指定的帐号名称开始, 直到所提供的数量上限结束

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "first",
        "limit"
    ],
    "id": "2",
    "method": "blockchain_list_accounts"
}
```

请求方法：blockchain_list_accounts

请求参数：其实查找首字母，默认为”a”、 查找数量，默认为20

Response Result:

```
{
    "id": "2",
    "result": [
        {
            "owner_key": "G6vBWNEsnNS5rMXSHsXLDxp3mMyBZd5M9gAgtMbvLevHQVrCgYY",
            "name": "gdw0",
            "registration_date": "2015-05-31T16:00:00",
            "last_update": "2015-05-31T16:00:00",
            "delegate_info": {
                "pay_balance": "20288476",
                "total_burned": "0",
                "blocks_missed": "804",
                "pay_rate": "100",
                "votes_for": "20288476",
                "blocks_produced": "64",
                "last_block_num_produced": "5645",
                "total_paid": "20288476",
                "signing_key_history": [
                    [
                        "0",
                        "G6vBWNEsnNS5rMXSHsXLDxp3mMyBZd5M9gAgtMbvLevHQVrCgYY"
                    ]
                ],
                "next_secret_hash": "4e549122f53e486cd2f5741965bcb030f6c714bc"
            },
            "public_data": null,
            "id": "1",
            "active_key_history": [
                [
                    "2015-05-31T16:00:00",
                    "G6vBWNEsnNS5rMXSHsXLDxp3mMyBZd5M9gAgtMbvLevHQVrCgYY"
                ]
            ]
        }
    ]
}
```

返回参数:

id: 账户ID

name : 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

delegate_info: 代理信息结构体

meta_data: 保留字段

传回尚未收录进区块的交易列表信息发送消息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "2",
    "method": "blockchain_list_pending_transactions"
}
```

请求方法：blockchain_list_pending_transactions

请求参数：无

Response Result:

```
{
    "id": "2",
    "result": [
        {
            "gdw_inport_asset": {
                "asset_id": "0",
                "amount": "100000"
            },
            "operations": [
                {
                    "type": "withdraw_op_type",
                    "data": {
                        "claim_input_data": "",
                        "amount": "101000",
                        "balance_id": "G5wSdtt5BF3oDXAYXoRJdWnw6Dzr8pc9vC"
                    }
                },
                {
                    "type": "deposit_op_type",
                    "data": {
                        "amount": "100000",
                        "condition": {
                            "asset_id": "0",
                            "slate_id": "0",
                            "type": "withdraw_signature_type",
                            "data": {
                                "owner": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
                            }
                        }
                    }
                }
            ],
            "signatures": [
"205c194f49a821cc7f9884356c89117332c3867967f09d22f0920a2deccf0a64a84c5252b0dde7972dfc905d407a3c8a
            ],
            "expiration": "2016-03-28T08:43:34",
            "gdw_account": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
        }
    ]
}
```

返回参数:

expiration: 交易过期时间

gdw_account: 交易入账方

gdw_inport_asset: 交易入账结构体

operations: 操作结构

signatures: 签名

获取一笔区块链上转帐的详细信息交易

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "transaction_id_prefix",
        "exact"
    ],
    "id": "2",
    "method": "blockchain_get_transaction"
}
```

请求方法：blockchain_get_transaction

请求参数：交易id、 是否精确查找，默认false

Response Result:

```
{
    "id": "2",
    "result": [
        "66294adf274750ba2cd312b13aa81cc0bee71ba6",
        {
            "withdraws": [
                [
                    "0",
                    "100000001000"
                ]
            ],
            "imessage_length": "0",
            "deltas": [
                [
                    "0",
                    [
                        [
                            "0",
                            "-100000001000"
                        ]
                    ]
                ],
                [
                    "1",
                    [
                        [
                            "0",
                            "100000000000"
                        ]
                    ]
                ]
            ],
            "required_fees": {
                "asset_id": "0",
                "amount": "0"
            },
            "alt_fees_paid": {
                "asset_id": "0",
                "amount": "0"
            },
            "deposits": [
                [
                    "0",
                    "100000000000"
                ]
            ],
            "yield": [],
            "trx": {
                "gdw_inport_asset": {
                    "asset_id": "0",
                    "amount": "0"
                },
                "operations": [
                    {
                        "type": "withdraw_op_type",
                        "data": {
                            "claim_input_data": "",
                            "amount": "100000001000",
                            "balance_id": "GMPUsyR4Qiomu2r6GvGNFyNxoZQB7oJXfr"
                        }
                    },
                    {
                        "type": "deposit_op_type",
                        "data": {
                            "amount": "100000000000",
                            "condition": {
                                "asset_id": "0",
                                "slate_id": "0",
                                "type": "withdraw_signature_type",
                                "data": {
                                    "owner": "GGb4Dbh82jY8Q5JrvNmCcsjgdZcawCD5Ex"
                                }
                            }
                        }
                    }
                ],
                "signatures": [

"1f109eaf6f4e95914b6a51bb5c8cbf67992ed3b9d575da7a34ed1f63aa1b0eeea7571ce2cebf18c253d1e4208de1f30f

                ],
                "expiration": "2016-03-22T08:11:00",
                "gdw_account": ""
            },
            "signed_keys": [],
            "chain_location": {
                "trx_num": "0",
                "block_num": "1277"
            },
            "delegate_vote_deltas": [],
            "balance": [
                [
                    "0",
                    "1000"
                ]
            ]
        }
    ]
}
```

返回参数:

trx_id: 交易id

trx: 详细交易数据

signed_keys: 签名

deposits: 入账资产

withdraws: 出账资产

yield: 保留参数

deltas: 保留参数

required_fees: 保留字段

alt_fees_paid: 保留字段

balance: 保留字段

delegate_vote_deltas: 保留参数

imessage_length: 备注长度

chain_location: 交易位置结构体

查询指定范围代理的详细信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "first",
        "count"
    ],
    "id": "26",
    "method": "blockchain_list_delegates"
}
```

请求方法：blockchain_list_delegates

请求参数：起始代理排名序号、 返回条目数

Response Result:

```
{
    "id": "17",
    "result": [
        {
            "owner_key": "G7zcWtxPx252gZhjYmj5nyKKZ6agzkVVc8E4HKBVDiiswZ2syS3",
            "name": "gdw4",
            "last_update": "2015-05-31T16:00:00",
            "delegate_info": {
                "pay_balance": "557579636",
                "total_burned": "0",
                "blocks_missed": "4865",
                "pay_rate": "100",
                "votes_for": "1030973384311",
                "blocks_produced": "5195",
                "last_block_num_produced": "494448",
                "total_paid": "587328636",
                "signing_key_history": [
                    [
                        "0",
                        "G7zcWtxPx252gZhjYmj5nyKKZ6agzkVVc8E4HKBVDiiswZ2syS3"
                    ]
                ],
                "next_secret_hash": "feab43c718df5bb693b3a54b855ecd7bd834c4a7"
            },
            "public_data": "null",
            "registation_date": "2015-05-31T16:00:00",
            "id": "5",
            "active_key_history": [
                [
                    "2015-05-31T16:00:00",
                    "G7zcWtxPx252gZhjYmj5nyKKZ6agzkVVc8E4HKBVDiiswZ2syS3"
                ]
            ]
        }
    ]
}
```

返回参数:

id: 账户ID

name: 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

delegate_info: 代理信息结构体

meta_data: 保留字段

返回指定范围区块的详细信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "max_block_num",
        "limit"
    ],
    "id": "26",
    "method": "blockchain_list_blocks"
}
```

请求方法：blockchain_list_blocks

请求参数：显示到这个区块号为止、 返回条目数

Response Result:

```
{
    "latency": "0",
    "user_transaction_ids": [
        "f7f984386f2cee6cb806e06ab1a5e955c04273a3",
        "f7f8b372d625c99267b5e7e04c42ac004885b102",
        "f7f0d80434c1f7618a084c5423e29bbd2ce49a1e",
        "f7ee970b029dde86ca053ce0976e3bc6112a0c78",
        "f7eac3727f9c4c0d49ee9d8116cfe0e7f5092254",
        "f7eab200cdee190c92a794253318ae533ca901d0",
        "f7e547f8bce59018a8fe504fae182548d560770b",
        "f7e03864045770f8a2eca96b2680a5cd7953a0b4",
        "f7db395033d208ba338391edb9f027c9a396dd48",
        "f7db2bbe686d4eb8682ad828a8f0e7e060e3b523",
        "f7dad29228c81987496edc3b3413cfc825b409e4",
        "f7c6af7fe6f9422b137af95cbf869dff89027cce",
        "f0b3bffd483f2a243502f83b143e1baa90833013",
        "cb896307d7cc7651a3e1f3853b65d254b2958b20",
        "975f8398135d3b90c9662a516d323a2e06b82fd1",
        "876de39d1f96f493788fa946d95a86c330d40669",
        "86ce132074b4735cfa7fdc89b3440527818c7b27",
        "6c56aa35222d8fb891ab0a073a80a14ffaf6360f",
        "6049c81451031be300a9eb5b928f3b9369955cec",
        "534c39ed29294e00e73262b56311cc7f715817a0",
        "48f1ef67e7806cb5d8550cc7bc94fdd9d871ba00",
        "3852885967c249ea3f52b25a71e7b62e3093ffe9",
        "127b092d5df103a3bb5a185d48269befabcb5108",
        "01628269891f5293672b038a3e9f681211650490"
    ],
    "delegate_signature":
"206631beee6020ee31c99c93e16171db60560263acbaa38d81c2cef2a79989d4ea53a019ba35f213a81eb66f07b250c9

    "signee_fees_collected": "0",
    "processing_time": "0",
    "timestamp": "2016-03-11T08:32:20",
    "signee_shares_issued": "0",
    "block_num": "1020",
    "previous_secret": "f0f100f90dc73c2ef9d0bd398e9513d65b2f51a7",
    "next_secret_hash": "b37d87e09b6d112c220a9651db24edd475c9f465",
    "signee_fees_destroyed": "0",
    "transaction_digest": "bdaae5cf29f597a1ec84435377e7e30f351c44abd553f4394dc7febf6b7139ce",
    "random_seed": "0000000000000000000000000000000000000000",
    "block_size": "3886",
    "id": "699a6bd96d6df589a63bc61621ad3a82a8016081",
    "previous": "87f4ffd55d673ebb99d9d377afb957ccb35cb0ac"
}
```

返回参数:

previous: 上一区块的hash

block_num: 当前区块的块号

timestamp: 区块发布时间

transaction_digest: 所有块中交易ID的hash

next_secret_hash: 下一轮出块凭证

previous_secret: 本轮展示出块凭证

delegate_signature: 代理签名

user_transaction_ids: 区块包含所有交易的ID集合

id: 区块的ID

block_size: 区块大小

latency: 保留字段

signee_shares_issued: 保留字段

signee_fees_collected: 保留字段

signee_fees_destroyed: 保留字段

random_seed: 保留字段

processing_time: 保留字段

查询指定区块的签名代理是谁

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "block"
    ],
    "id": "26",
    "method": "blockchain_get_block_signee"
}
```

请求方法：blockchain_get_block_signee

请求参数：区块hash或者区块高度

Response Result:

```
{
    "id": "26",
    "result": "gdw4"
}
```

返回参数:

account_name: 代理名

添加或者删除一个节点的连接

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "node",
        "command"
    ],
    "id": "1",
    "method": "network_add_node"
}
```

请求方法：network_add_node

请求参数：添加或者删除的ip和端口号、 对于该节点的操作方式

Response Result:

```
{
    "id": "1",
    "result": "null"
}
```

返回参数:

无:

获取节点当前连接数

调用前提：无

Request:

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

请求方法：network_get_connection_count

请求参数：无

Response Result:

