# ETM HTTP API文档

[ETM HTTP API文档](#_Toc2780)

[1 API使用说明](#_Toc21479)

[1.1 请求过程说明](#_Toc27056)

[2 接口](#_Toc28806)

[2.1 账户accounts](#_Toc29343)

[2.1.1 登录](#_Toc6903)

[2.1.2 获取账户信息](#_Toc32438)

[2.1.3 获取账户余额](#_Toc17168)

[2.1.4 获取账户公钥](#_Toc21872)

[2.1.5 生成公钥](#_Toc13686)

[2.1.6 根据地址获取其投票列表](#_Toc28582)

[2.1.7 获取受托人手续费设置](#_Toc4923)

[2.1.8 投票](#_Toc20285)

[2.1.9 获取账户排行榜前100名](#_Toc3945)

[2.2 交易transactions](#_Toc30973)

[2.2.1 获取交易信息](#_Toc27108)

[2.2.2 根据id查看交易详情](#_Toc19068)

[2.2.3 根据未确认交易id查看详情](#_Toc6720)

[2.2.4 获取[全网所有]未确认的交易详情](#_Toc29695)

[2.2.5 创建交易](#_Toc29322)

[2.3 区块blocks](#_Toc24255)

[2.3.1 获取特定id的区块详情](#_Toc275)

[2.3.2 获取最新的区块](#_Toc18930)

[2.3.3 获取区块链高度](#_Toc29107)

[2.3.4 获取交易手续费](#_Toc32136)

[2.3.5 获取里程碑](#_Toc29227)

[2.3.6 查看单个区块奖励](#_Toc10121)

[2.3.7 获取区块链当前最大供应值](#_Toc31371)

[2.3.8 区块链状态](#_Toc24927)

[2.4 受托人delegates](#_Toc8644)

[2.4.1 获取受托人总个数](#_Toc1187)

[2.4.2 根据公钥查看哪些人为其投了票](#_Toc10191)

[2.4.3 根据公钥或者用户名获取受托人详情](#_Toc29475)

[2.4.4 获取受托人列表](#_Toc27153)

[2.4.5 获取受托人设置的转账费](#_Toc21558)

[2.4.6 根据公钥查看其锻造情况](#_Toc23529)

[2.4.7 注册受托人](#_Toc6558)

[2.4.8 受托人开启锻造](#_Toc11393)

[2.4.9 受托人关闭锻造](#_Toc28805)

[2.4.10 受托人锻造状态查看](#_Toc19549)

[2.5 节点peers](#_Toc10075)

[2.5.1 获取全网节点信息](#_Toc22593)

[2.5.2 获取节点版本信息](#_Toc6042)

[2.5.3 获取特定ip节点信息](#_Toc23651)

[2.6 同步和加载](#_Toc23228)

[2.6.1 查看本地区块链加载状态](#_Toc14850)

[2.6.2 查看区块同步信息](#_Toc3056)

[2.7 二级密码signatures](#_Toc2773)

[2.7.1 设置二级密码](#_Toc9287)

[2.7.2 获取二级密码设置费](#_Toc30972)

[2.8 多重签名multisignatures](#_Toc11632)

[2.8.1 设置普通账户为多重签名账户](#_Toc30540)

[2.8.2 获取挂起的多重签名交易详情](#_Toc10049)

[2.8.3 非交易发起人对交易进行多重签名](#_Toc25907)

[2.8.4 获取多重签名账户信息](#_Toc7838)

[2.9 点对点传输tansport[安全的api]](#_Toc17031)

[2.9.1 说明](#_Toc10366)

[2.9.2 交易](#_Toc31481)

[2.10 矿机](#_Toc18965)

[2.10.1 获取矿工列表](#_Toc19980)

[2.10.2投选矿工](#_Toc24439)

[2.10.3 获取矿机信息](#_Toc16308)

[2.10.4 获取矿机算力](#_Toc23601)

[2.10.5 获取矿工SCV列表](#_Toc7953)

[2.10.6 获取选取矿工记录](#_Toc23521)

