

# mcc区块链API开发文档

---

## 功能一览

1. 创建账号
2. 查询余额
3. 批量查询余额
4. 查询交易列表
5. 创建转账交易
6. 查询交易详情
7. 查询token列表
8. 查询token详情
9. 根据区块号查询单个区块信息
10. 根据区块hash查询单个区块信息
11. 获取交易ID (txnId)
12. 根据块号查询块内包含的交易列表
13. 查询token挖矿收益列表
14. 查询可领取的token挖矿收益
15. token挖矿收益领取
16. 查询token挖矿收益详情
17. 申请节点
18. 节点信息查询
19. 申请节点状态查询

---

## api网址

<https://n1.ctk.bz>

在下面的内容中将用**server** 来代表服务器地址

---

### 通用返回参数说明

参数	说明
status	状态标识符： 0 未成功； 1 成功； 2 失败

参数	说明
info	接口返回信息描述，如错误信息等

data	接口返回数据
------	--------

消息体签名说明：

所有的签名消息都需要首先通过接口 17. 获取交易ID (txnId) 来获取 交易ID (txid)、随机数 (nonce) 。

消息体结构为多个参数拼接而成的字符串

转账消息体 (6. 创建转账交易)：

txid + cc + from + to + amount

交易tx、转账货币英文、转账人、收款人、转账数量 拼接而成的字符串。

发布token消息体 (10. 发布token)：

txid + cc + acc

交易tx、发布token的英文名+发布人账户地址 拼接而成的字符串。

领取抵押本金消息体 (24. 领取抵押本金)、领取挖矿奖励 (23. 领取抵押挖矿奖励)、货币抵押 (22. 货币抵押)：

txid + acc

交易tx、抵押账户地址

签名步骤：

- 1) 消息体转为字节数组，然后转为16进制字符串后，转小写，即是需要进行签名的消息体。
- 2) 消息体进行sha3的Keccak 256 hash，得到hash消息。
- 3) 利用私钥进行ECDSA算法签名。

1.创建帐号：

接口路径: server/acc

请求方法: get

参数	是否必填	说明
psw	必填	账号密码

## 返回数据说明

字段	说明
info	钱包地址或错误信息
data	钱包keystore内容
data->address	钱包地址
data->crypto->ciphertext	加密后的账号私钥

## 返回样例

### 成功时

```
{
  "status": "1",
  "info": "0xF1D0dFbC41F3FD3aeD603bef2A80e6d2c6CB8Ec7",
  "data": {
    "address": "f1d0dfbc41f3fd3aed603bef2a80e6d2c6cb8ec7",
    "crypto": {
      "cipher": "aes-128-ctr",
      "ciphertext": "51ea5e39c1faccca12f172e25c8446bcb772e86768c5b4b0c9135b2257a4a300",
      "cipherparams": {
        "iv": "e3873fb38fb0724d2c9d985af229fbaa",
        "kdf": "scrypt",
        "kdfparams": {
          "dklen": 32,
          "n": 262144,
          "p": 1,
          "r": 8,
          "salt": "34f455a5464bf53a85562a17a24919f4f980a8c3f63265096d06c251f1e43a06",
          "mac": "0a59135abf7322ab53be6305b95b4fa0e5babb2affba09a2793d0e050040dd66",
          "id": "c32f0c2a-5744-463d-812f-66bf7aaafb9f",
          "version": 3
        }
      }
    }
  }
}
```

### 失败时

```
{
  "status": "0",
  "info": "参数psw不可空",
  "data": ""
}
```

## 2. 查询余额：

接口路径: server/qry

请求方法: get

参数	是否必填	说明
cmd	必填	固定填写qryBal
args	必填	钱包地址

参数	是否必填	说明
cc	必填	token名称

返回数据说明

字段	说明
info	错误信息
data	钱包余额

### 返回样例

成功时

```
{"status":"1","info":"","data":999922179}
```

失败时

```
{"status":"0","info":"SendTransactionProposal failed: Transaction processor (peer0.org1.daima.mobi:7051) returned error 'rpc error: code = Unknown desc = chaincode error (status: 500, message: The parameter cannot be empty.)' for txID '50a17e9de1a90587cd9b6efb004c0881be8eabc963f5bd47d754b1ad60921b65',"data":""}
```

