

XDAG RPC Documentation

I . Data Structure

1. v0.8.0 Update
2. BlockResponse
3. TxLink
4. Link
5. XdagStatusResponse
6. SyncingResult
7. NetConnResponse

II. RPC Function

1. xdag_getBlockByHash
 - 1.1 xdag_getBlockByHash (String hash, int page)
PS: Show request without respond.
 - 1.2 xdag_getBlockByHash (String BlockHash, int Page, int PageSize)
 - 1.3 xdag_getBlockByHash (String hash, int page, String startTime, String endTime)
 - 1.4 xdag_getBlockByHash (String hash, int page, String startTime, String endTime, int PageSize)
2. xdag_getBlockByNumber
 - 2.1 xdag_getBlockByNumber(String BlockHeight, int Page)
 - 2.2 xdag_getBlockByNumber(String BlockHeight, int Page, int PageSize)
3. xdag_blockNumber
4. xdag_coinbase
5. xdag_getBalance
6. xdag_getTransactionNonce
7. xdag_syncing
8. xdag_getTotalBalance
9. xdag_getStatus
10. xdag_personal_sendTransaction
11. xdag_personal_sendSafeTransaction
12. xdag_getRewardByNumber
13. xdag_sendRawTransaction
14. xdag_netType
15. xdag_netConnectionList
- ~~16. xdag_poolConfig~~
- ~~17. xdag_updatePoolConfig~~

XDAG RPC Documentation

I . Data Structure

1. v0.8.0 Update
2. BlockResponse
3. TxLink
4. Link
5. XdagStatusResponse
6. SyncingResult
7. NetConnResponse

1. v0.8.0 Update

Added two RPC functions:

1. xdag_getTransactionNonce
2. xdag_personal_sendSafeTransaction

2. BlockResponse

Used to describe a block structure

```
long height; // block height, it will be zero if block is not a main block
String balance; // block balance
long blockTime; // block generate time
long timeStamp; // block generate time(xdag_time)
String state; // block state, "Main" "Rejected" "Accepted" "Pending"
String hash; // block hash
String address; // block address
String remark; // block remark
String diff; // block difficulty
String type; // block type, "Main" "wallet" "Transaction"
String flags; // block flags
List<Link> refs; // reference
List<TxLink> transactions; // transaction history when block as a address
```

3. TxLink

Used to describe transaction history

```
int direction; // 0 input 1 output 2 earning
String hashlow; // transaction hashlow (28bytes)
String address; // address
String amount; // xfer amount
long time; // transaction create time
String remark; // transaction remark
```

4. Link

Used to describe the reference

```
int direction; // 0 input 1 output 2 fee
String address; // block address
String hashlow; // block hashlow
String amount; // amount
```

5. XdagStatusResponse

Used to describe the status of the XDAG network

```
String nblock; // the number of block in this pool
String totalblocks; // the number of block in the mainnet
String nmain; // the number of mainblock in this pool
String totalmain; // the number of mainblock in the mainnet
String curDiff; // the max difficulty for this pool
String netDiff; // the max difficulty for the mainnet
String hashRateOurs; // 4hours hashrate for this pool
String hashRateTotal; // 4hours hashrate for the mainnet
String ourSupply; // pool supply
String netSupply; // current supply for xdag
```

6. SyncingResult

Used to describe the synchronization of the pool (node)

```
String currentBlock; // current height in this pool
String highestBlock; // current height for the mainnet
boolean isSyncDone; // "true" if synchronization is complete,"false" otherwise
```

7. NetConnResponse

Used to describe the net connection of the pool

```
InetSocketAddress nodeAddress;
long connectTime;
long inBound;
long outBound;
```

II. RPC Function

1. xdag_getBlockByHash
2. xdag_getBlockByNumber
3. xdag_blockNumber
4. xdag_coinbase
5. xdag_getBalance
6. xdag_getTransactionNonce
7. xdag_syncing
8. xdag_getTotalBalance
9. xdag_getStatus
10. xdag_personal_sendTransaction
11. xdag_personal_sendSafeTransaction
12. xdag_getRewardByNumber
13. xdag_sendRawTransaction
14. xdag_netType

- 15. ~~xdag_netConnectionList~~
- 16. ~~xdag_poolConfig~~
- 17. ~~xdag_updatePoolConfig~~

1. xdag_getBlockByHash

1.1 xdag_getBlockByHash (String hash, int page)

Used to return a block based on hash & page.