# ****1 API使用说明****

## ****1.1 请求过程说明****

1.1 构造请求数据，用户数据按照ETM提供的接口规则，通过程序生成签名，生成请求数据集合；  
1.2 发送请求数据，把构造完成的数据集合通过POST/GET等提交的方式传递给ETM；  
1.3 ETM对请求数据进行处理，服务器在接收到请求后，会首先进行安全校验，验证通过后便会处理该次发送过来的请求；  
1.4 返回响应结果数据，ETM把响应结果以JSON的格式反馈给用户，具体的响应格式，错误代码参见接口部分；  
1.5 对获取的返回结果数据进行处理；

# ****2 接口****

## ****2.1 账户accounts****

### ****2.1.1 登录****

#### ****2.1.1.1 本地加密后再登陆（推荐使用）****

接口地址：/api/accounts/open2/  
请求方式：post  
支持格式：json  
接口备注：公钥需要根据用户提供的密码在在本地用程序生成

请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publicKey | string | Y | etm账户公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否登陆成功 |
| account | json | 账户信息 |

请求示例：

var EtmJS = require('etm-js'); //etm-js具体安装方法见附录 var publicKey = EtmJS.crypto.getKeys(secret).publicKey; //根据密码生成公钥 // var address = EtmJS.crypto.getAddress(publicKey); //根据公钥生成地址

// 将上面生成的数据通过post提交到etm server

curl -X POST -H "Content-Type: application/json" -k -d '{"publicKey":"bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9"}' http://45.32.248.33:4096/api/accounts/open2/

JSON返回示例：

{

"success": true,

"account": {

"address": "16723473400748954103",

"unconfirmedBalance": 19480000000,

"balance": 19480000000,

"unconfirmedSignature": false,

"secondSignature": true,

"secondPublicKey": "edf30942beb74de5ed6368c792af8665e9636f32a5f1c9377bcdc3b252d3f277",

"multisignatures": [],

"u\_multisignatures": []

},

"latestBlock": {

"height": 111923,

"timestamp": 4446270

},

"version": {

"version": "1.0.0",

"build": "12:11:11 16/08/2016",

"net": "testnet"

}

#### ****2.1.1.2 本地不加密直接登陆（不推荐使用）****

接口地址：/api/accounts/open/  
请求方式：post  
支持格式：json  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否登陆成功 |
| account | json | 账户信息 |

请求示例：

curl -X POST -H "Content-Type: application/json" -k -d '{"secret":"fault still attack alley expand music basket purse later educate follow ride"}' http://45.32.248.33:4096/api/accounts/open/

JSON返回示例：

{

"success": true,

"account": {

"address": "16723473400748954103",

"unconfirmedBalance": 19480000000,

"balance": 19480000000,

"publicKey": "bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9",

"unconfirmedSignature": false,

"secondSignature": true,

"secondPublicKey": "edf30942beb74de5ed6368c792af8665e9636f32a5f1c9377bcdc3b252d3f277",

"multisignatures": [ ],

"u\_multisignatures": [ ]

}

}

### ****2.1.2 获取账户信息****

接口地址：/api/accounts  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| address | string | Y | 用户地址,最小长度：1 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| account | json | 账户信息 |
| latestBlock | json | 最新的区块信息 |
| version | json | 版本相关信息 |

请求示例：

curl -k -X GET http://45.32.248.33:4096/api/accounts?address=16723473400748954103

JSON返回示例：

{

"success": true,

"account": {

"address": "16723473400748954103", //etm地址

"unconfirmedBalance": 19480000000, //未确认和已确认的余额之和，该值大于等于balance

"balance": 19480000000, //余额

"publicKey": "bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9", //公钥

"unconfirmedSignature": false,

"secondSignature": true, //二级签名

"secondPublicKey": "edf30942beb74de5ed6368c792af8665e9636f32a5f1c9377bcdc3b252d3f277", //二级密码公钥

"multisignatures": [],

"u\_multisignatures": []

},

"latestBlock": {

"height": 114480, //区块高度

"timestamp": 4471890

},

"version": {

"version": "1.0.0",

"build": "12:11:11 16/08/2016", //构建日期

"net": "testnet" //区块链类型，是主链还是测试链

}

}

### ****2.1.3 获取账户余额****

接口地址：/api/accounts/getBalance  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| address | string | Y | 用户地址,最小长度：1 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| balance | integer | 余额 |
| unconfirmedBalance | integer | 未确认和已确认的余额之和，该值大于等于balance |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/accounts/getBalance?address=14636456069025293113'