### 3.批量查询余额：

接口路径: server/bal

请求方法: get

参数	是否必填	说明
arg	必填	钱包地址与token英文名对应的Json串
--	--	json串字段说明：
cclist	必填	对应的json串数组
cc	必填	token英文名

参数	是否必填	说明
addr	必填	查询账户地址

返回参数data说明:

返回的数据是一个json格式串，其中key是由参数串中的cc+下划线"+"addr拼接而成。value是当前查询地址的钱包余额

### 返回样例

#### 成功时

```
{"status": "1", "info": "", "data": {"dm_0x5fe071ffc42eeb0535af2b7dd96cdf8a89b8265d": "996421.270000000000", "dm_0x5fe071ffc42eeb0535af2b7dd96cdf8a89b8265d": "57500"}}
```

#### 失败时

```
{"status": "0", "info": "SendTransactionProposal failed: Transaction processor (peer0.org1.daima.mobi:7051) returned error 'rpc error: code = Unknown desc = chaincode error (status: 500, message: The parameter cannot be empty.)' for txID '50a17e9de1a90587cd9b6efb004c0881be8eabc963f5bd47d754b1ad60921b65'", "data": ""}
```

## 4. 查询交易列表 :

接口路径: server/trade

请求方法: get

参数	是否必填	说明
method	必填	固定填写history
cc	选填	token名称
acc	必填	钱包地址
page	必填	页数

参	是否必填	说明
number	必填	每页显示记录数(最多100)
status	选填	记录类型（全部0，转出1，转入2，失败3；默认不填查询全部。）

#### 返回数据说明

字段	说明
info	错误信息
data	json数组
data[n]->out	转出账号
data[n]->in	转入账号
data[n]->txId	交易id
data[n]->token	token名称
data[n]->flag	交易类型 1转出 0转入
data[n]->sum	数量合计(含手续费)
data[n]->fee	手续费
data[n]->amount	交易数量(不含手续费)
data[n]->time	时间(字符串)
data[n]->authority_sign	司法鉴定编号(可能为空)
data[n]->type	交易类型 1 表示token发布扣费，0表示正常转账
data[n]->blockNum	所属块号
data[n]->remark	转账备注
data[n]->error	错误日志
data[n]->status	状态 0处理中,1成功,2失败

字段	说明
返回样例	

#### 成功时

```
{
  "status": "1",
  "info": "",
  "data": [
    {
      "out": "0x5fe071ffc42eeb0535af2b7dd96cdf8a89b8265d",
      "in": "0x468a2acc04a9d01f462eae7e155d7a8bdf2694d4",
      "txId": "caab97fe743f0cc159187cdec48da0b77f32ee6803c52d3703b779781c63d9db",
      "token": "dm",
      "blockNum": "3319",
      "fee": "0",
      "amount": "100",
      "remark": " 转账备注 ",
      "time": "2018-04-30 19:26:59",
      "status": "1",
      "error": "错误日志"
    }
  ]
}
```

#### 失败时

```
{
  "status": "0",
  "info": "出错了",
  "data": ""
}
```

### 5.创建转账交易：

**接口路径:** server/ivk

**请求方法:** post

参数	是否必填	说明
cc	必填	token名称
from	必填	转出地址
to	必填	转入地址
amo	必填	转账数量

参数	是否必填	说明
sign	必填	签名
remark	选填	转账备注
txid	必填	交易ID
nonce	必填	交易随机数

返回数据说明

字段	说明
info	错误信息
data	
---	

返回样例

成功时

```
{ "status": "1", "info": "success", "data": "b31a2f170c7626d4a2405934fdad375688547a6cbaf3e66862f3ef2126730384" }
```

失败时

```
{ "status": "0", "info": "出错了", "data": "" }
```

6. 查询交易详情：

接口路径: server/trade

请求方法: get

参数	是否必填	说明
method	必填	固定填写txrst
tx	必填	交易txid

返回数据说明



字段	说明
info	错误信息，如果tx查询不到，则展示invalidTxid
data	json
data->out	转出账号
data->in	转入账号
data->txId	交易id
data->token	token名称
data->sum	数量合计(含手续费)
data->fee	手续费
data->amount	交易数量(不含手续费)
data->time	时间(字符串)
data->authority_sign	司法鉴定编号(可能为空)
data>type	交易类型 1 表示token发布扣费，0表示正常转账
data->blockNum	所属块号
data->remark	转账备注
data->error	错误信息
data->status	0正在处理中 1交易成功 2交易失败