Req

```
# get by accountAddress
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":
[\"AccountAddress\",\"Page\"],\"id\":1}"

## example 1:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":
[\"8PKBGqTAAnzSG8qbZGzDhouihgC3cyvCH\",\"1\"],\"id\":1}"

# get block by blockAddress
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":
[\"BlockAddress\",\"Page\"],\"id\":1}"

## example 2:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":
[\"UDbeuykAXIKWNXuVEzpj5uN4imEenqPK\",\"1\"],\"id\":1}"
```

Resp

```
# Response of example 1:
{
  "jsonrpc": "2.0",
  "result": {
    "height": 0,
    "balance": "10.000000000",
    "blockTime": 1720015040000,
    "timeStamp": 1761295400960,
    "state": "Accepted",
    "hash": null,
    "address": "8PKBGqTAAnzSG8qbZGzDhouihgC3cyvCH",
    "remark": null,
    "diff": null,
    "type": "wallet",
    "flags": null,
    "totalPage": 1,
    "refs": null,
    "transactions": [
      {
        "direction": 0,
```

```

        "hashlow":
"0000000000000000481baa9925612a67c1da7111a20a10befc477bbd9a5f8c72",
        "address": "coxfmr17R/y+EAqiEXHawwcqYSWZqhtI",
        "amount": "10.000000000",
        "time": 1739414189746,
        "remark": ""
    }
}
],
},
    "id": "1"
}

```

Response of example 2:

```

{
    "jsonrpc": "2.0",
    "result": {
        "height": 3145037,
        "balance": "0.000000000",
        "blockTime": 1739376191999,
        "timeStamp": 1781121220607,
        "state": "Main",
        "hash":
"e22eb2ef6ebce7b2caa39e1e618a78e3e6633a13957b35b0825c0029bbde3650",
        "address": "UDbeuykAXIKwNXuVEzpj5uN4imEenqPK",
        "remark": "XdagJ_test02_179",
        "diff": "0xcd6e16f95eec0a915eb634d5ee",
        "type": "Main",
        "flags": "3f",
        "totalPage": 1,
        "refs": [
            {
                "direction": 2,
                "address": "UDbeuykAXIKwNXuVEzpj5uN4imEenqPK",
                "hashlow":
"0000000000000000caa39e1e618a78e3e6633a13957b35b0825c0029bbde3650",
                "amount": "0.200000000"
            },
            {
                "direction": 1,
                "address": "gOycCH9eJhVqn7TfOAJADuuEd13IL6j5",
                "hashlow":
"0000000000000000f9a82fc85d7784eb0e400238dfb49f6a15265e7f709cec80",
                "amount": "0.000000000"
            },
            {
                "direction": 1,
                "address": "9pMKSvR5mBnzdyVawcg1rk1RWS42YZRh",
                "hashlow":
"0000000000000000619461362e595149ae35c8c15a8575f3199879f44a0a93f6",
                "amount": "0.000000000"
            },
            {
                "direction": 1,
                "address": "YMsXKKZ+fENDAq57Wyd4M3QehC85BT5j",
                "hashlow":
"0000000000000000633e05392f841e743378275b7bae0243437c7ea62817cb60",
                "amount": "0.000000000"
            }
        ]
    }
}

```

```

    },
    {
        "direction": 1,
        "address": "yCLrsDkihpCh3fPtbx7By801w2Xyz4h7",
        "hashlow":
"00000000000000007b8867f265c335cdcbc11e6fedf3dda190862239b0eb22c8",
        "amount": "0.00000000"
    }
],
"transactions": [
    {
        "direction": 2,
        "hashlow":
"0000000000000000caa39e1e618a78e3e6633a13957b35b0825c0029bbde3650",
        "address": "UDbeuykAXIKwNXuVEzpj5uN4imEenqPK",
        "amount": "64.200000000",
        "time": 1739376191999,
        "remark": "xdagJ_test02_179"
    }
]
},
"id": "1"
}

```

PS: Show request without respond.