JSON返回示例：

{

"success": true,

"balance": 5281328514990,

"unconfirmedBalance": 5281328514990

}

### ****2.1.4 获取账户公钥****

接口地址：/api/accounts/getPublickey  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| address | string | Y | 用户地址,最小长度：1 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| publicKey | string | 公钥 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/accounts/getPublickey?address=14636456069025293113'

JSON返回示例：

{

"success": true,

"publicKey": "ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7"

}

### ****2.1.5 生成公钥****

接口地址：/api/accounts/generatePublickey  
请求方式：post  
支持格式：json  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| publicKey | string | 公钥 |

请求示例：

curl -k -H "Content-Type: application/json" -X POST -d '{"secret":"fault still attack alley expand music basket purse later educate follow ride"}' 'http://45.32.248.33:4096/api/accounts/generatePublickey'

JSON返回示例：

{

"success": true,

"publicKey": "bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9"

}

### ****2.1.6 根据地址获取其投票列表****

接口地址：/api/accounts/delegates  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| address | string | Y | 投票人地址 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| delegates | Array | 已投票的受托人详情数组 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/accounts/delegates?address=14636456069025293113'

JSON返回示例：

{

"success": true,

"delegates": [{

"username": "wgl\_002",

"address": "14636456069025293113",

"publicKey": "ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7",

"vote": 9901985415600500,

"producedblocks": 1373,

"missedblocks": 6,

"rate": 1,

"approval": "98.54",

"productivity": "99.56"

},

{

"username": "wgl\_003",

"address": "9961157415582672274",

"publicKey": "c292db6ea14d518bc29e37cb227ff260be21e2e164ca575028835a1f499e4fe2",

"vote": 9891995435600500,

"producedblocks": 1371,

"missedblocks": 8,

"rate": 2,

"approval": "98.44",

"productivity": "99.41"

},

{

"username": "wgl\_001",

"address": "1869971419039689816",

"publicKey": "c547df2dde6cbb4508aabcb5970d8f9132e5a1d1c422632da6bc20bf1df165b8",

"vote": 32401577128413,

"producedblocks": 969,

"missedblocks": 8,

"rate": 102,

"approval": "0.32",

"productivity": 0

}]

}

### ****2.1.7 获取受托人手续费设置****

接口地址：/api/accounts/delegates/fee  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| fee | integer | 手续费设置 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/accounts/delegates/fee

JSON返回示例：

{

"success": true,

"fee": 100000000

}

### ****2.1.8 投票****

接口地址：/api/accounts/delegates  
请求方式：put  
支持格式：json  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |
| secondSecret | string | N | etm账户二级密码，最小长度：1，最大长度：100 |
| delegates | Array | 受托人公钥数组，每个公钥前需要加上+或者-号，代表增加/取消对其的投票 |  |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transaction | json | 投票交易详情 |

请求示例：

curl -k -H "Content-Type: application/json" -X PUT -d '{"secret":"call scissors pupil water friend timber spend brand vote obey corn size","publicKey":"3ec1c9ec08c0512641deba37c0e95a0fe5fc3bdf58424009f594d7d6a4e28a2a","delegates":["+fafcd01f6b813fdeb3c086e60bc7fa9bfc8ef70ae7be47ce0ac5d06e7b1a8575"]}' 'http://45.32.248.33:4096/api/accounts/delegates'

JSON返回示例：

{

"success": true,

"transaction": {

"type": 3, //投票的交易类型为3

"amount": 0,

"senderPublicKey": "3ec1c9ec08c0512641deba37c0e95a0fe5fc3bdf58424009f594d7d6a4e28a2a",

"requesterPublicKey": null,

"timestamp": 5056064,

"asset": {

"vote": {

"votes": ["+fafcd01f6b813fdeb3c086e60bc7fa9bfc8ef70ae7be47ce0ac5d06e7b1a8575"]