```
{
    "id": "1",
    "result": "1"
}
```

返回参数:

result冒号后的字段: 当前节点连接数

传回每一个已经连接的节点信息

调用前提：返回的是一组数据

Request:

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

请求方法：network_get_peer_info

请求参数：是否只输出防火墙之后的节点，默认false

Response Result:

```
{
    "id": "2",
    "result": [
        {
            "lastsend": "1466573207",
            "gdw_git_revision_sha": "1.2 (same as ours)",
            "conntime": "2016-06-22T05:22:33",
            "addrlocal": "10.23.3.161:60555",
            "gdw_git_revision_age": "1 years6 months ago (same as ours)",
            "fc_git_revision_age": "46 years ago (same as ours)",
            "current_head_block_number": "689",
            "fc_git_revision_sha": "0 (same as ours)",
            "pingwait": "",
            "current_head_block": "1dea351ba0d6fc22958ebc1e45ba9c68c7e20333",
            "addr": "119.254.161.28:60555",
            "gdw_git_revision_unix_timestamp": "2015-01-08T18:02:32",
            "lastrecv": "1466573207",
            "platform": "win32",
            "firewall_status": "firewalled",
            "pingtime": "",
            "bytesrecv": "7696",
            "bytessent": "2288",
            "startingheight": "",
            "inbound": false,
            "version": "",
            "current_head_block_time": "2016-06-22T05:26:50",
            "services": "00000001",
            "sgdwver": "gdw_client",
            "syncnode": "",
            "fc_git_revision_unix_timestamp": "1970-01-01T00:00:00",
            "banscore": ""
        }
    ]
}
```

返回参数:

addr: 连接的节点的ip和端口号

addrlocal: 本地节点的ip和端口号

services: 保留字段

lastsend: 最后一次发送时间

lastrecv: 最后一次接收时间

bytessent: 发送字节数

bytesrecv: 接受字节数

conntime: 连接时间

pingtime: 保留字段

pingwait: 保留字段

version: 版本

sgdwver: 保留字段

inbound: 保留字段

firewall_status: 防火墙状态

startingheight: 保留字段

banscore: 保留字段

syncnode: 保留字段

gdw_git_revision_sha: git版本

gdw_git_revision_unix_timestamp: git时间戳

gdw_git_revision_age: git版本时间长度

fc_git_revision_sha: fc库的版本

fc_git_revision_unix_timestamp: fc库的git时间戳

fc_git_revision_age: fc库的git版本时间长度

platform: 平台

current_head_block_number: 当前块的高度

current_head_block: 当前块的hash

current_head_block_time: 当前块的时间

获取当前网络信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "26",
    "method": "network_get_info"
}
```

请求方法：network_get_info

请求参数：无

Response Result:

```
{
    "id": "26",
    "result": {
        "listening_on": "0.0.0.0:60696",
        "node_public_key":
"03dafc378c8a22d136b338a3913bb0acbe551f970d0af1d0955861a7103eb17252",
        "node_id": "6819fe268e09df3ce4a90ab543fb98cc42390a986e5f5037921e911b437b06b23b",
        "firewalled": "firewalled"
    }
}
```

返回参数:

listening_on: 客户端监听的地址和端口

node_public_key: 节点与节点之间加密通讯使用的公钥（程序第一次启动创建）

node_id: 每次启动客户端都会重新生成一个，为了可以多次连接同一个节点

firewalled: 是否在防火墙后面

获取潜在连接点列表

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "26",
    "method": "network_list_potential_peers"
}
```

请求方法：network_list_potential_peers

请求参数：无

Response Result:

```
{
    "id": "26",
    "result": [
        {
            "endpoint": "59.108.127.182:60666",
            "last_connection_attempt_time": "2016-06-22T02:17:47",
            "last_error": {
                "message": "unspecified",
                "code": "0",
                "name": "exception",
                "stack": [
                    {
                        "data": {
                            "message": "?????????????(Э??/??????/???)??????????P?"
                        },
                        "context": {
                            "thread_name": "asio",
                            "level": "error",
                            "timestamp": "2016-06-21T04:16:03",
                            "hostname": "",
                            "file": "asio.cpp",
                            "line": "60",
                            "method": "fc::asio::detail::error_handler"
                        },
                        "format": "${message} "
                    }
                ]
            },
            "number_of_failed_connection_attempts": "5",
            "last_seen_time": "2016-06-22T05:18:17",
            "last_connection_disposition": "last_connection_succeeded",
            "number_of_successful_connection_attempts": "10"
        },
        {
            "endpoint": "119.254.161.28:60666",
            "last_connection_attempt_time": "2016-06-22T02:17:49",
            "last_error": {
                "message": "unspecified",
                "code": "0",
                "name": "exception",
                "stack": [
                    {
                        "data": {
                            "peer": "119.254.161.28:60666",
                            "status": "connecting",
                            "timeout": "10",
                            "sent": "0",
                            "received": "0"
                        },
                        "context": {
                            "thread_name": "p2p",
                            "level": "warn",
                            "timestamp": "2016-06-20T06:55:21",
                            "hostname": "",
                            "file": "Node.cpp",
                            "line": "1256",
                            "method":
"gdw::net::detail::NodeImpl::terminate_inactive_connections_loop"
                        },
                        "format": "Terminating handshaking connection due to inactivity of
${timeout} seconds.  Negotiating status: ${status}, bytes sent: ${sent}, bytes received:
${received}"
                    }
                ]
            },
            "number_of_failed_connection_attempts": "5",
            "last_seen_time": "2016-06-22T05:18:17",
            "last_connection_disposition": "last_connection_succeeded",
            "number_of_successful_connection_attempts": "10"
        }
    ]
}
```

返回参数:

endpoint: 潜在节点IP

last_seen_time: 上一次节点出现时间

last_connection_disposition: 上次连接的状态

last_connection_attempt_time: 上次尝试连接时间

number_of_successful_connection_attempts: 成功连接的数量

number_of_failed_connection_attempts: 失败连接的数量

last_error: 保留字段

查询UPNP基本信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "26",
    "method": "network_get_upnp_info"
}
```

请求方法：network_get_upnp_info

请求参数：无

Response Result:

```
{
    "id": "26",
    "result": {
        "external_ip": "0.0.0.0",
        "upnp_enabled": true,
        "mapped_port": "0"
    }
}
```

返回参数:

upnp_enabled: 是否启动upnp

external_ip: 本地监听IP

mapped_port: 映射端口号

以指定名称创建钱包

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "wallet_name",
        "password",
        ""
    ],
    "id": "3",
    "method": "wallet_create"
}
```

请求方法：wallet_create

请求参数：创建钱包名称、 创建密码

Response Result:

```
{
    "id": "1",
    "result": "null"
}
```

返回参数:

无:

获取钱包信息

调用前提：无

Request:

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

请求方法：wallet_get_info

请求参数：无

Response Result:

```
{
    "id": "1",
    "result": {
        "scan_progress": "1.00000000000000000",
        "data_dir": "e:/gdwtest/3/wallets",
        "name": "op",
        "unlocked": true,
        "last_scanned_block_num": null,
        "open": true,
        "transaction_expiration_secs": "3600",
        "unlocked_until_timestamp": "2016-03-28T07:29:49",
        "automatic_backups": true,
        "version": "109",
        "transaction_scanning_enabled": false,
        "transaction_fee": {
            "asset_id": "0",
            "amount": "1000"
        },
        "num_scanning_threads": "4",
        "unlocked_until": "3598",
        "last_scanned_block_timestamp": null
    }
}
```

返回参数:

data_dir: 文件路径

num_scanning_threads: 扫描线程数

open: 钱包是否打开

name: 钱包名称

automatic_backups: 是否开启自动备份

transaction_scanning_enabled: 交易扫描是否开启

last_scanned_block_num: 上一个扫描块号

last_scanned_block_timestamp: 上一个扫描块时间

transaction_fee: 钱包默认交易手续费

transaction_expiration_secs: 经多少时间交易过期

unlocked: 钱包是否解锁

unlocked_until: 钱包解锁状态所剩的秒数

unlocked_until_timestamp: 钱包重新锁上的时间

scan_progress: 扫描进程

version: 钱包版本

如果目前的钱包是打开的就关闭它

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "6",
    "method": "wallet_close"
}
```

请求方法：wallet_close

请求参数：无

Response Result:

```
{
    "id": "6",
    "result": "null"
}
```

返回参数:

无:

打开指定名称的钱包

调用前提：当前指定名称的钱包名称已经存在

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "wallet_name"
    ],
    "id": "2",
    "method": "wallet_open"
}
```

请求方法：wallet_open

请求参数：输入钱包名称

Response Result:

```
{
    "id": "6",
    "result": "null"
}
```

返回参数:

无:

解锁钱包内的私钥以启用支付操作

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "timeout",
        "password"
    ],
    "id": "17",
    "method": "wallet_unlock"
}
```

请求方法：wallet_unlock

请求参数：解锁过期时间、 密码

Response Result:

```
{
    "id": "4",
    "result": "op"
}
```

返回参数:

无:

将私钥导入本地钱包中, 并传回其所导入的真实帐号

调用前提：钱包打开且钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "wif_key_to_import",
        "account_name",
        "create_account",
        "wallet_rescan_blockchain"
    ],
    "id": "5",
    "method": "wallet_import_private_key"
}
```

请求方法：wallet_import_private_key

请求参数：要导入的私钥、 导入账户的名称、 是否需要创建新账户、 是否重新扫描新钱包

Response Result:

```
{
    "id": "5",
    "result": "aa"
}
```

返回参数:

result冒号后的字段: 导入账户名称

将当前的钱包数据导出为一个JSON 文件

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "filename"
    ],
    "id": "7",
    "method": "wallet_backup_create"
}
```

请求方法：wallet_backup_create

请求参数：备份存储路径

Response Result:

```
{
    "id": "6",
    "result": "null"
}
```

返回参数:

无:

用备份的JSON 文件创建(复原)一个新的钱包

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "filepath",
        "wallet_name",
        "password"
    ],
    "id": "8",
    "method": "wallet_backup_restore"
}
```

请求方法：wallet_backup_restore

请求参数：钱包备份文件的路径、 钱包名称、 备份钱包的密码

Response Result:

```
{
    "id": "8",
    "result": "null"
}
```

返回参数:

无:

设置钱包是否开启自动备份功能并返回设置后的状态

调用前提：钱包打开，当你创建账户时候，自动备份如果开启，就会自动创建一个备份

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "enabled"
    ],
    "id": "9",
    "method": "wallet_set_automatic_backups"
}
```

请求方法：wallet_set_automatic_backups

请求参数：是否开启自动备份功能

Response Result:

```
{
    "id": "9",
    "result": false
}
```

返回参数:

result冒号后的字段: 钱包当前备份是否自动开启的状态

设置交易过期时间并返回设置后的过期时间

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "per_sec"
    ],
    "id": "10",
    "method": "wallet_set_transaction_expiration_time"
}
```

请求方法：wallet_set_transaction_expiration_time

请求参数：超时时间以秒计数

Response Result:

```
{
    "id": "10",
    "result": "3600"
}
```

返回参数:

result冒号后的字段: 当前交易的超时时间

列出指定帐号的交易历史

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name",
        "asset_symbol",
        "limit",
        "start_block_num",
        "end_block_num"
    ],
    "id": "11",
    "method": "wallet_account_transaction_history"
}
```

请求方法：wallet_account_transaction_history

请求参数：账户名称默认值是空、 资产标示默认值是空、 返回的交易数量上限默认值是0，0代表没有限制、 开始查询的block编号，默认值为0、 结束的block编号默认值为-1，代表着返回到当前块的交易