1.2 xdag_getBlockByHash (String BlockHash, int Page, int PageSize)

Used to return a block based on hash & page size.

Req

```

curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBlockByHash","params":
[\'BlockHash\',\'Page\',\'PageSize\'],\'id\':1}'

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBlockByHash","params":
[\'4mvr3DNkpwY9ikpgY4maamsQqUmXjR2hp\',\'1\',\'3\'],\'id\':1}'

```

1.3 xdag_getBlockByHash (String hash, int page, String startTime, String endTime)

Used to return a block based on hash & page & time range.

Req

```
# Time range -- Date:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":[\"BlockHash\",\"Page\",\"StartTime\",\"EndTime\"],\"id\":1}"

## example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":[\"Y988dNXpwuWn13oeL1e3dEJ/d9ths8ho\",\"1\",\"2023-7-27 12:30:10\",\"2023-7-27 13:05:20\"],\"id\":1}"

# Time range -- Timestamp:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":[\"BlockHash\",\"Page\",\"StartTimestamp\",\"EndTimestamp\"],\"id\":1}"

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":[\"55Tffne2cwGSDRJU3kouvZfRNjk19ZaE7\",\"1\",\"1690418353515\",\"1690433215999\"],\"id\":1}"
```

1.4 xdag_getBlockByHash (String hash, int page, String startTime, String endTime, int PageSize)

Used to return a block based on hash & time range & page size.

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":[\"BlockHash\",\"Page\",\"StartTimestamp\",\"EndTimestamp\",\"PageSize\"],\"id\":1}"

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByHash\",\"params\":[\"4mvr3DNkpWY9ikpGy4maMSQqUmXjR2hp\",\"1\",\"1691675158000\",\"1691675168999\", \"3\"],\"id\":1}"
```

2. xdag_getBlockByNumber

2.1 xdag_getBlockByNumber(String BlockHeight, int Page)

Used to return a block based on block number & page

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByNumber\",\"params\":[\"BlockHeight\",\"Page\"],\"id\":1}"

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_getBlockByNumber\",\"params\":[\"3145037\",\"1\"],\"id\":1}"
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": {
    "height": 3145037,
    "balance": "0.000000000",
    "blockTime": 1739376191999,
    "timeStamp": 1781121220607,
    "state": "Main",
    "hash":
    "e22eb2ef6ebce7b2caa39e1e618a78e3e6633a13957b35b0825c0029bbde3650",
    "address": "UDbeuykAXIKWNXuVEzpj5uN4imEenqPK",
    "remark": "xdagJ_test02_179",
    "diff": "0xcd6e16f95eec0a915eb634d5ee",
    "type": "Main",
    "flags": "3f",
    "totalPage": 1,
    "refs": [
      {
        "direction": 2,
        "address": "UDbeuykAXIKWNXuVEzpj5uN4imEenqPK",
        "hashlow":
        "0000000000000000caa39e1e618a78e3e6633a13957b35b0825c0029bbde3650",
        "amount": "0.200000000"
      },
      {
        "direction": 1,
        "address": "gOycCH9eJhVqn7TfOAJADuuEd13IL6j5",
        "hashlow":
        "0000000000000000f9a82fc85d7784eb0e400238dfb49f6a15265e7f709cec80",
        "amount": "0.000000000"
      },
      {
        "direction": 1,
        "address": "9pMKsvR5mBnzdyVawcg1rk1rWS42YZRh",
        "hashlow":
        "00000000000000000619461362e595149ae35c8c15a8575f3199879f44a0a93f6",
        "amount": "0.000000000"
      },
      {
        "direction": 1,
        "address": "YMsXKKZ+fENDAq57Wyd4M3QehC85BT5j",
        "hashlow":
        "00000000000000000633e05392f841e743378275b7bae0243437c7ea62817cb60",
        "amount": "0.000000000"
      }
    ]
  }
}
```



```

    },
    {
        "direction": 1,
        "address": "yCLrsDkihpCh3fPtbx7By801w2Xyz4h7",
        "hashlow":
"000000000000000007b8867f265c335cdcbc11e6fedf3dda190862239b0eb22c8",
        "amount": "0.00000000"
    }
],
"transactions": [
    {
        "direction": 2,
        "hashlow":
"00000000000000000caa39e1e618a78e3e6633a13957b35b0825c0029bbde3650",
        "address": "UDbeuykAXIKwNXuVEzpj5uN4imEenqPK",
        "amount": "64.200000000",
        "time": 1739376191999,
        "remark": "xdagJ_test02_179"
    }
]
},
"id": "1"
}