}

},

"recipientId": null,

"signature": "0bff58c7311fc59b3c8b3ffc236bbfece9850c334fb0c292ab087f78cf9a6c0f4d3e541c501887a2c2ec46294c777e8f7bf7dea9cb7c9a175fdec641bb684f08",

"id": "5630629337798595849",

"fee": 10000000,

"senderId": "15238461869262180695"

}

}

### ****2.1.9 获取账户排行榜前100名****

接口地址：/api/accounts/top  
请求方式：get  
支持格式：无  
请求参数说明：如果不加请求参数则返回持币量前100名账户信息

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| limit | integer | N | 限制结果集个数，最小值：0,最大值：100 |
| offset | integer | N | 步长，最小值0 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| accounts | json | 账户信息元组，每个元素包含地址、余额、公钥 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/accounts/top?limit=5&offset=0' //返回前5名账户信息

JSON返回示例：

{

"success": true,

"accounts": [{

"address": "355198157736313687",

"balance": 4400099900000000, //44000999 ETM

"publicKey": "0b8e120db026d58cbf9d3f392f88eefe3a82a0a3023298b9466d7ed64ff05881"

},

{

"address": "3196144307608101364",

"balance": 3750000020000000,

"publicKey": "988eb82a603dd033f94a4f3b6f9f9ef4a7d3d066607c433e5255d50ea7270720"

},

{

"address": "9248745407080572308",

"balance": 988703397029757,

"publicKey": "02cedc56da08099532e312c5e563e2859bc5b93cc594eb3e5d350f368d681988"

},

{

"address": "15745540293890213312",

"balance": 498186229718623,

"publicKey": "d39d6f26869067473d685da742339d1a9117257fe14b3cc7261e3f2ed5a339e3"

},

{

"address": "8812460086240160222",

"balance": 100704426831866,

"publicKey": "0af92cc32f54d50dd83c4f7de14e71223a57843a40e993bc0813454aa9270053"

}

}

## ****2.2 交易transactions****

### ****2.2.1 获取交易信息****

接口地址：/api/transactions  
请求方式：get  
支持格式：urlencoded  
接口备注：如果请求不加参数则会获取全网所有交易  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| blockId | string | N | 区块id |
| limit | integer | N | 限制结果集个数，最小值：0,最大值：100 |
| type | integer | N | 交易类型,0:普通转账，1:设置二级密码，2:注册受托人，3:投票，4:多重签名，5:DAPP，6:IN\_TRANSFER，7:OUT\_TRANSFER |
| orderBy | string | N | 根据表中字段排序，senderPublicKey:desc |
| offset | integer | N | 步长，最小值0 |
| senderPublicKey | string | N | 发送者公钥 |
| ownerPublicKey | string | N |  |
| ownerAddress | string | N |  |
| senderId | string | N | 发送者地址 |
| recipientId | string | N | 接收者地址,最小长度：1 |
| amount | integer | N | 金额 |
| fee | integer | N | 手续费 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactions | 列表 | 多个交易详情json构成的列表 |
| count | int | 获取到的交易总个数 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/transactions?recipientId=16723473400748954103&orderBy=t\_timestamp:desc&limit=3'

JSON返回示例：

{

"success": true,

"transactions": [{

"id": "17192581936339156329",

"height": "105951",

"blockId": "15051364118100195665",

"type": 0,

"timestamp": 4385190,

"senderPublicKey": "d39d6f26869067473d685da742339d1a9117257fe14b3cc7261e3f2ed5a339e3",

"senderId": "15745540293890213312",

"recipientId": "16723473400748954103",

"amount": 10000000000,

"fee": 10000000,

"signature": "98d65df3109802c707eeed706e90a907f337bddab58cb4c1fbe6ec2179aa1c85ec2903cc0cf44bf0092926829aa5a0a6ec99458f65b6ebd11f0988772e58740e",

"signSignature": "",

"signatures": null,

"confirmations": "31802",

"asset": {

}

},

{

"id": "7000452951235123088",

"height": "105473",

"blockId": "11877628176330539727",

"type": 0,

"timestamp": 4380147,

"senderPublicKey": "fafcd01f6b813fdeb3c086e60bc7fa9bfc8ef70ae7be47ce0ac5d06e7b1a8575",