## 返回样例

### 成功时

```
{
  "status": "1",
  "info": "success",
  "data": {
    "out": "转出账号",
    "in": "转入账号",
    "txId": "交易id",
    "token": "token名称",
    "fee": "手续费",
    "amount": "交易数量",
    "time": "时间",
    "blockNum": "所属块号",
    "remark": "转账备注",
    "error": "错误信息",
    "status": "状态"
  }
}
```

### 失败时

```
{
  "status": "0",
  "info": "参数tx不可空",
  "data": ""
}
```

## 7.查询token列表：

接口路径: server/tklist

请求方法: get

参	是否必填	说明
arg	选填	选填参数，如果不填，则查询所有token列表，填写后，进行模糊匹配查找token列表
page	选填	选填参数，如果不填，默认查询第一页内容（100条记录为一页）
status	选填	不填查询所有状态token，0 未完成，1已完成，2错误

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	json数组
data[n]->name	token中文名称
data[n]->cc	token英文名称
data[n]->creator	token创建者
data[n]->total	货币总量
data[n]->mineral	年化自动挖矿百分比
data[n]->decimal	小数点精确位数
data[n]->award	年化率(自动挖矿)
data[n]->logo	token的logo(http url)
data[n]->desc	token说明

字段	说明
data[n]->url	官方网址
data[n]->status	token实例化状态，0未完成，1成功，2失败
data[n]->error	失败原因

### 返回样例

#### 成功时

```
{
  "status": "1",
  "info": "success",
  "data": [
    {
      "id": "3",
      "name": "hu",
      "chaincode": "hu",
      "policy": "AND ('Org1MSP.member', 'Org2MSP.member')",
      "orgid": "org1,org2",
      "creator": "0x4B75542331E2B5Bc1d6804Ce639A6bD712EB16F9",
      "desc": "描述信息",
      "total": "",
      "url": "dsfdf",
      "decimal": "6",
      "logo": "./chaincode/token_logo/0x4B75542331E2B5Bc1d6804Ce639A6bD712EB16F9_hu.jpg",
      "version": "1.0",
      "status": "1",
      "error": ""
    }
  ]
}
```

#### 失败时

```
{
  "status": "0",
  "info": "出错了",
  "data": ""
}
```

### 8. 查询token详情：

**接口路径:** server/tkinfo

**请求方法:** get

参数	是否必填	说明
cc	必填	token名称

#### 返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	json

字段	说明
data->name	token中文名称
data->creator	token创建者
data->cc	token英文名称
data->total	货币总量
data->mineral	年化自动挖矿百分比
data->decimal	小数点精确位数
data->award	预挖数量
data->logo	token的logo(http url)
data->desc	token说明
data->url	官方网址

## 返回样例

### 成功时

```
{
  "status": "1",
  "info": "success",
  "data": {
    "name": "爆发型货币",
    "chaincode": "token2",
    "creator": "0xbd0f20927c86ee910961d0e98db20d8760464461",
    "desc": "",
    "total": "10000000",
    "url": "http://daima.mobi",
    "decimal": "10",
    "logo": "/img?id=0xbd0f20927c86ee910961d0e98db20d8760464461_token_001.jpg",
    "mineral": "1000000",
    "award": "35",
    "email": "bossyt@163.com"
  }
}
```

### 失败时

```
{
  "status": "0",
  "info": "SendTransactionProposal failed: Transaction processor (peer0.org1.daima.mobi:7051) returned error 'rpc error: code = Unknown desc = could not find chaincode with name 'dm111' - make sure the chaincode dm111 has been successfully instantiated and try again' for txID '68962dbf097e06439c652692d1de74226f8c1456b752b0efa190b96632242af9'",
  "data": ""
}
```

## 9.根据区块号查询单个区块信息：

接口路径: server/qry

请求方法: get

参数	是否必填	说明
cmd	必填	固定填写qryblo
args	必填	区块号码

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	json
data->Number	块号码
data->PreviousHash	前一个区块hash
data->DataHash	hash
data->Creator	创建者
data->Transactions	包含交易数
data->MinerList	矿工列表
data->Timestamp	出块时间
data->Ledgers	账本数据
data->Ledgers[n]->token	token英文名称
data->Ledgers[n]->key	账本Key
data->Ledgers[n]->readValue	账本更新前的值
data->Ledgers[n]->writeValue	账本更新后的值