```

2.2 xdag_getBlockByNumber(String BlockHeight, int Page, int PageSize)

Used to return a block based on block number & page & page size.

Req

```

curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBlockByNumber","params":
[["BlockHeight","Page","PageSize"],"id":1}]'

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBlockByNumber","params":
["2650572","1","2"],"id":1}'

```

3. xdag_blockNumber

Used to return the current main block height

Req

```

curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_blockNumber","params":
[],"id":1}'

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_blockNumber","params":
[],"id":1}'

```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": "3145719",
  "id": "1"
}
```

4. xdag_coinbase

Used to return the current pool miner

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_coinbase","params":[],"id":1}'

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_coinbase","params":[],"id":1}'
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": "JJPqDSdKcQQoHoUD57ezFSKMXoJCDt3raU",
  "id": "1"
}
```

5. xdag_getBalance

Used to return the balance of someone

Req

```
# getBalance by accountAddress
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBalance","params":
["AccountAddress"],"id":1}'

## example 1:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBalance","params":
["K5q0ews/ma110QLUzePetOdU+EwYKrud"],"id":1}'

# getBalance by blockAddress
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBalance","params":
["BlockAddress"],"id":1}'

## example 2:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getBalance","params":
["UDbeuykAXIKwNXuVEzpj5uN4imEenqPK"],"id":1}'
```

Resp

```
# Response of example 1:
{
  "jsonrpc": "2.0",
  "result": "10.000000000",
  "id": "1"
}

# Response of example 2:
{
  "jsonrpc": "2.0",
  "result": "0.000000000",
  "id": "1"
}
```

6. xdag_getTransactionNonce

Used to return the balance of someone

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getTransactionNonce","params":
["accountAddress"],"id":1}'

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_getTransactionNonce","params":
["8PKBGqTAAanzSG8qbZGzDhoUihgC3cyvCH"],"id":1}'
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": "1",
  "id": "1"
}
```

7. xdag_syncing

Check whether the current synchronization is completed

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_syncing\",\"params\": [], \"id\":\"1\"}"

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data "{\"jsonrpc\":\"2.0\",\"method\":\"xdag_syncing\",\"params\": [], \"id\":\"1\"}"
```

Resp

```
# if sync done
{
  "jsonrpc": "2.0",
  "result": {
    "currentBlock": "3145729",
    "highestBlock": "3145729",
    "isSyncDone": true
  },
  "id": "1"
}

# else
{
  "jsonrpc": "2.0",
  "result": {
    "currentBlock": null,
    "highestBlock": null,
    "isSyncDone": false
  },
  "id": "1"
}
```

8. xdag_getTotalBalance

Used to return the current balance of this pool

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"xdag_getTotalBalance","params":[],"id":1}'

# example:
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"xdag_getTotalBalance","params":[],"id":1}'
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": "324972.800000000",
  "id": "1"
}
```

9. xdag_getStatus

Used to return the status of the XDAG network

Req

```
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"xdag_getStatus","params":[],"id":1}'