"senderId": "16358246403719868041",

"recipientId": "16723473400748954103",

"amount": 10000000000,

"fee": 10000000,

"signature": "dc84044d4f6b4779eecc3a986b6507e458cc5964f601ebeb4d3b68a96129813f4940e14de950526dd685ca1328b6e477e6c57e95aeac45859a2ea62a587d0204",

"signSignature": "",

"signatures": null,

"confirmations": "32280",

"asset": {

}

},

{

"id": "14093929199102906687",

"height": "105460",

"blockId": "2237504897174225512",

"type": 0,

"timestamp": 4380024,

"senderPublicKey": "fafcd01f6b813fdeb3c086e60bc7fa9bfc8ef70ae7be47ce0ac5d06e7b1a8575",

"senderId": "16358246403719868041",

"recipientId": "16723473400748954103",

"amount": 10000000000,

"fee": 10000000,

"signature": "73ceddc3cbe5103fbdd9eee12f7e4d9a125a3bcf2e7cd04282b7329719735aeb36936762f17d842fb14813fa8f857b8144040e5117dffcfc7e2ae88e36440a0f",

"signSignature": "",

"signatures": null,

"confirmations": "32293",

"asset": {

}

}],

"count": 3

}

### ****2.2.2 根据id查看交易详情****

接口地址：/api/transactions/get  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| Id | string | Y | 交易id |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactions | json | 交易详情 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/transactions/get?id=14093929199102906687'

JSON返回示例：

{

"success": true,

"transaction": {

"id": "14093929199102906687",

"height": "105460",

"blockId": "2237504897174225512",

"type": 0,

"timestamp": 4380024,

"senderPublicKey": "fafcd01f6b813fdeb3c086e60bc7fa9bfc8ef70ae7be47ce0ac5d06e7b1a8575",

"senderId": "16358246403719868041",

"recipientId": "16723473400748954103",

"amount": 10000000000,

"fee": 10000000,

"signature": "73ceddc3cbe5103fbdd9eee12f7e4d9a125a3bcf2e7cd04282b7329719735aeb36936762f17d842fb14813fa8f857b8144040e5117dffcfc7e2ae88e36440a0f",

"signSignature": "",

"signatures": null,

"confirmations": "34268",

"asset": {

}

}

}

### ****2.2.3 根据未确认交易id查看详情****

接口地址：/api/transactions/unconfirmed/get  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| id | string | Y | 未确认交易id |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transaction | json | 未确认交易详情 |

请求示例：

curl -k -X GET http://45.32.248.33:4096/api/transactions/unconfirmed/get?id=7557072430673853692 //正常情况，该未确认交易存在时间极短0~10秒

JSON返回示例：

{

"success": true,

"transaction": {

"type": 0,

"amount": 10000,

"senderPublicKey": "3ec1c9ec08c0512641deba37c0e95a0fe5fc3bdf58424009f594d7d6a4e28a2a",

"requesterPublicKey": null,

"timestamp": 5082322,

"asset": {

},

"recipientId": "16723473400748954103",

"signature": "3a97f8d63509ef964bda3d816366b8e9e2d9b5d4604a660e7cbeefe210cb910f5de9a51bece06c32d010f55502c62f0f59b8224e1c141731ddfee27206a88d02",

"id": "7557072430673853692",

"fee": 10000000,

"senderId": "15238461869262180695"

}

}

### ****2.2.4 获取[全网所有]未确认的交易详情****

接口地址：/api/transactions/unconfirmed  
请求方式：get  
支持格式：urlencoded  
接口说明：如果不加参数，则会获取全网所有未确认交易  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| senderPublicKey | string | N | 发送者公钥 |
| address | string | N | 地址 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactions | Array | 未确认交易列表 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/transactions/unconfirmed'