Response Result:

```
{
    "id": "26",
    "result": [
        {
            "is_market": false,
            "fee": {
                "asset_id": "0",
                "amount": "1000"
            },
            "is_confirmed": true,
            "is_market_cancel": false,
            "timestamp": "2016-06-21T03:20:20",
            "is_virtual": false,
            "block_num": "462950",
            "ledger_entries": [
                {
                    "from_account": "gdw6",
                    "amount": {
                        "asset_id": "0",
                        "amount": "10000000000"
                    },
                    "running_balances": [
                        [
                            "sy234",
                            [
                                [
                                    "0",
                                    {
                                        "asset_id": "0",
                                        "amount": "10000000000"
                                    }
                                ]
                            ]
                        ]
                    ],
                    "memo": "",
                    "to_account": "sy234"
                }
            ],
            "trx_id": "78d148258f0488e12c7ad59d02a5700cd7cf194a",
            "expiration_timestamp": "2016-06-21T04:20:06"
        }
    ]
}
```

返回参数:

is_virtual: 保留字段

is_confirmed: 交易是否被确认

is_market: 保留字段

is_market_cancel: 保留字段

trx_id: 交易单号

block_num: 交易被确认的区块

ledger_entries: 交易解析

fee: 交易支付的手续费

timestamp: 交易创建时间

expiration_timestamp: 交易失效时间

根据出入账查询交易记录

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name",
        "asset_symbol",
        "limit",
        "transaction_type"
    ],
    "id": "26",
    "method": "wallet_transaction_history_splite"
}
```

请求方法：wallet_transaction_history_splite

请求参数：用户名、 资产标识、 条数限制、 交易类型

Response Result:

```
{
    "id": "26",
    "result": [
        {
            "is_market": false,
            "fee": {
                "asset_id": "0",
                "amount": "1000"
            },
            "is_confirmed": true,
            "is_market_cancel": false,
            "timestamp": "2016-06-21T03:20:20",
            "is_virtual": false,
            "block_num": "462950",
            "ledger_entries": [
                {
                    "from_account": "gdw6",
                    "amount": {
                        "asset_id": "0",
                        "amount": "10000000000"
                    },
                    "running_balances": [
                        [
                            "sy234",
                            [
                                [
                                    "0",
                                    {
                                        "asset_id": "0",
                                        "amount": "10000000000"
                                    }
                                ]
                            ]
                        ]
                    ],
                    "memo": "",
                    "to_account": "sy234"
                }
            ],
            "trx_id": "78d148258f0488e12c7ad59d02a5700cd7cf194a",
            "expiration_timestamp": "2016-06-21T04:20:06"
        }
    ]
}
```

返回参数:

is_virtual: 保留字段

is_confirmed: 交易是否被确认

is_market: 保留字段

is_market_cancel: 保留字段

trx_id: 交易单号

block_num: 交易被确认的区块

ledger_entries: 交易解析

fee: 交易支付的手续费

timestamp: 交易创建时间

expiration_timestamp: 交易失效时间

获取当前pending区所有错误记录

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "filepath"
    ],
    "id": "26",
    "method": "wallet_get_pending_transaction_errors"
}
```

请求方法：wallet_get_pending_transaction_errors

请求参数：文件路径

Response Result:

```
{
    "id": "26",
    "result": [
        [
            "d128986f41ada51d4937171047acc33781c8f321",
            {
                "message": "expired transaction",
                "code": "31010",
                "name": "expired_transaction",
                "stack": [
                    {
                        "data": {
                            "_current_state->now()": "2016-06-23T09:07:50",
                            "trx_arg": {
                                "gdw_inport_asset": {
                                    "asset_id": "0",
                                    "amount": "0"
                                },
                                "operations": [
                                    {
                                        "type": "withdraw_op_type",
                                        "data": {
                                            "claim_input_data": "",
                                            "amount": "101000",
                                            "balance_id": "G5Bf23THE4RoWzQqWAKn7PkmzJdgUWHDJV"
                                        }
                                    },
                                    {
                                        "type": "deposit_op_type",
                                        "data": {
                                            "amount": "100000",
                                            "condition": {
                                                "asset_id": "0",
                                                "slate_id": "0",
                                                "type": "withdraw_signature_type",
                                                "data": {
                                                    "owner":
"GM3rr9yc2MB5GQRdBVY2Q5baF1FRBU8gG6"
                                                }
                                            }
                                        }
                                    }
                                ],
                                "signatures": [

"2044d58b6e469f0f075a2faade9d806477aaa134f749a1b9bce25e3455ab8b9f6015bcaa40cf74276fdcf084c6c76dc7

                                ],
                                "expiration": "2016-06-23T09:03:43",
                                "gdw_account": ""
                            },
                            "expired_by_sec": "247"
                        },
                        "context": {
                            "thread_name": "th_a",
                            "level": "error",
                            "timestamp": "2016-06-23T09:07:51",
                            "hostname": "",
                            "file": "TransactionEvaluationState.cpp",
                            "line": "152",
                            "method": "gdw::blockchain::TransactionEvaluationState::evaluate"
                        },
                        "format": ""
                    },
                    {
                        "data": {
                            "trx_arg": {
                                "gdw_inport_asset": {
                                    "asset_id": "0",
                                    "amount": "0"
                                },
                                "operations": [
                                    {
                                        "type": "withdraw_op_type",
                                        "data": {
                                            "claim_input_data": "",
                                            "amount": "101000",
                                            "balance_id": "G5Bf23THE4RoWzQqWAKn7PkmzJdgUWHDJV"
                                        }
                                    },
                                    {
                                        "type": "deposit_op_type",
                                        "data": {
                                            "amount": "100000",
                                            "condition": {
                                                "asset_id": "0",
                                                "slate_id": "0",
                                                "type": "withdraw_signature_type",
                                                "data": {
                                                    "owner":
"GM3rr9yc2MB5GQRdBVY2Q5baF1FRBU8gG6"
                                                }
                                            }
                                        }
                                    }
                                ],
                                "signatures": [

"2044d58b6e469f0f075a2faade9d806477aaa134f749a1b9bce25e3455ab8b9f6015bcaa40cf74276fdcf084c6c76dc7

                                ],
                                "expiration": "2016-06-23T09:03:43",
                                "gdw_account": ""
                            }
                        },
                        "context": {
                            "thread_name": "th_a",
                            "level": "warn",
                            "timestamp": "2016-06-23T09:07:51",
                            "hostname": "",
                            "file": "TransactionEvaluationState.cpp",
                            "line": "192",
                            "method": "gdw::blockchain::TransactionEvaluationState::evaluate"
                        },
                        "format": ""
                    }
                ]
            }
        ],
        [
            "ecc6cfb278e2f10e0d77f122fa840f2160dedcc4",
            {
                "message": "expired transaction",
                "code": "31010",
                "name": "expired_transaction",
                "stack": [
                    {
                        "data": {
                            "_current_state->now()": "2016-06-23T09:07:50",
                            "trx_arg": {
                                "gdw_inport_asset": {
                                    "asset_id": "0",
                                    "amount": "0"
                                },
                                "operations": [
                                    {
                                        "type": "withdraw_op_type",
                                        "data": {
                                            "claim_input_data": "",
                                            "amount": "101000",
                                            "balance_id": "G5Bf23THE4RoWzQqWAKn7PkmzJdgUWHDJV"
                                        }
                                    },
                                    {
                                        "type": "deposit_op_type",
                                        "data": {
                                            "amount": "100000",
                                            "condition": {
                                                "asset_id": "0",
                                                "slate_id": "0",
                                                "type": "withdraw_signature_type",
                                                "data": {
                                                    "owner":
```

```
"GM3rr9yc2MB5GQRdBVY2Q5baF1FRBU8gG6"
                                                }
                                            }
                                        }
                                    }
                                ],
                                "signatures": [

"1f4892d2720a119b861560422ed5bab49875e8d8a4cdca3fe40ab51e54ec9449c712fc54c34bc90976e94146975601e8

                                ],
                                "expiration": "2016-06-23T09:03:34",
                                "gdw_account": ""
                            },
                            "expired_by_sec": "256"
                        },
                        "context": {
                            "thread_name": "th_a",
                            "level": "error",
                            "timestamp": "2016-06-23T09:07:51",
                            "hostname": "",
                            "file": "TransactionEvaluationState.cpp",
                            "line": "152",
                            "method": "gdw::blockchain::TransactionEvaluationState::evaluate"
                        },
                        "format": ""
                    },
                    {
                        "data": {
                            "trx_arg": {
                                "gdw_inport_asset": {
                                    "asset_id": "0",
                                    "amount": "0"
                                },
                                "operations": [
                                    {
                                        "type": "withdraw_op_type",
                                        "data": {
                                            "claim_input_data": "",
                                            "amount": "101000",
                                            "balance_id": "G5Bf23THE4RoWzQqWAKn7PkmzJdgUWHDJV"
                                        }
                                    },
                                    {
                                        "type": "deposit_op_type",
                                        "data": {
                                            "amount": "100000",
                                            "condition": {
                                                "asset_id": "0",
                                                "slate_id": "0",
                                                "type": "withdraw_signature_type",
                                                "data": {
                                                    "owner":
"GM3rr9yc2MB5GQRdBVY2Q5baF1FRBU8gG6"
                                                }
                                            }
                                        }
                                    }
                                ],
                                "signatures": [

"1f4892d2720a119b861560422ed5bab49875e8d8a4cdca3fe40ab51e54ec9449c712fc54c34bc90976e94146975601e8

                                ],
                                "expiration": "2016-06-23T09:03:34",
                                "gdw_account": ""
                            }
                        },
                        "context": {
                            "thread_name": "th_a",
                            "level": "warn",
                            "timestamp": "2016-06-23T09:07:51",
                            "hostname": "",
                            "file": "TransactionEvaluationState.cpp",
                            "line": "192",
                            "method": "gdw::blockchain::TransactionEvaluationState::evaluate"
                        },
                        "format": ""
                    }
                ]
            }
        ]
    ]
}
```

返回参数:

transactionID: 交易单号

Exception: 保留字段（交易异常）

修改密码

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "old_passphrase",
        "passphrase"
    ],
    "id": "26",
    "method": "wallet_change_passphrase"
}
```

请求方法：wallet_change_passphrase

请求参数：钱包旧密码、 钱包新密码

Response Result:

```
{
    "id": "26",
    "result": "null"
}
```

返回参数:

无:

判断密码是否正确

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "123456789"
    ],
    "id": "19",
    "method": "wallet_check_passphrase"
}
```

请求方法：wallet_check_passphrase

请求参数：待校验密码

Response Result:

```
{
    "id": "19",
    "result": true
}
```

返回参数:

result: 是否为正确的密码

判断一个地址是否合法

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
    ],
    "id": "20",
    "method": "wallet_check_address"
}
```

请求方法：wallet_check_address

请求参数：待校验的地址

Response Result:

```
{
    "id": "19",
    "result": true
}
```

返回参数:

result: 是否正确的地址

创建一个账户

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "test"
    ],
    "id": "22",
    "method": "wallet_account_create"
}
```

请求方法：wallet_account_create

请求参数：待创建的用户名

Response Result:

```
{
    "id": "22",
    "result": "GQJDzx3AzQWGhCBucE6zLj3mkFo6KG9qSw"
}
```

返回参数:

addr: 创建账户地址

设置支持账户

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "gdw0",
        "1"
    ],
    "id": "23",
    "method": "wallet_account_set_approval"
}
```

请求方法：wallet_account_set_approval

请求参数：支持账户、 是否同意

Response Result:

```
{
    "id": "3",
    "result": "1"
}
```

返回参数:

result: 是否支持

无:

无:

无:

无:

无:

无:

转账到某个地址

调用前提：钱包解锁，钱包内有账户

Request:

```
{
    "jsonrpc": 2.0,
    "params": [
        "amount_to_transfer",
        "asset_symbol",
        "from_account_name",
        "to_address",
        "memo_message",
        "strategy"
    ],
    "id": "1",
    "method": "wallet_transfer_to_address"
}
```

请求方法：wallet_transfer_to_address

请求参数：转账金额、 转账资产类型、 取钱账户、 转账到账户的地址、 备注信息、 投票状态（vote_none不投票，vote_all投所有人最多108人,vote_radom随机投票，从支持者中随机选取一定的人数进行投票最多不超过36人， vote_recommended根据已经选择的投票人进行投票，如果选择的投票人的pgdwlish_data中有其他投票策略加入到自己的投票策略中）

Response Result:

```
{
    "id": "26",
    "result": {
        "index": "0",
        "is_market": false,
        "fee": {
            "asset_id": "0",
            "amount": "1000"
        },
        "is_confirmed": false,
        "ledger_entries": [
            {
                "from_account": "G5eLFrZG5gvc5g5xqEU6J3GELp5WcQYoqhz4JBPT7BVgTpqqxbN",
                "amount": {
                    "asset_id": "0",
                    "amount": "100000"
                },
                "memo": "aa"
            }
        ],
        "received_time": "2016-06-01T08:32:20",
        "is_virtual": false,
        "block_num": "0",
        "trx": {
            "gdw_inport_asset": {
                "asset_id": "0",
                "amount": "0"
            },
            "operations": [
                {
                    "type": "withdraw_op_type",
                    "data": {
                        "claim_input_data": "",
                        "amount": "101000",
                        "balance_id": "GJEk1wuT9Vix4vQRidmAafz5hCWrS48un2"
                    }
                },
                {
                    "type": "deposit_op_type",
                    "data": {
                        "amount": "100000",
                        "condition": {
                            "asset_id": "0",
                            "slate_id": "0",
                            "type": "withdraw_signature_type",
                            "data": {
                                "owner": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
                            }
                        }
                    }
                },
                {
                    "type": "imessage_memo_op_type",
                    "data": {
                        "imessage": "aa"
                    }
                }
            ],
            "signatures": [

"203abaffeaa2735b4f9c79bedae77143ac5da1451c92fd7390c1ae502b53578ef47b2751afa034ff7e0232e37f8fb755

            ],
            "expiration": "2016-06-01T09:32:19",
            "gdw_account": ""
        },
        "entry_id": "0ba68a7e9945f9119cb151a2ef5ff110e4d15925",
        "created_time": "2016-06-01T08:32:20",
        "extra_addresses": [
            "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
        ]
    }
}
```

返回参数:

index: 保留字段

entry_id: 交易单号

block_num: 块号

is_virtual: 保留字段

is_confirmed: 是否确认

is_market: 保留字段

trx: 结构体类型名

转账到某个账户

调用前提：钱包解锁，钱包内有账户

Request:

```
{
    "jsonrpc": 2.0,
    "params": [
        "amount_to_transfer",
        "asset_symbol",
        "from_account_name",
        "to_account_name",
        "memo_message",
        "strategy"
    ],
    "id": "1",
    "method": "wallet_transfer_to_public_account"
}
```

请求方法：wallet_transfer_to_public_account

请求参数：转账金额、 转账资产类型、 取钱账户、 转账目的账户、 备注信息、 投票状态（vote_none不投票，vote_all投所有人最多108人,vote_random随机投票，从支持者中随机选取一定的人数进行投票最多不超过36人， vote_recommended根据已经选择的投票人进行投票，如果选择的投票人的pgdwlish_data中有其他投票策略加入到自己的投票策略中）

Response Result:

```
{
    "id": "26",
    "result": {
        "index": "0",
        "is_market": false,
        "fee": {
            "asset_id": "0",
            "amount": "1000"
        },
        "is_confirmed": false,
        "ledger_entries": [
            {
                "from_account": "G5eLFrZG5gvc5g5xqEU6J3GELp5WcQYoqhz4JBPT7BVgTpqqxbN",
                "amount": {
                    "asset_id": "0",
                    "amount": "100000"
                },
                "memo": "aa"
            }
        ],
        "received_time": "2016-06-01T08:32:20",
        "is_virtual": false,
        "block_num": "0",
        "trx": {
            "gdw_inport_asset": {
                "asset_id": "0",
                "amount": "0"
            },
            "operations": [
                {
                    "type": "withdraw_op_type",
                    "data": {
                        "claim_input_data": "",
                        "amount": "101000",
                        "balance_id": "GJEk1wuT9Vix4vQRidmAafz5hCWrS48un2"
                    }
                },
                {
                    "type": "deposit_op_type",
                    "data": {
                        "amount": "100000",
                        "condition": {
                            "asset_id": "0",
                            "slate_id": "0",
                            "type": "withdraw_signature_type",
                            "data": {
                                "owner": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
                            }
                        }
                    }
                },
                {
                    "type": "imessage_memo_op_type",
                    "data": {
                        "imessage": "aa"
                    }
                }
            ],
            "signatures": [

"203abaffeaa2735b4f9c79bedae77143ac5da1451c92fd7390c1ae502b53578ef47b2751afa034ff7e0232e37f8fb755

            ],
            "expiration": "2016-06-01T09:32:19",
            "gdw_account": ""
        },
        "entry_id": "0ba68a7e9945f9119cb151a2ef5ff110e4d15925",
        "created_time": "2016-06-01T08:32:20",
        "extra_addresses": [
            "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
        ]
    }
}
```

返回参数:

index: 保留字段

entry_id: 交易单号

block_num: 块号

is_virtual: 保留字段

is_confirmed: 是否确认

is_market: 保留字段

trx: 结构体类型名

重新扫描区块，从中提取相关交易到钱包

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "29",
    "method": "wallet_rescan_blockchain"
}
```

请求方法：wallet_rescan_blockchain

请求参数：无

Response Result:

```
{
    "id": "29",
    "result": "null"
}
```

返回参数:

无:

取消当前区块的扫描操作

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "30",
    "method": "wallet_cancel_scan"
}
```

请求方法：wallet_cancel_scan

请求参数：无

Response Result:

```
{
    "id": "30",
    "result": "null"
}
```

返回参数:

无:

查询交易信息

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "transaction_id_prefix"
    ],
    "id": "26",
    "method": "wallet_get_transaction"
}
```

请求方法：wallet_get_transaction

请求参数：交易单号

Response Result:

```
{
    "id": "26",
    "result": {
        "index": "24",
        "is_market": false,
        "fee": {
            "asset_id": "0",
            "amount": "1000"
        },
        "is_confirmed": true,
        "ledger_entries": [
            {
                "from_account": "G7zZSxzHsgSzmcdcMsYV9DZqAro71WrL5HnePJH5YbdQGfJRFKy",
                "amount": {
                    "asset_id": "0",
                    "amount": "100000"
                },
                "memo": "To: G8XLL7...",
                "to_account": "G7VpiEMGDW9LSYxk3APqJg7PPi256M8TSC9vzCEQhokjYYWHQmKx"
            }
        ],
        "received_time": "2016-06-28T01:42:26",
        "is_virtual": false,
        "block_num": "477920",
        "trx": {
            "gdw_inport_asset": {
                "asset_id": "0",
                "amount": "0"
            },
            "operations": [
                {
                    "type": "withdraw_op_type",
                    "data": {
                        "claim_input_data": "",
                        "amount": "101000",
                        "balance_id": "G5Bf23THE4RoWzQqWAKn7PkmzJdgUWHDJV"
                    }
                },
                {
                    "type": "deposit_op_type",
                    "data": {
                        "amount": "100000",
                        "condition": {
                            "asset_id": "0",
                            "slate_id": "0",
                            "type": "withdraw_signature_type",
                            "data": {
                                "owner": "G8XLL78anJzHw1qeMqzHsq5Hh447wuvYro"
                            }
                        }
                    }
                }
            ],
            "signatures": [

"1f34cfc2598fe247a81926b7814d0a8bb1929bd9d3278143b7422fe334677a19d851f6c4e71792d60c15c8b227f93717

            ],
            "expiration": "2016-06-28T01:42:28",
            "gdw_account": ""
        },
        "entry_id": "58bae775e828049429e3a26b7e625072d63da374",
        "created_time": "2016-06-28T01:42:26",
        "extra_addresses": [
            "G8XLL78anJzHw1qeMqzHsq5Hh447wuvYro"
        ]
    }
}
```

返回参数:

index: 保留字段

entry_id: 交易单号

block_num: 块号

is_virtual: 保留字段

is_confirmed: 是否确认

is_market: 保留字段

trx: 结构体类型名

账号注册

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name",
        "pay_from_account",
        "public_data",
        "delegate_pay_rate",
        "account_type"
    ],
    "id": "26",
    "method": "wallet_account_register"
}
```

请求方法：wallet_account_register

请求参数：要注册的账户名、 支付本次注册费用的账户、 公开数据、 代理受托率、 账户类型(保留字段)

Response Result:

```
{
    "id": "26",
    "result": {
        "index": "0",
        "is_market": false,
        "fee": {
            "asset_id": "0",
            "amount": "1000"
        },
        "is_confirmed": false,
        "ledger_entries": [
            {
                "from_account": "G6e57Ms177rEGt4Db8rmjYMFveXQYVunsYjjmvma73tAg1XsHxx",
                "amount": {
                    "asset_id": "0",
                    "amount": "0"
                },
                "memo": "register sy234",
                "to_account": "G6e57Ms177rEGt4Db8rmjYMFveXQYVunsYjjmvma73tAg1XsHxx"
            }
        ],
        "received_time": "2016-06-28T02:56:56",
        "is_virtual": false,
        "block_num": "0",
        "trx": {
            "gdw_inport_asset": {
                "asset_id": "0",
                "amount": "0"
            },
            "operations": [
                {
                    "type": "register_account_op_type",
                    "data": {
                        "owner_key": "G6e57Ms177rEGt4Db8rmjYMFveXQYVunsYjjmvma73tAg1XsHxx",
                        "name": "sy234",
                        "active_key": "G7zZSxzHsgSzmcdcMsYV9DZqAro71WrL5HnePJH5YbdQGfJRFKy",
                        "meta_data": {
                            "type": "titan_account",
                            "data": ""
                        },
                        "public_data": "",
                        "delegate_pay_rate": "255"
                    }
                },
                {
                    "type": "withdraw_op_type",
                    "data": {
                        "claim_input_data": "",
                        "amount": "1000",
                        "balance_id": "G5Bf23THE4RoWzQqWAKn7PkmzJdgUWHDJV"
                    }
                }
            ],
            "signatures": [

"204f4cb30eef8da006f74eb81a3991a45882117a63d9847b21cd992c82543f0a60647a0a2b4269c4baac2eb8c41d14d0

            ],
            "expiration": "2016-06-28T02:56:58",
            "gdw_account": ""
        },
        "entry_id": "d29cee8715b8c53001a7f0b4b1f30eea781090ca",
        "created_time": "2016-06-28T02:56:56",
        "extra_addresses": []
    }
}
```

返回参数:

index: 保留字段

entry_id: 交易单号

block_num: 块号

is_virtual: 保留字段

is_confirmed: 是否确认

is_market: 保留字段

trx: 结构体类型名

查询钱包内所有相关账户信息

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "26",
    "method": "wallet_list_accounts"
}
```