返回样例

## 成功时

```
{
  "status": "1",
  "info": "",
  "data": {
    "Number": "199",
    "MinerList": "0x4a8a0b617187c3b5355705af3e48107705b40c01,0xd56f93963ebe0811837c2348ee8aac26213f8928",
    "PreviousHash": "AQwjoC_k7vaEnpRNCMF9N6o93C-fCEeu6_YloL9Fh-Y",
    "DataHash": "Ktsa6TpR_HYm-siBcKwXvqhUH3zfr--fh_P1qCrsAQQ",
    "Creator": "CgpPcmRlcmVyTVNQEusFLS0tLS1CRUdJTtAtLS0tLQpNSUIDQ3pDQ0FiR2dBd0lCQWdJUkFLaml4SmFiMzdoMndDUeItcmdlZ05jd0NnWUllb1pJemowRUF3SXdaekVMck1Ba0dBMVVFQmhNQ1ZWtXhFekFSQmdOVkJBZ1RDa05oYkdsbWlzSnVhV0V4RmpBVUJnTlZCQWNURFZOaGJpQkcKY21GdVkybHpZMjh4RXpBUklnTlZCQW9UQ21SaGFxMWhMbTF2WW1reEZqQVVCZ05WQkFNVERXTmhMbVJoYVcxaApMbTF2WW1rd0hoY05NVGd3TkRFNE1EazFNekV4V2hjTk1qZ3dOREUxTURrMU16RXhXakJZTVFzd0NRWURWUUVFHckV3SIZVekVUTUJFR0ExVUVDQk1LUTJGc2FXWnZjbTVwWVRfV01CUUdBMVVFQnhNTlUyRnVJRVP5WVc1amFYTmoKYnpFY01Cb0dBMVVFQXhNVGIzSmtaWEpsY2pFdVpHRnBiV0V1Ylc5aWU01FUQlPnQk1HQnlxR1NNNDIBZ0VHQ0NxrWpTTTQ5QXdfSEEWsUFCQVIvTE1VdTNjQ01SR3kwT3kwSm1KT2FPV2FaaWtvbnQyNIRZZmxyNEFueEZWZ1VyL2M3CIVYUWhKUVU1cVdqbkNRT0hsNUhKY0lrOGt0cFFpM0kxYmJHaIRUQkxNQTRHQTfVZER3RUlvd1FFQXdJSGdEQU0KQmdOVkhSTUJBZjhFQWpBQU1Dc0dBMVVksXhRa01DS0FJR1hGcnFJaXZvOGVJbXR0ZmllbHR5M1o5bIJRa3NUMwpWUHJMUWhYbC85OERNQW9HQ0Nxr1NNNDICQU1DQTBnQU1FVUNJUUN5ZVFleVo4UUUxeW9jcWFLUTM0N0w2bTJTCih3empWY0ZvM0gyUTd0N0tBd0lnR09vaHVwS283N2VidVlIZjlteDRRbTNIQXhjZUdTbzZQYWJsK2luYXVhWT0KLS0tLS1FTkQgLS0tLS0K",
    "Transactions": "1",
    "Timestamp": "2018-05-29 09:47:52",
    "Ledgers": [
      {
        "token": "",
        "key": "",
        "readValue": "",
        "writeValue": "",
        "Timestamp": ""
      }
    ]
  }
}
```

```
},
{
  "token": "",
  "key": "",
  "readValue": "",
  "writeValue": "",
  "Timestamp": ""
},
{
  "token": "",
  "key": "",
  "readValue": "",
  "writeValue": "",
  "Timestamp": ""
},
{
  "token": "",
  "key": "",
  "readValue": "",
  "writeValue": "",
  "Timestamp": ""
},
{
  "token": "",
  "key": "",
  "readValue": "",
  "writeValue": "",
  "Timestamp": ""
},
{
  "token": "",
  "key": "",
  "readValue": "",
  "writeValue": "",
  "Timestamp": ""
}
]
```