JSON返回示例：

{

"success": true,

"transactions": [] //全网目前不存在未确认的交易

}

### ****2.2.5 创建交易****

接口地址：/api/transactions  
请求方式：PUT  
支持格式：json  
接口备注：接收者账户需在web端钱包登陆过  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| amount | integer | Y | 金额，最小值：1，最大值：10000000000000000 |
| recipientId | string | Y | 接收者地址,最小长度：1 |
| publicKey | string | N | 发送者公钥 |
| secondSecret | string | N | 发送者二级密码，最小长度1，最大长度：100 |
| multisigAccountPublicKey | string | N | 多重签名账户公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactionId | string | 交易id |

请求示例：

curl -k -H "Content-Type: application/json" -X PUT -d '{"secret":"unaware label emerge fancy concert long fiction report affair appear decide twenty","amount":1000000,"recipientId":"16723473400748954103"}' 'http://45.32.248.33:4096/api/transactions'

JSON返回示例：

{

"success": true,

"transactionId": "16670272591943275531"

}

## ****2.3 区块blocks****

### ****2.3.1 获取特定id的区块详情****

接口地址：/api/blocks/get  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| id | string | 参数3选1 | 区块id |
| height | string | 参数3选1 | 区块高度 |
| hash | string | 参数3选1 | 区块hash |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| block | json | 区块详情 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/get?id=6076474715648888747'

JSON返回示例：

{

"success": true,

"block": {

"id": "6076474715648888747",

"version": 0,

"timestamp": 4734070,

"height": 140538,

"previousBlock": "16033230167082515105", //上一个区块id

"numberOfTransactions": 0, //交易数

"totalAmount": 0, //交易额

"totalFee": 0,

"reward": 350000000, //奖励

"payloadLength": 0,

"payloadHash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",

"generatorPublicKey": "1d352950c8141e1b35daba4a974a604519d7a2ef3a1ec0a503ce2653646aa052",

"generatorId": "6656029629904254066",

"blockSignature": "a53de66922cdc2f431acd0a474beec7cf7c420a8460b7b7caf84999be7caebb59fb7fbb7166c2c7013dbb431585ea7294722166cb08bf9663abf50b6bd81cd05",

"confirmations": "2",

"totalForged": 350000000

}

}

### ****2.3.2 获取最新的区块****

接口地址：/api/blocks  
请求方式：get  
支持格式：urlencoded  
接口说明：不加参数则获取全网区块详情  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| limit | integer | N | 限制结果集个数，最小值：0,最大值：100 |
| orderBy | string | N | 根据表中字段排序，如height:desc |
| offset | integer | N | 步长，最小值0 |
| generatorPublicKey | string | N | 区块生成者公钥 |
| totalAmount | integer | N | 交易总额，最小值：0，最大值：10000000000000000 |
| totalFee | integer | N | 手续费总额，最小值：0，最大值：10000000000000000 |
| reward | integer | N | 奖励金额，最小值：0 |
| previousBlock | string | N | 上一个区块 |
| height | integer | N | 区块高度 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| blocks | Array | 由区块详情json串构成的数组 |
| count | integer | 区块链高度 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks?limit=2&offset=0&orderBy=height:desc'

JSON返回示例：

{

"success": true,

"blocks": [{

"id": "12634047624004615059",

"version": 0,

"timestamp": 4708080,

"height": 137986,

"previousBlock": "3498191422350401106",

"numberOfTransactions": 0, // 交易数

"totalAmount": 0, // 金额

"totalFee": 0, // 手续费

"reward": 350000000, // 奖励

"payloadLength": 0,

"payloadHash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",

"generatorPublicKey": "44db7bec89ef289d0def257285675ca14f2a947dfd2b70e6b1cff4392ce42ada",

"generatorId": "4925169939071346193",

"blockSignature": "83a2124e3e8201c1a6099b2ac8ab1c117ad34867978add3a90d41a64df9d2ad8fabc9ec14d27a77cd34c08a6479ef684f247c11b1cbbcb0e9767dffc85838600",

"confirmations": "1",

"totalForged": 350000000

},

{

"id": "3498191422350401106",

"version": 0,

"timestamp": 4708070,

"height": 137985,

"previousBlock": "14078155423801039323",

"numberOfTransactions": 0,

"totalAmount": 0,

"totalFee": 0,

"reward": 350000000,

"payloadLength": 0,

"payloadHash": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",

"generatorPublicKey": "500b1ec025cd64d36008341ed8d2508473ecf559be213ca5f9580620a21a592c",