请求方法：wallet_list_accounts

请求参数：无

Response Result:

```
{
    "id": "26",
    "result": [
        {
            "index": "25",
            "owner_key": "G6pZUWPujhRUEPaoSBuKgwPVY3Ay3VYwfLZ33EnSreCLWRHyZ6M",
            "name": "gdw1",
            "block_production_enabled": false,
            "registration_date": "2015-05-31T16:00:00",
            "last_update": "2015-05-31T16:00:00",
            "private_data": null,
            "delegate_info": {
                "pay_balance": "71063974",
                "total_burned": "0",
                "blocks_missed": "4481",
                "pay_rate": "100",
                "votes_for": "1136147884",
                "blocks_produced": "5045",
                "last_block_num_produced": "479440",
                "total_paid": "571302974",
                "signing_key_history": [
                    [
                        "0",
                        "G6pZUWPujhRUEPaoSBuKgwPVY3Ay3VYwfLZ33EnSreCLWRHyZ6M"
                    ]
                ],
                "next_secret_hash": "abe20661166e5e00a583d2dccdc23a25a706b190"
            },
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": null,
            "id": "2",
            "is_my_account": false,
            "approved": "1",
            "active_key_history": [
                [
                    "2015-05-31T16:00:00",
                    "G6pZUWPujhRUEPaoSBuKgwPVY3Ay3VYwfLZ33EnSreCLWRHyZ6M"
                ]
            ]
        },
        {
            "index": "27",
            "owner_key": "G7d6PWo9bJWbGkv1naudu9CCm8jvvpcVNevE8nXS4QmrnHTCRbn",
            "name": "gdw2",
            "block_production_enabled": false,
            "registration_date": "2015-05-31T16:00:00",
            "last_update": "2015-05-31T16:00:00",
            "private_data": null,
            "delegate_info": {
                "pay_balance": "539997745",
                "total_burned": "0",
                "blocks_missed": "4499",
                "pay_rate": "100",
                "votes_for": "1436524291",
                "blocks_produced": "5029",
                "last_block_num_produced": "479448",
                "total_paid": "569642745",
                "signing_key_history": [
                    [
                        "0",
                        "G7d6PWo9bJWbGkv1naudu9CCm8jvvpcVNevE8nXS4QmrnHTCRbn"
                    ]
                ],
                "next_secret_hash": "b1b022fb4831605311d5a96275332ef023f8f7fc"
            },
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": null,
            "id": "3",
            "is_my_account": false,
            "approved": "1",
            "active_key_history": [
                [
                    "2015-05-31T16:00:00",
                    "G7d6PWo9bJWbGkv1naudu9CCm8jvvpcVNevE8nXS4QmrnHTCRbn"
                ]
            ]
        },
        {
            "index": "14",
            "owner_key": "G6e57Ms177rEGt4Db8rmjYMFveXQYVunsYjjmvma73tAg1XsHxx",
            "name": "sy234",
            "block_production_enabled": false,
            "registration_date": "2016-06-28T02:56:50",
            "meta_data": {
                "type": "titan_account",
                "data": ""
            },
            "last_update": "2016-06-28T02:56:50",
            "private_data": null,
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": "",
            "id": "177",
            "is_my_account": true,
            "approved": "0",
            "active_key_history": [
                [
                    "2016-06-28T02:56:50",
                    "G7zZSxzHsgSzmcdcMsYV9DZqAro71WrL5HnePJH5YbdQGfJRFKy"
                ]
            ]
        },
        {
            "index": "33",
            "owner_key": "G5aRA9JVFD1R8TvLJpKzjXrJUCG8Wg6BWKLnHpSYpGH9qPicPMd",
            "name": "sy345",
            "block_production_enabled": false,
            "registration_date": "2016-06-28T03:59:40",
            "meta_data": {
                "type": "titan_account",
                "data": ""
            },
            "last_update": "2016-06-28T03:59:40",
            "private_data": null,
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data":
"jjjjjjjjjjjjjjjjjjjjfdsfsadjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj

            "id": "178",
            "is_my_account": true,
            "approved": "0",
            "active_key_history": [
                [
                    "2016-06-28T03:59:40",
                    "G6gUVodY7jnmSWk2Vkp8RKvSrmPgxpnpsXMWpodKxbDBww9Wwzt"
                ]
            ]
        },
        {
            "index": "38",
            "owner_key": "G6uC6kdWehzhJV7e6MkowZvw8V2tjT5GU9CKoVrsb3REp7oT4s8",
            "name": "sy678",
            "block_production_enabled": false,
            "registration_date": "1970-01-01T00:00:00",
            "last_update": "2016-06-28T06:07:18",
            "private_data": null,
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": null,
            "id": "0",
            "is_my_account": true,
            "approved": "0",
            "active_key_history": [
                [
                    "2016-06-28T06:07:18",
                    "G5R7s2UhnG97xrtKEmFpx2iq6VxV3aDpNvJn3DvtYB9tQmFzfxR"
                ]
            ]
        },
        {
                "index": "41",
                "owner_key": "G5YWupJFNhc6SAw33UGvBwK4cKtqC5UbZC18dwXLL1WSRimHY1s",
                "name": "sy6789",
                "block_production_enabled": false,
                "registration_date": "1970-01-01T00:00:00",
                "last_update": "2016-06-28T06:08:32",
                "private_data": null,
                "last_used_gen_sequence": "0",
                "is_favorite": false,
                "public_data": null,
                "id": "0",
                "is_my_account": true,
                "approved": "0",
                "active_key_history": [
                    [
                        "2016-06-28T06:08:32",
                        "G6V7YdCqxvS4cRWzQ2YSzGwN6A15ZaNfHoZHB9AALxZB13riWrw"
                    ]
                ]
            }
        ]
}
```

返回参数:

index: 钱包数据库条目ID

id: 账户ID

name : 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

delegate_info: 代理信息结构体

is_my_account: 是否拥有私钥

approved: 是否支持该账户

is_favorite: 保留字段

block_production_enabled: 是否开启产块功能

last_used_gen_sequence: 保留字段

private_data: 保留字段

查询钱包内所有未注册的账户

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "26",
    "method": "wallet_list_unregistered_accounts"
}
```

请求方法：wallet_list_unregistered_accounts

请求参数：无

Response Result:

```
{
    "id": "26",
    "result": [
        {
            "index": "38",
            "owner_key": "G6uC6kdWehzhJV7e6MkowZvw8V2tjT5GU9CKoVrsb3REp7oT4s8",
            "name": "sy678",
            "block_production_enabled": false,
            "registration_date": "1970-01-01T00:00:00",
            "last_update": "2016-06-28T06:07:18",
            "private_data": null,
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": null,
            "id": "0",
            "is_my_account": true,
            "approved": "0",
            "active_key_history": [
                [
                    "2016-06-28T06:07:18",
                    "G5R7s2UhnG97xrtKEmFpx2iq6VxV3aDpNvJn3DvtYB9tQmFzfxR"
                ]
            ]
        },
        {
            "index": "41",
            "owner_key": "G5YWupJFNhc6SAw33UGvBwK4cKtqC5UbZC18dwXLL1WSRimHY1s",
            "name": "sy6789",
            "block_production_enabled": false,
            "registration_date": "1970-01-01T00:00:00",
            "last_update": "2016-06-28T06:08:32",
            "private_data": null,
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": null,
            "id": "0",
            "is_my_account": true,
            "approved": "0",
            "active_key_history": [
                [
                    "2016-06-28T06:08:32",
                    "G6V7YdCqxvS4cRWzQ2YSzGwN6A15ZaNfHoZHB9AALxZB13riWrw"
                ]
            ]
        }
    ]
}
```

返回参数:

index: 钱包数据库条目ID

id: 账户ID

name : 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

is_my_account: 是否拥有私钥

approved: 是否支持该账户

is_favorite: 保留字段

block_production_enabled: 是否开启产块功能

last_used_gen_sequence: 保留字段

private_data: 保留字段

列出钱包内拥有私钥的所有帐号

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "40",
    "method": "wallet_list_my_accounts"
}
```

请求方法：wallet_list_my_accounts

请求参数：无

Response Result:

```
{
    "id": "17",
    "result": [
        {
            "index": "11",
            "owner_key": "G5VoZWa93vxBF9DudYMhtdDdDjJvcqpk9WJpczFAMLcJ6H7mp8a",
            "name": "gdwhzk1",
            "block_production_enabled": false,
            "registration_date": "1970-01-01T00:00:00",
            "last_update": "2016-06-28T02:20:11",
            "private_data": "null",
            "last_used_gen_sequence": "0",
            "is_favorite": false,
            "public_data": "null",
            "id": "0",
            "is_my_account": true,
            "approved": "0",
            "active_key_history": [
                [
                    "2016-06-28T02:20:11",
                    "G5VoZWa93vxBF9DudYMhtdDdDjJvcqpk9WJpczFAMLcJ6H7mp8a"
                ]
            ]
        }
    ]
}
```

返回参数:

index: 钱包数据库条目ID

id: 账户ID

name : 账户名称

public_data: 注册账户时添加的公开数据

owner_key: 拥有者公钥

active_key_history: 活跃公钥历史

registration_date: 注册时间

last_update: 账户信息最近一次更新时间

delegate_info: 代理信息结构体

is_my_account: 是否拥有私钥

approved: 是否支持该账户

is_favorite: 保留字段

block_production_enabled: 是否开启产块功能

last_used_gen_sequence: 保留字段

private_data: 保留字段

获取本地钱包中或者注册到链上一个账户的地址

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account"
    ],
    "id": "43",
    "method": "wallet_get_account_public_address"
}
```

请求方法：wallet_get_account_public_address

请求参数：账户名

Response Result:

```
{
    "id": "43",
    "result": "GFuj3UjtwsMxZsebUjb6kGFLYWT43GQwky"
}
```

返回参数:

addr: 账户地址

重命名一个未注册的账户

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "currentaccount",
        "newaccount"
    ],
    "id": "45",
    "method": "wallet_account_rename"
}
```

请求方法：wallet_account_rename

请求参数：当前账户名、 新的账户名

Response Result:

```
{
    "id": "29",
    "result": "null"
}
```

返回参数:

无:

获取指定账户余额

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account"
    ],
    "id": "49",
    "method": "wallet_account_balance"
}
```

请求方法：wallet_account_balance

请求参数：账户名

Response Result:

```
{
    "id": "49",
    "result": [
        [
            "gdw0",
            [
                [
                    "0",
                    "999999696000"
                ]
            ]
        ]
    ]
}
```

返回参数:

account: 账户

asset_id: 资产类型

balance: 当前余额

列出指定帐号的所有的余额记录ID

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name"
    ],
    "id": "50",
    "method": "wallet_account_balance_ids"
}
```

请求方法：wallet_account_balance_ids

请求参数：查询账户的用户名

Response Result:

```
{
    "id": "50",
    "result": [
        [
            "gdw0",
            [
                "G6P1pnNuJR4SW7mbiAy1ibfmJwZSZZLZWC",
                "GG7NGwoYicogH7bpZ5MDghBaTAUGzwd53Z"
            ]
        ]
    ]
}
```

返回参数:

不带GDW的字段: 账户名称

账户名称后”[]”内数据: 隶属于该账户的balanceid

受托人领取转账工资接口

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "delegate_name",
        "to_account_name",
        "amount_to_withdraw"
    ],
    "id": "52",
    "method": "wallet_delegate_withdraw_pay"
}
```

请求方法：wallet_delegate_withdraw_pay

请求参数：领取工资的代理名称、 要领取到的账户名称、 领取的工资金额

Response Result:

```
{
    "id": "26",
    "result": {
        "index": "0",
        "is_market": false,
        "fee": {
            "asset_id": "0",
            "amount": "1000"
        },
        "is_confirmed": false,
        "ledger_entries": [
            {
                "from_account": "G5eLFrZG5gvc5g5xqEU6J3GELp5WcQYoqhz4JBPT7BVgTpqqxbN",
                "amount": {
                    "asset_id": "0",
                    "amount": "100000"
                },
                "memo": "aa"
            }
        ],
        "received_time": "2016-06-01T08:32:20",
        "is_virtual": false,
        "block_num": "0",
        "trx": {
            "gdw_inport_asset": {
                "asset_id": "0",
                "amount": "0"
            },
            "operations": [
                {
                    "type": "withdraw_op_type",
                    "data": {
                        "claim_input_data": "",
                        "amount": "101000",
                        "balance_id": "GJEk1wuT9Vix4vQRidmAafz5hCWrS48un2"
                    }
                },
                {
                    "type": "deposit_op_type",
                    "data": {
                        "amount": "100000",
                        "condition": {
                            "asset_id": "0",
                            "slate_id": "0",
                            "type": "withdraw_signature_type",
                            "data": {
                                "owner": "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
                            }
                        }
                    }
                },
                {
                    "type": "imessage_memo_op_type",
                    "data": {
                        "imessage": "aa"
                    }
                }
            ],
            "signatures": [

"203abaffeaa2735b4f9c79bedae77143ac5da1451c92fd7390c1ae502b53578ef47b2751afa034ff7e0232e37f8fb755

            ],
            "expiration": "2016-06-01T09:32:19",
            "gdw_account": ""
        },
        "entry_id": "0ba68a7e9945f9119cb151a2ef5ff110e4d15925",
        "created_time": "2016-06-01T08:32:20",
        "extra_addresses": [
            "G5iSiYMm1fkUtTcW7dQSVNDFZEGXNgj4fy"
        ]
    }
}
```

返回参数:

index: 保留字段

entry_id: 交易单号

block_num: 块号

is_virtual: 保留字段

is_confirmed: 是否确认

is_market: 保留字段

trx: 结构体类型名

受托人工资查询

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name"
    ],
    "id": "53",
    "method": "wallet_delegate_pay_balance_query"
}
```

请求方法：wallet_delegate_pay_balance_query

请求参数：查询代理的用户名

Response Result:

```
{
    "id": "53",
    "result": {
        "pay_balance": "1300000",
        "total_balance": "1300000"
    }
}
```

返回参数:

total_balance: 总工资

pay_balance: 可领取工资

获取钱包内代理的出块状态

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name"
    ],
    "id": "54",
    "method": "wallet_get_delegate_statue"
}
```

请求方法：wallet_get_delegate_statue

请求参数：查询代理的用户名

Response Result:

```
{
    "id": "54",
    "result": false
}
```

返回参数:

result冒号后的字段: 账户当前钱包内的出块状态

删除本地钱包账户

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account_name"
    ],
    "id": "36",
    "method": "wallet_account_delete"
}
```

请求方法：wallet_account_delete

请求参数：要删除的账户名

Response Result:

```
{
    "id": "36",
    "result": true
}
```

返回参数:

result冒号后的字段: 固定返回true

设置本地转账手续费

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "fee"
    ],
    "id": "36",
    "method": "wallet_set_transaction_fee"
}
```

请求方法：wallet_set_transaction_fee

请求参数：要设置的手续费

Response Result:

```
{
    "id": "36",
    "result": {
        "asset_id": "0",
        "amount": "100"
    }
}
```

返回参数:

Asset: 当前手续费

获取当前本地转账手续费

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "symbol"
    ],
    "id": "36",
    "method": "wallet_get_transaction_fee"
}
```

请求方法：wallet_get_transaction_fee

请求参数：要查询的资产标识

Response Result:

```
{
    "id": "36",
    "result": {
        "asset_id": "0",
        "amount": "100"
    }
}
```

返回参数:

Asset: 当前手续费

设置本地钱包是否启用扫描

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "enable"
    ],
    "id": "36",
    "method": "wallet_set_transaction_scanning"
}
```

请求方法：wallet_set_transaction_scanning

请求参数：是否启动扫描

Response Result:

```
{
    "id": "36",
    "result": true
}
```

返回参数:

result冒号后的字段: 返回当前的状态

根据本地拥有的账户的地址或者公钥获取私钥

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "addr"
    ],
    "id": "61",
    "method": "wallet_dump_private_key"
}
```

请求方法：wallet_dump_private_key

请求参数：地址/公钥

Response Result:

```
{
    "id": "61",
    "result": "5JfeSoRUiosgA9kzcPTwmUZs6rEnYPz4iSqGHc9uKcscdjc96g9"
}
```

返回参数:

private_key: 地址

对一个特定的受托人帐号, 启用或禁用区块的生产

调用前提：钱包打开

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "delegate_name",
        "enabled"
    ],
    "id": "67",
    "method": "wallet_delegate_set_block_production"
}
```

请求方法：wallet_delegate_set_block_production

请求参数：受托人账户名称、 是否启用

Response Result:

```
{
    "id": "67",
    "result": "null"
}
```

返回参数:

无:

获取本地钱包指定账户私钥

调用前提：钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "account",
        "key_type"
    ],
    "id": "62",
    "method": "wallet_dump_account_private_key"
}
```

请求方法：wallet_dump_account_private_key

请求参数：账户名、 私钥类型

Response Result:

```
{
    "id": "61",
    "result": "5JfeSoRUiosgA9kzcPTwmUZs6rEnYPz4iSqGHc9uKcscdjc96g9"
}
```

返回参数:

private_key: 地址

借由区块数或ID 来指定区块并获取其区块头 (header)信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "blocknum"
    ],
    "id": "2",
    "method": "blockchain_get_block"
}
```

请求方法：blockchain_get_block

请求参数：块号

Response Result:

```
{
    "id": "2",
    "result": {
        "latency": "0",
        "user_transaction_ids": [],
        "delegate_signature":
"2002db1a3aa58e3768c794fd5295aa4a58691a294b650f6956c4edbea299df72bc282e54d762a6374765287bf7fad734

        "signee_fees_collected": "0",
        "processing_time": "0",
        "timestamp": "2016-03-21T05:16:40",
        "signee_shares_issued": "0",
        "block_num": "1000",
        "previous_secret": "7976c83635b5bc21cac6d4d762b789f97797681d",
        "next_secret_hash": "64159580b455cd4b95a4d22670745454f3693c1d",
        "signee_fees_destroyed": "0",
        "transaction_digest":
"c8cf12fe3180ed901a58a0697a522f1217de72d04529bd255627a4ad6164f0f0",
        "random_seed": "0000000000000000000000000000000000000000",
        "block_size": "166",
        "id": "2df402a360d9956d51749428815ed671cc40ea5c",
        "previous": "5071675dc378fe85a9e4e76cb90651471bb47380"
    }
}
```

返回参数:

previous: 上一块hash

block_num: 块号

timestamp: 块创建时间

transaction_digest: 所有交易签名

next_secret_hash: 所有交易签名

previous_secret: 下一轮出块凭证

delegate_signature: 代理签名

user_transaction_ids: 包含所有的交易id

id: 块id

block_size: 块大小

latency: 产块和本地同步到块时间差

signee_shares_issued: 保留参数

signee_fees_collected: 块中统计的手续费

signee_fees_destroyed: 块中销毁的手续费

random_seed: 随机种子

processing_time: 本地处理时间

获取代理设置参数

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [],
    "id": "36",
    "method": "delegate_get_config"
}
```

请求方法：delegate_get_config

请求参数：无

Response Result:

```
{
    "id": "36",
    "result": {
        "transaction_blacklist": [],
        "transaction_min_fee": "1000",
        "network_min_connection_count": "4",
        "block_max_transaction_count": "4294967295",
        "transaction_max_size": "1024000",
        "operation_blacklist": [],
        "transaction_canonical_signatures_required": false,
        "block_max_size": "1024000",
        "block_max_production_time": "3000000",
        "transaction_imessage_max_soft_length": "40",
        "transaction_imessage_min_fee_coe": "100"
    }
}
```

返回参数:

network_min_connection_count: 允许产块的最小连接数

block_max_transaction_count: 块中最大存放交易的数量

block_max_size: 块最大大小

block_max_production_time: 一个区块最大产块时间

transaction_max_size: 一个交易的最大大小

transaction_canonical_signatures_required: 保留字段

transaction_min_fee: 交易最小手续费

transaction_imessage_max_soft_length: 备注最大长度

transaction_imessage_min_fee_coe: 备注每一字节的花费

transaction_blacklist: 保留字段

operation_blacklist: 保留字段

设置当前代理产块所需的最小连接数

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "count"
    ],
    "id": "36",
    "method": "delegate_set_network_min_connection_count"
}
```

请求方法：delegate_set_network_min_connection_count

请求参数：最小所需连接数

Response Result:

```
{
    "id": "36",
    "result": "null"
}
```

返回参数:

无:

设置当前代理产块最多容纳交易数

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "count"
    ],
    "id": "36",
    "method": "delegate_set_block_max_transaction_count"
}
```

请求方法：delegate_set_block_max_transaction_count

请求参数：本地最大交易数量限制

Response Result:

```
{
    "id": "36",
    "result": "null"
}
```

返回参数:

无:

代理属性设置出的一个块的最大值

调用前提：可以通过73指令验证修改是否成功

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "blocksize"
    ],
    "id": "2",
    "method": "delegate_set_block_max_size"
}
```

请求方法：delegate_set_block_max_size

请求参数：区块的大小

Response Result:

```
{
    "id": "36",
    "result": "null"
}
```

返回参数:

无:

代理属性设置一笔交易的大小的最大值

调用前提：可以通过73指令验证修改是否成功

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "trx_size"
    ],
    "id": "2",
    "method": "delegate_set_transaction_max_size"
}
```

请求方法：delegate_set_transaction_max_size

请求参数：交易的大小

Response Result:

```
{
    "id": "36",
    "result": "null"
}
```

返回参数:

无:

设置代理承认的最小交易费

调用前提：可以通过73指令验证修改是否成功

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
        "transaction"
    ],
    "id": "2",
    "method": "delegate_set_transaction_min_fee"
}
```

请求方法：delegate_set_transaction_min_fee

请求参数：代理认可可以放进自己缓冲区的手续费金额

Response Result:

```
{
    "id": "36",
    "result": "null"
}
```

返回参数:

无:

获取挖矿相关信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
    ],
    "id": "2",
    "method": "get_mining_info"
}
```

请求方法：get_mining_info

请求参数：无

Response Result:

```
    {
"id": "1",
"result": {
    "blocks": 208,
    "currentblocksize": 175,
    "currentblocktx": 0,
    "proof_of_work": "0.00477836420759559",
    "proof_of_stake": "242.18963623046875000",
    "errors": "",
    "genproclimit": 0,
    "networkhashps": "0.00000000000000000",
    "pooledtx": 0,
    "chain": "main",
    "generate": false,
    "hashespersec": "0.00000000000000000"
            }
    }
```

返回参数: blocks: 当前块高度

currentblocksize: 当前块大小

currentblocktx: 当前块中的交易

proof_of_work: pow的全网难度

proof_of_stake: pos的全网难度

errors: 保留字段

genproclimit: 当前产块线程数

networkhashps: 当前节点挖矿速度

pooledtx: 缓存中交易数量

chain: 链种类

generate: 是否挖矿

hashespersec: 本程序挖矿速率

获取当前全网难度

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
    ],
    "id": "2",
    "method": "get_difficulty"
}
```

请求方法：get_difficulty

请求参数：无

Response Result:

```
    {
"id": "1",
"result": {
    "proof_of_work": "0.00342694343999028",
    "proof_of_stake": "284.94445800781250000"
}
```

}

返回参数: proof_of_work: pow的全网难度

proof_of_stake: pos的全网难度

设置是否挖矿

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [true,1
    ],
    "id": "2",
    "method": "set_generate"
}
```

请求方法：set_generate

请求参数： is_generate: 是否进行挖矿

procNum: 挖矿线程数（默认-1 所有线程数）

Response Result:

```
    {
"id": "1",
"result": true
    }
```

返回参数:

无:

获取POW工作包

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
    ],
    "id": "2",
    "method": "get_work"
}
```

请求方法：get_work

请求参数：无

Response Result:

```
    {
"id": "1",
"result": {
    "hashnoNonce": "141a0988a31d29adc5759b8468e6455eefc050ff892942a21196c56194f1be6e",
    "data":
"dd6127f3a54c1e82d9df88653c00ad4ccac9cc8ee4000000a180025ac8cf12fe3180ed901a58a0697a522f1217de72d0

    "target": "0000011218000000000000000000000000000000000000000000000000000000"
            }
    }
```

返回参数: hashnoNonce: 不包含nonce计算出的hash值

data: 块的hash数据，最后16位是nonce的16进制数据

target: 块的目标hash值

提交pow工作结果

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": ["hashnoNonce",25655
    ],
    "id": "2",
    "method": "sgdwmit_block"
}
```

请求方法：sgdwmit_block

请求参数： hashnoNonce: 需要提交POW工作包的hashNoNonce

Nonce: 满足结果的nonce

Response Result:

```
    {
"id": "1",
"result": true
    }
```

返回参数: is_success: 是否成功提交

设置挖矿收款人

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": ["test"
    ],
    "id": "2",
    "method": "set_coinbase"
}
```

请求方法：set_coinbase

请求参数： account_name: 账户名或者地址

Response Result:

```
    {
"id": "1",
"result": true
    }
```

返回参数: is_success: 是否设置成功

获取pos相关信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "params": [
    ],
    "id": "2",
    "method": "get_staking_info"
}
```

请求方法：get_staking_info

请求参数：无

Response Result:

```
    {
"id": "1",
"result": {
    "enabled": true,
    "currentblocksize": 175,
    "currentblocktx": 0,
    "difficulty": "541.02917480468750000",
    "weight": "34560000000",
    "netstakeweight": 0,
    "nExpectedTime": 0
}
```

返回参数: enabled: 是否开启pos

currentblocksize: 当前块大小

currentblocktx: 当前块交易数量

difficulty: 当前pos难度

weight: 当前账户权重

netstakeweight: 当前网络权重

nExpectedTime: 挖到块预计需要时间

合约相关

注册合约（链上）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_register",
    "params": [
        "test01",
        "C:/Users/hasee/Desktop/test1.gpc",
        "GDW",
        "0.1"
    ]
}
```

请求方法：contract_register

请求参数：注册人，字节码文件路径，资产标识，执行花费上限

Response Result:

```
{
    "id": "1",
    "result": "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
}
```

返回参数：合约id

注册合约（testing）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_register_testing",
    "params": [
        "test01",
        "C:/Users/hasee/Desktop/test1.gpc"
    ]
}
```

请求方法：contract_register_testing

请求参数：注册人，合约字节码文件

Response Result:

```
{
    "id": "1",
    "result": [
        {
            "amount": 1000,
            "asset_id": 0
        },
        {
            "amount": 10000000,
            "asset_id": 0
        },
        {
            "amount": 473,
            "asset_id": 0
        },
        {
            "amount": 100,
            "asset_id": 0
        }
    ]
}
```

返回参数： 交易手续费，合约保证金，执行花费，字节码花费，注册执行花费

升级合约（链上）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_upgrade",
    "params": [
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "test01",
        "my_contract_1",
        "this is my first contract"
        "GDW"
        "10"
    ]
}
```

请求方法：contract_upgrade

请求参数：合约id，升级发起人，合约新名称，合约描述，资产符号，执行花费上限

Response Result:

```
{
    "id": "1",
    "result": {
        "index": 0,
        "entry_id": "1a6f38c365c7b14d4d24a263c74c9d80f0296394",
        "block_num": 0,
        "is_virtual": false,
        "is_confirmed": false,
        "is_market": false,
        "trx": {
            "expiration": "2016-09-28T06:44:46",
            "gdw_account": "",
            "gdw_inport_asset": {
                "amount": 0,
                "asset_id": 0
            },
            "operations": [
                {
                    "type": "contract_upgrade_op_type",
                    "data": {
                        "id": "GJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
                        "name": "my_contract_1",
                        "desc": "this is my first contract",
                        "balances": [
                            [
                                "GPe9u4ZmBj38D5GzPPSz9LLe8J3LpM89o8",
                                1000
                            ]
                        ]
                    }
                }
            ],
            "signatures": [

"2027fbcf936eeb898593fade96c26bef189462d60b1e60a8fb95f73cbfd0e6632423346678f5691a9f7f5e20681656b7

            ]
        },
        "ledger_entries": [
            {
                "amount": {
                    "amount": 0,
                    "asset_id": 0
                },
                "memo": ""
            }
        ],
        "fee": {
            "amount": 10001000,
            "asset_id": 0
        },
        "created_time": "2016-09-28T05:44:46",
        "received_time": "2016-09-28T05:44:46",
        "extra_addresses": [
            "GJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
        ]
    }
}
```

返回参数：交易记录

升级合约（testing）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_upgrade_testing",
    "params": [
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "test01",
        "my_contract_1",
        "this is my first contract"
    ]
}
```

请求方法：contract_upgrade_testing

请求参数：合约id，升级发起人，合约新名称，合约描述

Response Result:

```
{
    "id": "1",
    "result": [
        {
            "amount": 1000,
            "asset_id": 0
        },
        {
            "amount": 16,
            "asset_id": 0
        },
        {
            "amount": 1000000,
            "asset_id": 0
        }
    ]
}
```

返回参数：交易手续费，执行花费，保证金

删除合约（链上）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_destroy",
    "params": [
        "cPZdkAuyDthiftdDzp8Hg7PqyK9MiTnL2q",
        "test01"
        "GDW"
        "10"
    ]
}
```

请求方法：contract_destroy

请求参数: 合约id，删除发起人，资产id,执行费用上限

Response Result:

```
{
    "id": "1",
    "result": {
        "index": 0,
        "entry_id": "111eae250a783bba4d9f5e8bee56af762b447dfe",
        "block_num": 0,
        "is_virtual": false,
        "is_confirmed": false,
        "is_market": false,
        "trx": {
            "expiration": "2016-09-28T07:02:06",
            "gdw_account": "",
            "gdw_inport_asset": {
                "amount": 0,
                "asset_id": 0
            },
            "operations": [
                {
                    "type": "contract_destroy_op_type",
                    "data": {
                        "id": "GPZdkAuyDthiftdDzp8Hg7PqyK9MiTnL2q",
                        "balances": [
                            [
                                "GPe9u4ZmBj38D5GzPPSz9LLe8J3LpM89o8",
                                1000
                            ]
                        ]
                    }
                }
            ],
            "signatures": [

"1f4d9614000a6000940b98186d28f1e28faa0326f912b234ed31822c10140889b72183551449b0b18da4c9f4c355b345

            ]
        },
        "ledger_entries": [
            {
                "amount": {
                    "amount": 0,
                    "asset_id": 0
                },
                "memo": ""
            }
        ],
        "fee": {
            "amount": 1000,
            "asset_id": 0
        },
        "created_time": "2016-09-28T06:02:06",
        "received_time": "2016-09-28T06:02:06",
        "extra_addresses": [
            "GPZdkAuyDthiftdDzp8Hg7PqyK9MiTnL2q"
        ]
    }
}
```

返回参数：交易记录

删除合约（testing）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_destroy_testing",
    "params": [
        "cPZdkAuyDthiftdDzp8Hg7PqyK9MiTnL2q",
        "test01"
    ]
}
```

请求方法：contract_destroy_testing

请求参数：合约id，删除发起人

Response Result:

```
{
    "id": "1",
    "result": [
        {
            "amount": 1000,
            "asset_id": 0
        },
        {
            "amount": 16,
            "asset_id": 0
        }
    ]
}
```

返回参数：交易手续费，执行花费

调用合约（链上）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "0",
    "method": "contract_call",
    "params": [
        "cMR24hkhmuJ8X6eiEhdT2vqV1FvCcS5vDf",
        "gdw0",
        "print",
        "",
        "GDW",
        "1"
    ]
}
```

请求方法：contract_call

请求参数：合约id，调用发起人，调用方法，调用参数，资产符号，执行花费上限

Response Result:

```
{
    "id": "0",
    "result": {
        "index": 0,
        "entry_id": "d57f3b339a40eb1d9a72359d1a3c9d9e67434673",
        "block_num": 0,
        "is_virtual": false,
        "is_confirmed": false,
        "is_market": false,
        "trx": {
            "expiration": "2016-11-21T06:53:19",
            "gdw_account": "",
            "gdw_inport_asset": {
                "amount": 0,
                "asset_id": 0
            },
            "operations": [
                {
                    "type": "contract_call_op_type",
                    "data": {
                        "caller": "G8CqTJMSBzgqFxudU5TWzQyVDgmqrqax3ApNDGDwEZV7uXKTND2",
                        "balances": [
                            [
                                "GJqWqjsVdAXCK99knVmeCE7EpAuFdTmFux",
                                101000
                            ]
                        ],
                        "contract": "cMR24hkhmuJ8X6eiEhdT2vqV1FvCcS5vDf",
                        "costlimit": {
                            "amount": 100000,
                            "asset_id": 0
                        },
                        "method": "print",
                        "args": ""
                    }
                }
            ],
            "signatures": [
 "2030cd70eadf10532a4b27613252c3c2c5bd61ccd5f22911c18727676625cae9411ffed0f6383efdd48568ed9559d981

            ]
        },
        "ledger_entries": [],
        "fee": {
            "amount": 1000,
            "asset_id": 0
        },
        "created_time": "2016-11-21T05:53:19",
        "received_time": "2016-11-21T05:53:19",
        "extra_addresses": []
    }
}
```

返回参数：交易记录

调用合约（testing）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_call_testing",
    "params": [
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "test01",
        "function_not_exist",
        ""
    ]
}
```

请求方法：contract_call_testing

请求参数: 合约id，调用发起人，调用方法，调用参数

Response Result:

```
{
    "id": "1",
    "result": [
        {
            "amount": 1000,
            "asset_id": 0
        },
        {
            "amount": 103,
            "asset_id": 0
        }
    ]
}
```

返回参数：交易手续费，执行花费

向合约转账（链上）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "wallet_transfer_to_contract",
    "params": [
        "10",
        "GDW",
        "test01",
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "1"
    ]
}
```

请求方法：wallet_transfer_to_contract

请求参数：转账金额，资产标识，转出账户，合约id，执行花费上限

Response Result:

```
{
    "id": "1",
    "result": {
        "index": 0,
        "entry_id": "1003bfb6f006509eadaa0567152b995aca63db1b",
        "block_num": 0,
        "is_virtual": false,
        "is_confirmed": false,
        "is_market": false,
        "trx": {
            "expiration": "2016-09-28T07:34:09",
            "gdw_account": "",
            "gdw_inport_asset": {
                "amount": 0,
                "asset_id": 0
            },
            "operations": [
                {
                    "type": "transfer_contract_op_type",
                    "data": {
                        "from": "G785tCDTiTNu2FxDkzKazkTvFbVeJ4bAo9A1VuJV7yBdB7J7UbB",
                        "costlimit": {
                            "amount": 100000,
                            "asset_id": 0
                        },
                        "transfer_amount": {
                            "amount": 1000000,
                            "asset_id": 0
                        },
                        "balances": [
                            [
                                "GPe9u4ZmBj38D5GzPPSz9LLe8J3LpM89o8",
                                1101000
                            ]
                        ],
                        "contract_id": "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
                    }
                }
            ],
            "signatures": [

"1f0acbf0ab2c3338e478157e02bcd770e22a832b85ace93bb58cc7e9c3f241b59e7029f1c7a029417d3753dc508712e9