# example:
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"xdag_getStatus","params":[],"id":1}'
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": {
    "nblock": "16252",
    "totalNblocks": "16252",
    "nmain": "3145733",
    "totalNmain": "3145733",
    "curDiff": "0xcd6e16f95f2dea45901ab58240",
    "netDiff": "0xcd6e16f95f2dea45901ab58240",
    "hashRateOurs": "1.547266524860277",
    "hashRateTotal": "1.547266524860277",
    "ourSupply": "1247065152.000000000",
    "netSupply": "1247065152.000000000"
  },
  "id": "1"
}
```

10. xdag_personal_sendTransaction

Used to transfer from pool to other address.

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --data "{
  \"jsonrpc\": \"2.0\",
  \"method\": \"xdag_personal_sendTransaction\",
  \"params\": [
    {
      \"to\": \"toAddress\",
      \"value\": \"value\",
      \"remark\": \"remark\",
      \"password\": \"password\",
      \"id\": 1
    }
  ]
}" #replace password

# example:
curl http://127.0.0.1:9999/ -s -X POST -H "Content-Type: application/json" --data "{
  \"jsonrpc\": \"2.0\",
  \"method\": \"xdag_personal_sendTransaction\",
  \"params\": [
    {
      \"to\": \"PuUubHqqkRaoiJpYCVWJ5pRyCpUGbGKSH\",
      \"value\": \"0.1\",
      \"remark\": \"8888\",
      \"123\",
      \"id\": 1
    }
  ]
}"

# args {TransactionRequest,password}
# TransactionRequest:
## String from; // from can be null, because it is pool default.
## String to;
## String value;
## String remark;
```

Resp

```
# success
{
  \"jsonrpc\": \"2.0\",
  \"result\": {
    \"code\": 0,
    \"result\": [
      \"dqy+26fvbAH2oYXV2zLtYNzNiJO9/O4w\"
    ],
    \"errMsg\": null,
    \"resInfo\": [
      \"dqy+26fvbAH2oYXV2zLtYNzNiJO9/O4w\"
    ]
  },
  \"id\": \"1\"
}

# failed
{
  \"jsonrpc\": \"2.0\",
  \"result\": {
    \"code\": -10000,
    \"result\": null,
    \"errMsg\": \"To address is illegal\",
    \"resInfo\": null
  },
  \"id\": \"1\"
}
```

```
# failed
{
  "jsonrpc": "2.0",
  "result": {
    "code": -10201,
    "result": null,
    "errMsg": "balance not enough",
    "resInfo": null
  },
  "id": "1"
}
```

11. xdag_personal_sendSafeTransaction

Used to transfer from pool to other address. When making a transfer, you need to first query the transaction nonce and use the transaction nonce as an input parameter.

Req

```
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json"
--data "
{
  \"jsonrpc\": \"2.0\",
  \"method\": \"xdag_personal_sendSafeTransaction\",
  \"params\": [
    [
      \"to\": \"toAddress\",
      \"value\": \"value\",
      \"remark\": \"remark\",
      \"nonce\": \"nonce\",
      \"password\": \"password\"
    ],
    \"id\": 1
  ]
}
```

example:

```
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json"
--data "
{
  \"jsonrpc\": \"2.0\",
  \"method\": \"xdag_personal_sendSafeTransaction\",
  \"params\": [
    [
      \"to\": \"LVQW3xsJbnHtvovAvKyE69SR3bqVgehr8\",
      \"value\": \"10\",
      \"remark\": \"8888\",
      \"nonce\": \"1\",
      \"123\"
    ],
    \"id\": 1
  ]
}
```

args {CallArguments,password}

CallArguments:

- ## String from; from can be null, because it is pool default.
- ## String to;
- ## String value;
- ## String remark;
- ## String nonce;

Resp

```
# success
{
  "jsonrpc": "2.0",
  "result": {
    "code": 0,
    "result": [
      "NbPoK3m8g2ihTJEGSq7rwrDCAXX/ye/I"
    ],
    "errMsg": null,
    "resInfo": [