"generatorId": "16006295608945777169",

"blockSignature": "a0b5ed6c94b1f33c4d0f017f21a08357061493392b19e34eeedf274b77c751e3f86c92443280de09ea1754d62fe7ef00e02acbdc3bc0c1063cef344bacaa4f07",

"confirmations": "2",

"totalForged": 350000000

}],

"count": 137986

}

### ****2.3.3 获取区块链高度****

接口地址：/api/blocks/getHeight  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| height | integer | 区块链高度 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/getheight'

JSON返回示例：

{"success":true,"height":140569}

### ****2.3.4 获取交易手续费****

接口地址：/api/blocks/getFee  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| fee | integer | 交易手续费 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/getfee'

JSON返回示例：

{"success":true,"fee":10000000} //手续费为0.1 ETM

### ****2.3.5 获取里程碑****

接口地址：/api/blocks/getMilestone  
请求方式：get  
支持格式：无  
请求参数说明：无  
返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| milestone | integer |  |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/getMilestone'

JSON返回示例：

{"success":true,"milestone":0}

### ****2.3.6 查看单个区块奖励****

接口地址：/api/blocks/getReward  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| reward | integer | 区块奖励 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/getReward'

JSON返回示例：

{"success":true,"reward":350000000} //每个生成一个block奖励3.5 ETM

### ****2.3.7 获取区块链当前最大供应值****

接口地址：/api/blocks/getSupply  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| supply | integer | 全网ETM个数 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/getSupply'

JSON返回示例：

{"success":true,"supply":10049222600000000} //当前testnet共有100492226ETM

### ****2.3.8 区块链状态****

接口地址：/api/blocks/getStatus  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| height | integer | 区块链高度 |
| fee | integer | 交易手续费 |
| milestone | integer |  |
| reward | integer | 区块奖励 |
| supply | integer | 全网ETM个数 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/blocks/getStatus'

JSON返回示例：

{

"success": true,

"height": 140649,

"fee": 10000000,

"milestone": 0,

"reward": 350000000,

"supply": 10049227150000000

}

## ****2.4 受托人delegates****

### ****2.4.1 获取受托人总个数****

接口地址：/api/delegates/count  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| count | integer | 受托人总个数 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/delegates/count'

JSON返回示例：

{"success":true,"count":234}

### ****2.4.2 根据公钥查看哪些人为其投了票****

接口地址：/api/delegates/voters  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publicKey | string | Y | 受托人公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| accounts | Array | 账户json串组成的数组 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/delegates/voters?publicKey=ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7'

JSON返回示例：

{

"success": true,

"accounts": [{

"address": "2918354313445278349",

"publicKey": "4fde4c49f1297d5d3a24b1494204543c4281aff17917ff7ff8ff32da3b4b222f",

"balance": 1338227722727,

"weight": 0.013316660647014596

},

{

"address": "1523444724068322527",

"publicKey": "8a6a61c28dc47541aadf1eecec2175c8f768f2331eea3472b1593bf1aa4e1fb4",

"balance": 2109297623765,

"weight": 0.020989552213127274

},

{

"address": "14483826354741911727",

"publicKey": "5dacb7983095466b9b037690150c3edec0f073815326e33a4744b6d1d50953e2",

"balance": 5135815841470,

"weight": 0.051106336795243436

}

}]

}

### ****2.4.3 根据公钥或者用户名获取受托人详情****

接口地址： /api/delegates/get/  
请求方式：get  
支持格式：urlencoded  
接口备注：通过公钥或者用户名获取受托人信息  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publickey | string | 二选一 | 受托人公钥 |
| username | string | 二选一 | 受托人用户名 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| delegate | json | 委托人详情 |

请求示例：

curl -k -X GET http://45.32.248.33:4096/api/delegates/get?publicKey=bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9

curl -k -X GET http://45.32.248.33:4096/api/delegates/get?username=delegate\_register