            ]
        },
        "ledger_entries": [
            {
                "from_account": "G785tCDTiTNu2FxDkzKazkTvFbVeJ4bAo9A1VuJV7yBdB7J7UbB",
                "amount": {
                    "amount": 1000000,
                    "asset_id": 0
                },
                "memo": ""
            }
        ],
        "fee": {
            "amount": 1000,
            "asset_id": 0
        },
        "created_time": "2016-09-28T06:34:09",
        "received_time": "2016-09-28T06:34:09",
        "extra_addresses": [
            "GJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
        ]
    }
}
```

返回参数：交易记录

向合约转账（testing）

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "wallet_transfer_to_contract_testing",
    "params": [
        "10000",
        "GDW",
        "test01",
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
    ]
}
```

请求方法：wallet_transfer_to_contract_testing

请求参数：转账金额，资产标识，转出账户，合约id

Response Result:

```
{
    "id": "1",
    "result": [
        {
            "amount": 1000,
            "asset_id": 0
        },
        {
            "amount": 1000000000,
            "asset_id": 0
        },
        {
            "amount": 0,
            "asset_id": 0
        }
    ]
}
```

返回参数：交易手续费，转账金额，执行花费

查询合约信息

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_get_info",
    "params": [
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
    ]
}
```

请求方法：contract_get_info

请求参数：合约id

Response Result:

```
{
    "id": "1",
    "result": {
        "contract_name": "my_contract_1",
        "id": "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "level": "forever",
        "owner": "G785tCDTiTNu2FxDkzKazkTvFbVeJ4bAo9A1VuJV7yBdB7J7UbB",
        "owner_address": "GLMSC5rhePQeVgLQrjqVMdC9pmB97451CG",
        "owner_name": ""
        "state": "valid",
        "description": "this is my first contract",
        "code_printable": {
            "abi": [
                "init",
                "start"
            ],
            "offline_abi": [],
            "events": [],
            "printable_storage_properties": [],
            "printable_code":
"1b4c7561530019930d0a1a0a0408040808785600000000000000000000002877400121 40453a5c476f6f70616c332e30

            "code_hash": "df6d3e2343718599ac6343ba2bd3d4e7cc7e7347"
        },
        "trx_id": "ccb1c3cdadfb6a0883863d5b2c9605cd3ba9ab99"
    }
}
```

返回参数：

contract_name：合约名

id：合约Id

level：临时合约还是永久合约

owner：合约注册人

state：启用状态

description:合约描述

code_printable：字节码信息

trx_id:合约被注册的交易id

查询合约余额

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_get_balance",
    "params": [
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
    ]
}
```

请求方法：contract_get_balance

请求参数：合约id

Response Result:

```
{
    "id": "1",
    "result": [
        {
            "condition": {
                "asset_id": 0,
                "slate_id": 0,
                "type": "withdraw_signature_type",
                "data": {
                    "owner": "GJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
                }
            },
            "balance": 1000000,
            "deposit_date": "2016-09-28T06:34:00",
            "last_update": "2016-09-28T06:34:00",
            "meta_data": null
        },
        {
            "condition": {
                "asset_id": 0,
                "slate_id": 0,
                "type": "withdraw_signature_type",
                "data": {
                    "owner": "GJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE"
                }
            },
            "balance": 0,
            "deposit_date": "2016-09-26T09:08:40",
            "last_update": "2016-09-28T05:44:40",
            "meta_data": null
        }
    ]
}
```

返回参数：合约保证金balance信息，合约余额balance信息

打开/关闭本地解释器验证开关

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

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

请求方法：blockchain_set_node_vm_enabled

请求参数：true/false,是否启动vm

Response Result:

```
{
    "id": "1",
    "result": null
}
```

返回参数:无

获取是否打开本地解释器验证开关

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "blockchain_get_node_vm_enabled",
    "params": [

    ]
}
```

请求方法：blockchain_get_node_vm_enabled

请求参数：无

Response Result:

```
{
    "id": "1",
    "result": true
}
```

返回参数：vm启用状态

导出合约到文件

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "contract_load_to_file",
    "params": [
        "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "D:/contract_to_export.gpc"
    ]
}
```

请求方法：contract_load_to_file

请求参数: 合约id，导出的文件的路径

Response Result:

```
{
    "id": "1",
    "result": {
        "contract_name": "my_contract_1",
        "id": "cJuXHfWSGo51cGbD3dXVpQs8B5v7MrFZcE",
        "level": "forever",
        "owner": "G785tCDTiTNu2FxDkzKazkTvFbVeJ4bAo9A1VuJV7yBdB7J7UbB",
        "state": "valid",
        "description": "this is my first contract",
        "code_printable": {
            "abi": [
                "init",
                "start"
            ],
            "offline_abi": [

            ],
            "events": [

            ],
            "printable_code":
"1b4c7561530019930d0a1a0a04080408087856000000000000000000000000002877400121404553a5c476f6f70616c332e30

            "code_hash": "df6d3e2343718599ac6343ba2bd3d4e7cc7e7347"
        },
        "trx_id": "ccb1c3cdadfb6a0883863d5b2c9605cd3ba9ab99"
    }
}
```

返回参数：

contract_name：合约名

id：合约Id

level：临时合约还是永久合约

owner：合约注册人

state：启用状态

description:合约描述

code_printable：字节码信息

trx_id:合约被注册的交易id

获取合约结果交易hash

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "id": "0",
    "method": "get_result_trx_id",
    "params": [
        "84f277e2097b9039b75bb66b85d09fa2e97381f6"
    ]
}
```

请求方法：get_result_trx_id

请求参数：原始请求交易id

Response Result:

```
{
    "id": "0",
    "result": "94201d7b852e887b165c93458fe13ba48bd6f57e"
}
```

返回参数：结果交易id

根据原始交易hash(创建合约)获取合约地址

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "id": "0",
    "method": "get_contract_registered_in_transaction",
    "params": [
        "8bfce6f1d70f6b61a98d62910c18c25b2b8a8154"
    ]
}
```

请求方法：get_contract_registered_in_transaction

请求参数：合约id

Response Result:

```
{
    "id": "0",
    "result": "c8wiH4jjovUC9c6FG7B15kwD6eqaVLVBJw"
}
```

返回参数：合约id

根据合约地址获取创建此合约的原始交易hash

调用前提：无

Request:

```
{
    "jsonrpc": "2.0",
    "id": "0",
    "method": "get_transaction_id_contract_registered",
    "params": [
        "cWyvc7YvfGxs74aMuGY4cb4bX9RyxTJWw"
    ]
}
```

请求方法：get_transaction_id_contract_registered

请求参数：合约id

Response Result:

```
{
    "id": "0",
    "result": "10c14b782d5574b4d0e4a70eeff788eb07dd7a3f"
}
```

返回参数：交易id

获取链上的所有合约信息

调用前提：无

Request:

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

请求方法：blockchain_get_all_contracts

请求参数：无

Response Result:

```
{
    "id":1,
    "result":
    ["c9dAKb5nDttSvH3T9oEuTdu3WeScKt8vcv","cCs5JotmSyYK696nafFcYAX6iPZh7Xbh2u"]
}
```

返回参数：合约地址数组

获取链上的所有永久合约信息

调用前提：无

Request:

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

请求方法：blockchain_get_forever_contracts

请求参数：无 Response Result:

```
{
    "id":1,"
    "result":[["G9dAKb5nDttSvH3T9oEuTdu3WeScKt8vcv","wens"]]
}
```

返回参数：合约名数组

获取钱包下账户的合约信息

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

```
{
    "jsonrpc": "2.0",
    "id": "1",
    "method": "wallet_get_contracts",
    "params": ["account_name"] (如果不填参数默认是获取钱包下所有账户的合约信息)
}
```

请求方法：wallet_get_contracts

请求参数：账户名

Response Result:

```
{
    "id":1,
    "result":["c9dAKb5nDttSvH3T9oEuTdu3WeScKt8vcv","cCs5JotmSyYK696nafFcYAX6iPZh7Xbh2u"]
}
```

请求参数:合约id数组

扫描更新钱包下所有账户的合约信息

调用前提：RPC登录认证成功，钱包打开，钱包解锁

Request:

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

请求方法：wallet_scan_contracts

请求参数：无

Response Result:

```
{
    "id":1,
    "result":null
}
```

返回参数：无

RPC Tutorial

JsonRpc的支持

支持两种JsonRpc模式

- Tcp-JsonRpc模式
- Http-JsonRpc模式

如何启动RPC服务端监听

- 启动Tcp-JsonRpc模式

```
gdw --rpcuser RPC用户名 --rpcpassword RPC密码 --rpcport 监听端口 --server --data-dir 数据目录
```

示例：

```
gdw --rpcuser admin --rpcpassword 123456 --rpcport 10086 --server --data-dir chain_data
```

- 启动Http-JsonRpc模式

```
gdw --rpcuser RPC用户名 --rpcpassword RPC密码 --httpdendpoint 监听Endpoint --server --data-dir
数据目录
```

示例：

```
gdw --rpcuser admin --rpcpassword 123456 --httpdendpoint 127.0.0.1:8080 --server --data-dir
chain_data
```

如何进行一次有效的RPC交互

- 在Tcp-JsonRpc模式下

1. 发起一个TCP连接
2. 将RPC请求做为数据流，发送登录(login)的RPC请求，并接收到正确的RPC返回
3. 进行RPC交互
4. 连接断开

- 在Http-JsonRpc模式下

1. 新增一个http header，key为Authorization，value为字符串”Basic ” + base64(rpcuser:rpcpassword) 或者使用Basic Authorization格式的url, 比如 http://rpcuser:rpcpassword@127.0.0.1:8080/rpc
2. 将RPC请求做为HTTP包的包体，发起一个HTTP POST请求
3. 获取HTTP返回，包体内容就是RPC的返回

简单DEMO示例

Python 2.7简单示例

- 在Tcp-JsonRpc模式下

SimpleTcpJsonRpc.py

```python
#!/usr/bin/env python
# encoding: utf-8


import socket

def is_receive_complete(data):
    if data is None or data == '':
        return False
    json_start = False
    json_tag_count = 0
    for c in data:
        if c == '{':
            if not json_start:
                json_start = True
            json_tag_count += 1
        elif c == '}':
            json_tag_count -= 1
        if json_start and json_tag_count == 0:
            return True
    return False

def recv_until_json_complete(sd):
    left_data = ""
    while not is_receive_complete(left_data):
        data = sd.recv(4096)
        left_data += data
    return left_data


login_payload = ''' { "jsonrpc": "2.0", "params": [ "admin", "123456" ], "id": "1", "method": "login" } '''
get_info_payload = ''' { "jsonrpc": "2.0", "params": [], "id": "2", "method": "get_info" } '''

RpcServerEndpoint = "127.0.0.1:10086"
conn_tuple = RpcServerEndpoint.split(":")
endpoint_tuple = (conn_tuple[0], int(conn_tuple[1]))

sd = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sd.connect(endpoint_tuple)

sd.sendall(login_payload)
recv_data = recv_until_json_complete(sd)
print recv_data

sd.sendall(get_info_payload)
recv_data = recv_until_json_complete(sd)
print recv_data

sd.close()
```

- 在Http-JsonRpc模式下

SimpleHttpJsonRpc.py

```
#!/usr/bin/env python
# encoding: utf-8


import base64
from requests import Request, Session

get_info_payload = {"jsonrpc": "2.0", "params": [], "id": "2", "method": "info"}

rpc_auth = "Basic " + base64.b64encode("%s:%s" % ("admin", "123456"))
auth_headers = {'Content-Type': 'application/json', 'Authorization': rpc_auth}
post_url = "http://127.0.0.1:8080/rpc"

s = Session()
req = Request('POST', post_url, json=get_info_payload, headers=auth_headers)
prepped = req.prepare()
resp = s.send(prepped)

print resp.text
```

Indices and tables

- Index
- Module Index