失败时

```
{"status":"0","info":"出错了","data":""}
```

## 10. 根据区块hash查询单个区块信息：

接口路径: server/qry

请求方法: get

参数	是否必填	说明
cmd	必填	固定填写qryhash
args	必填	区块hash

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	json
data->Number	块号码
data->PreviousHash	前一个区块hash
data->DataHash	hash
data->Creator	创建者
data->Transactions	包含交易数
data->MinerList	矿工列表
data->Timestamp	出块时间
data->Ledgers	账本数据
data->Ledgers[n]->token	token英文名称
data->Ledgers[n]->key	账本Key



字段	说明
data->Ledgers[n]->readValue	账本更新前的值
data->Ledgers[n]->writeValue	账本更新后的值

## 返回样例

成功时

```
{
  "status": "1",
  "info": "",
  "data": {
    "Number": "199",
    "MinerList": "0x4a8a0b617187c3b5355705af3e48107705b40c01,0xd56f93963ebe0811837c2348ee8aac26213f8928",
    "PreviousHash": "AQwjoC_k7vaEnpRNCMF9N6o93C-fCEeu6_YloL9Fh-Y",
    "DataHash": "Ktsa6TpR_HYm-siBcKwXvqhUH3zfr--fh_P1qCrsAQQ",
    "Creator": "CgpPcmRlcmVyTVNQEusFLS0tLS1CRUdJTtAtLS0tLQpNSUIDQ3pDQ0FiR2dBd0lCQWdJUkFLaml4SmFiMzdoMndDUETlcmdlZ05jd0NnWUlb1pJemowRUF3SXdaekVMck1Ba0dBMVVFQmhNQ1ZWtXhFekFSQmdOVkJBZ1RDa05oYkdsbWlzSnVhV0V4RmpBVUJnTlZCQWNURFZOaGJpQkcKY21GdVkybHpZMjh4RXpBUklnTlZCQW9UQ21SaGFxMWhMbTF2WW1reEZqQVVCZ05WQkFNVERXTmhMbVJoYVcxaApMbTF2WW1rd0hoY05NVGd3TkRFNE1EazFNekV4V2hjTk1qZ3dOREUxTURrMU16RXhXakJZTVFzd0NRWURWUUVFHCKv3SIZVekVUTUJFR0ExVUVDQk1LUTJGc2FXWnZjbTVwWVRfV01CUUdBMVVFQnhNTlUyRnVJRVP5WVc1amFYTmoKYnpFY01Cb0dBMVVFQXhNVGlzSmtaWEpsY2pFdVpHRnBiV0V1Ylc5aWFUQlplNQk1HQnlxR1NNNDIBZ0VHQ0NxRwpTTTQ5QXdfSEEWsUFCQVIvTE1VdTNjQ01SR3kwT3kwSm1KT2FPV2FaaWtvbnQyNIRZZmxyNEFueEZWZ1VyL2M3CIVYUWhKUVU1cVdqbkNRT0hsNUhKY0lrOGt0cFFpM0kxYmJHaIRUQkxNQTRHQTfVZER3RUlvd1FFQXdJSGdEQU0KQmdOVkhSTUJBZjhFQWpBQU1Dc0dBMVVKSXc0R01DS0FJR1hGcnFJaXZvOGVJbXR0ZmllbHR5M1o5bIJRa3NUMwpWUHJMUWhYbC85OERNQW9HQ0NxR1NNNDICQU1DQTBnQU1FVUNJUUN5ZVFleVo4UUUxeW9jcWFLUTM0N0w2bTJTCih3empWY0ZvM0gyUTd0N0tBd0lnR09vaHVwS283N2VidVlIZjlteDRRbTNIQXhjZUdUbzZQYWJsK2luYXVvaWT0KLS0tLS1FTkQgLS0tLS0K",
    "Transactions": "1",
    "Timestamp": "2018-05-29 09:47:52",
  }
}
```

```
"Ledgers":[
{
"token": "",
"key": "",
"readValue": "",
"writeValue": "",
"Timestamp": ""
},
{
"token": "",
"key": "",
"readValue": "",
"writeValue": "",
"Timestamp": ""
},
{
"token": "",
"key": "",
"readValue": "",
"writeValue": "",
"Timestamp": ""
},
{
"token": "",
"key": "",
"readValue": "",
"writeValue": "",
"Timestamp": ""
},
{
"token": "",
"key": "",
"readValue": ""
}
```

```
"writeValue":"","  
"Timestamp":"","  
}  
]  
}  
}
```