```

```
        "NbPoK3m8g2ihTJEGSq7rwrnCAXX/ye/I"
    ]
},
"id": "1"
}

# failed
{
    "jsonrpc": "2.0",
    "result": {
        "code": -10000,
        "result": null,
        "errMsg": "To address is illegal",
        "resInfo": null
    },
    "id": "1"
}

# failed
{
    "jsonrpc": "2.0",
    "result": {
        "code": -10201,
        "result": null,
        "errMsg": "balance not enough",
        "resInfo": null
    },
    "id": "1"
}

# failed
{
    "jsonrpc": "2.0",
    "result": {
        "code": -10500,
        "result": null,
        "errMsg": "The nonce passed is incorrect. Please fill in the nonce
according to the query value",
        "resInfo": null
    },
    "id": "1"
}

# failed
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Transaction nonce cannot be empty and must be positive
number",
        "data": null
    },
    "id": "1"
}
```


12. xdag_getRewardByNumber

Used to return the reward of some height

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"xdag_getRewardByNumber","params":["BlockHeight"],"id":1}'
```

```
# example:
```

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --data "{\"jsonrpc\":\"2.0\", \"method\":\"xdag_getRewardByNumber\", \"params\": [\"3000000\"], \"id\":1}"
```

Resp

Response of example:

```
{
  "jsonrpc": "2.0",
  "result": "64.0000000000",
  "id": "1"
}
```

13. xdag_sendRawTransaction

Used to send transfer by raw 512 data

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --data '{"jsonrpc":"2.0","method":"xdag_sendRawTransaction","params":["TransactionRawData"],"id":1}'
```

```
# example:
```

[illegible]**Resp**

```
# success
{
  "jsonrpc": "2.0",
  "result": "L/FTj6KuI6+gsxSHCyov7FH82DMIP45y",
  "id": "1"
}

# failed
{
  "jsonrpc": "2.0",
  "result": "INVALID_BLOCK null",
  "id": "1"
}
```

14. xdag_netType

Used to return the net type xdag running for.

Req

```
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json"
--data '{"jsonrpc":"2.0","method":"xdag_netType","params":
[],"id":1}'

# example:
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json"
--data '{"jsonrpc":"2.0","method":"xdag_netType","params":
[],"id":1}'
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": "mainnet",
  "id": "1"
}
```

15. xdag_netConnectionList

Used to return the net conn list

Req

```
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json"
--data '{"jsonrpc":"2.0","method":"xdag_netConnectionList","params":
[],"id":1}'

# example:
curl http://118.26.111.179:10001/ -s -X POST -H "Content-Type: application/json"
--data '{"jsonrpc":"2.0","method":"xdag_netConnectionList","params":
[],"id":1}'
```

Resp

```
# Response of example:
{
  "jsonrpc": "2.0",
  "result": [
    {
      "nodeAddress": "152.32.129.160:8001",
      "connectTime": 1761295400960,
      "inBound": 0,
      "outBound": 0
    },
    {
      "nodeAddress": "118.26.111.179:8001",
      "connectTime": 1761295400960,
      "inBound": 0,
      "outBound": 0
    }
  ],
  "id": "1"
}
```

16. xdag_poolConfig

Remove to XdagPool-Go, XdagJabandon.

Req

```
curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_poolConfig","params":
[],"id":1}'
```

Resp

```
# Response
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "poolIp": "127.0.0.1",
    "poolPort": 7001,
    "nodeIp": "127.0.0.1",
    "nodePort": 8001,
  }
}
```

```

        "globalMinerLimit":8192,
        "maxConnectMinerPerIp":256,
        "maxMinerPerAccount":256,
        "poolFeeRation":"5.0",
        "poolRewardRation":"5.0",
        "poolDirectRation":"5.0",
        "poolFundRation":"5.0"
    }
}

```

17. ~~xdag_updatePoolConfig~~

Remove to XdagPool-Go, Xdagj abandon.

Req

```

curl http://127.0.0.1:10001/ -s -X POST -H "Content-Type: application/json" --
data '{"jsonrpc":"2.0","method":"xdag_updatePoolConfig","params":
[{"poolFeeRation":"12","poolRewardRation":"11","poolDirectRation":"13
","poolFundRation":"14.2"},"password"],"id":1}' #replace password

# args {CallArguments,password}
# CallArguments:
## String poolRewardRation
## String poolDirectRation
## String poolFundRation
## String poolFeeRation

```

Resp

```

# Response
{
    "jsonrpc":"2.0",
    "id":1,
    "result":"Success"
}

```