#### 失败时

```
{"status":"0","info":"出错了","data":""}
```

### 11. 获取交易ID (txnid) :

接口路径: server/crtTx

请求方法: get

参数	是否必填	说明
cnt	选填	获取的数量

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	
data[n]->txId	交易ID
data[n]->nonce	随机数

### 返回样例

#### 成功时

```
{"status":"0","info":"success","data":  
[{"txnid":"f0c54cdab34fc366b2e959335b88a4d49dc8e0282ed56fcf2dadd71  
7b8be911e","Nonce":"DlaPpcvyD4HzXPRDGxuAyD9ma8rsX/Sp"},
```

```
{"txnID":"1bec40a2a1cb87641b3e73e7c67783bbe4a36c8c4f676fbdaec236ef11e75dd4","Nonce":"bfpxru/vgWw3Rg4/L9r9HA57hqpdFsXn"}]}
```

失败时

```
{"status":"0","info":"参数cnt非数字! ","data":""}
```

## 12. 根据块号查询块内包含的交易列表:

接口路径: server/trade

请求方法: get

参数	是否必填	说明
method	必填	固定填写qryTxByBlk
blk	必填	块号

返回数据说明

字段	说明
info	错误信息
data	json
data[n]->black	黑洞费用
data[n]->blockNum	块号
data[n]->chaincode	token名称
data[n]->facc	转账人
data[n]->tacc	收款人
data[n]->minerNormal	普通矿工
data[n]->minerSuper	超级矿工
data[n]->amount	交易数量(不含手续费)
data[n]->time	时间(字符串)

字段	说明
data[n]->authority_sign	司法鉴定编号(可能为空)
data[n]>type	交易类型 1 表示token发布扣费，0表示正常转账
data[n]->sMinerFee	超级矿工费
data[n]->nMinerFee	普通矿工费
data[n]->remark	转账备注
data[n]->error	错误信息
data[n]->status	0,3正在处理中 1交易成功 2交易失败

## 返回样例

### 成功时

```
{
  "status": "0",
  "info": "",
  "data": [
    {
      "authority_sign": "16b2eaad72090ecfdf03cc38bb44bc07",
      "black": "0.1000000000",
      "blockNum": "191",
      "chaincode": "dm",
      "error": "MVCC_READ_CONFLICT",
      "facc": "0xbd0f20927c86ee910961d0e98db20d8760464461",
      "invokeTime": "2018-05-29 09:02:08",
      "minerNormal": "0xd56f93963ebe0811837c2348ee8aac26213f8928",
      "minerSuper": "0x468a2acc04a9d01f462eae7e155d7a8bdf2694d4",
      "nMinerFee": "0.4500000000",
      "remark": "lui9rOi0pil=",
      "sMinerFee": "0.4500000000",
      "status": "2",
      "tacc": "0x8fc2e72e0532d50addd67ca3bf120f645d9f9239",
      "transAmou": "0.9900000000",
      "txId": "3a0e45ea6f70f03ef383f4baa9b02001f374bbf422cb14189cdc19d6a2"
    }
  ]
}
```

```
ae7303"
},
{
  "authority_sign": "102d7aeac4b3e48cb80579ccc17cc8e8",
  "black": "0.1000000000",
  "blockNum": "191",
  "chaincode": "dm",
  "error": "MVCC_READ_CONFLICT",
  "facc": "0xbd0f20927c86ee910961d0e98db20d8760464461",
  "invokeTime": "2018-05-29 09:02:08",
  "minerNormal": "0xcef4031161eb96e911551352e1008d0739c1cb17",
  "minerSuper": "0x4a8a0b617187c3b5355705af3e48107705b40c01",
  "nMinerFee": "0.4500000000",
  "remark": "lui9rOi0pil=",
  "sMinerFee": "0.4500000000",
  "status": "2",
  "tacc": "0x8fc2e72e0532d50addd67ca3bf120f645d9f9239",
  "transAmou": "0.9900000000",
  "txId": "762055edc9844b9ba1186570217c05d68fc7e321cf18c5079360959b5c6f869d"
},
{
  "authority_sign": "8f4c4dbf86ba51cddfd179ecf716579f",
  "black": "0.1000000000",
  "blockNum": "191",
  "chaincode": "dm",
  "error": "VALID",
  "facc": "0xbd0f20927c86ee910961d0e98db20d8760464461",
  "invokeTime": "2018-05-29 09:02:08",
  "minerNormal": "0xad846479c9db19d1dc42e6decd9b38a6ab3967c9",
  "minerSuper": "0x22dcf8e4113af996c7bd5c91ff0162f4b0c85f54",
  "nMinerFee": "0.4500000000",
  "remark": "lui9rOi0pil=",
  "sMinerFee": "0.4500000000",
  "status": "1",
  "tacc": "0x8fc2e72e0532d50addd67ca3bf120f645d9f9239",
  "transAmou": "0.9900000000",
  "txId": "b22113767271af8bf1f956bc95fd2607d3b3832fca48855d9885c67ae1bd16cc"
}
```

```
]
}
```

失败时

```
{"status":"0","info":"参数blk不可空","data":""}
```

### 13. 查询token挖矿收益列表:

接口路径: server/tokenawdlist

请求方法: get

参数	是否必填	说明
token	必填	token英文名
acc	必填	账号地址

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	
data[n]->draw_earnings	挖矿收益
data[n]->draw_time	挖矿时间

### 返回样例

成功时

```
{"status":"1","info":"","data":
[{"draw_earnings":"44378.0817480000","draw_time":"2018-06-16"},
{"draw_earnings":"0","draw_time":"2018-05-25"},
{"draw_earnings":"19711.5374178424","draw_time":"2018-06-16"}]}
```

失败时

`{"status":"0","info":"签名校验失败! ","data":""}`

**14. 查询可领取的token挖矿收益:**

**接口路径:** server/awardpreview

**请求方法:** get

参数	是否必填	说明
token	必填	token英文名
acc	必填	账号地址

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	收益数值

**返回样例**

成功时

`{"status":"1","info":"","data":"100"}`

失败时

`{"status":"0","info":"","data":""}`

**15.token挖矿收益领取:**

**接口路径:** server/award

**请求方法:** get

参数	是否必填	说明
txid	必填	交易ID



参数	是否必填	说明
nonce	必填	随机数
acc	必填	账号地址
cc	必填	token英文名
sign	必填	签名

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	

返回样例

成功时

```
{"status":"1","info":"","data":""}
```

失败时

```
{"status":"0","info":"签名校验失败! ","data":""}
```

16. 查询token挖矿收益详情:

接口路径: server/tokenawddetail

请求方法: get

参数	是否必填	说明
txid	必填	交易ID
nonce	必填	随机数
acc	必填	账号地址
cc	必填	token英文名

参数	是否必填	说明
sign	必填	签名

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	
undraw_earnings	未领取收益
draw_earnings	已领取收益
mineral	年化自动挖矿百分比
period	抵押周期(月,无抵押则为空)
period_end	抵押结束时间（无抵押则为空）
awardlist	奖励列表

返回样例

成功时

```
{
  "status": "1",
  "info": "",
  "data": {
    "undraw_earnings": "0",
    "draw_earnings": "0.0000000000",
    "mineral": "10",
    "awardlist": []
  }
}
```

失败时

```
{
  "status": "0",
  "info": "签名校验失败！ ",
  "data": ""
}
```

17. 申请节点:

接口路径: server/applySpareNode

请求方法: get

参数	是否必填	说明
txid	必填	交易ID
nonce	必填	随机数
acc	必填	账号地址
amount	必填	手续费
sign	必填	签名txid+acc

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	

返回样例

成功时

```
{"status":"1","info":"","data":""}
```

失败时

```
{"status":"0","info":"签名校验失败! ","data":""}
```

18. 查看申请节点信息:

接口路径: server/spareNodeSecret

请求方法: get

参数	是否必填	说明
txid	必填	交易ID
acc	必填	账号地址

sign	必填	签名txid+acc
------	----	------------

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	

返回样例

成功时

```
{"status":"1","info":"","data":"67465657465165745667465987456dsfd5657465165745667465657465165745665745645"}
```

失败时

```
{"status":"0","info":"签名校验失败! ","data":""}
```

## 19. 查看申请节点状态:

接口路径: server/checkApplyStatus

请求方法: get

参数	是否必填	说明
acc	必填	账号地址

返回数据说明

字段	说明
status	1: 调用成功 0: 调用失败
info	错误信息
data	

返回样例

成功时

```
{"status":"1","info":"已申请备用节点","data":""}
```

失败时

```
{"status":"0","info":"还未申请备用节点","data":""}
```