JSON返回示例：

{

"success": true,

"delegate": {

"username": "delegate\_register",

"address": "16723473400748954103",

"publicKey": "bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9",

"vote": 0,

"producedblocks": 0,

"missedblocks": 0,

"fees": 0,

"rewards": 0,

"rate": 191,

"approval": 0,

"productivity": 0,

"forged": "0"

}

}

### ****2.4.4 获取受托人列表****

接口地址：/api/delegates  
请求方式：get  
支持格式：urlencoded  
接口说明：如果不加参数则会返回全网受托人列表  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| address | string | N | 受托人地址 |
| limit | int | N | 限制返回结果数据集的个数 |
| offset | integer | N | 步长，最小值：0 |
| orderBy | string | N | 排序字段:排序规则，如:desc |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| delegates | Array | 受托人详情列表 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/delegates?orderby=approval:desc&limit=2' //按照得票率降序排序，取出前2名

JSON返回示例：

{

"success": true,

"delegates": [{

"username": "wgl\_002", //受托人名字

"address": "14636456069025293113", //受托人地址

"publicKey": "ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7", //受托人公钥

"vote": 9901984015600500, //得票数

"producedblocks": 1371, //生成的区块数

"missedblocks": 6, //丢失的区块数

"fees": 12588514990,

"rewards": 276850000000, //已经得到的奖励

"rate": 1,

"approval": 98.54, //得票率

"productivity": 99.56, //生产率

"forged": "289438514990" //锻造产生的所有奖励

},

{

"username": "wgl\_003",

"address": "9961157415582672274",

"publicKey": "c292db6ea14d518bc29e37cb227ff260be21e2e164ca575028835a1f499e4fe2",

"vote": 9891994035600500,

"producedblocks": 1370,

"missedblocks": 8,

"fees": 12355148480,

"rewards": 275100000000,

"rate": 2,

"approval": 98.44,

"productivity": 99.42,

"forged": "287455148480"

}],

"totalCount": 233

}

### ****2.4.5 获取受托人设置的转账费****

接口地址：/api/delegates/fee  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publicKey | string | Y | 受托人公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| fee | integer | 转账费 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/delegates/fee?publicKey=ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7'

JSON返回示例：

{"success":true,"fee":10000000000} //0.1 ETM

### ****2.4.6 根据公钥查看其锻造情况****

接口地址：/api/delegates/forging/getForgedByAccount  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| generatorPublicKey | string | Y | 区块生成者公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| fees | integer | 收取的手续费 |
| rewards | integer | 已获得奖励 |
| forged | integer | 锻造获得的总奖励 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/delegates/forging/getForgedByAccount?generatorPublicKey=ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7'

JSON返回示例：

{

"success": true,

"fees": 12589307065,

"rewards": 285600000000,

"forged": 298189307065

}

### ****2.4.7 注册受托人****

接口地址：/api/delegates  
请求方式：put  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |
| secondSecret | string | N | etm账户二级密码，最小长度：1，最大长度：100 |
| username | string | N | 受托人名字 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transaction | json | 注册受托人交易详情 |

请求示例：

curl -k -H "Content-Type: application/json" -X PUT -d '{"secret":"unaware label emerge fancy concert long fiction report affair appear decide twenty","username":"delegate\_0821"}' 'http://45.32.248.33:4096/api/delegates'

JSON返回示例：

{

"success": true,

"transaction": {

"type": 2, //注册受托人的交易类型为2

"amount": 0,

"senderPublicKey": "3b64f1833e6328043e1f2fee31e638bdaa6dfff5c7eb9c8577a5cefcf11261f2",

"requesterPublicKey": null,

"timestamp": 4737615,

"asset": {

"delegate": {

"username": "delegate\_0821",

"publicKey": "3b64f1833e6328043e1f2fee31e638bdaa6dfff5c7eb9c8577a5cefcf11261f2"

}

},

"recipientId": null,

"signature": "7f8417e8db5f58ddff887c86c789c26b32fd3f01083ef1e3c8d4e18ed16622bf766492d78518c6c7a07aada1c98b1efc36d40c8e09394989dbde229d8e3f8103",

"id": "16351320834453011577",

"fee": 10000000000,

"senderId": "250438937633388106"

}

}

### ****2.4.8 受托人开启锻造****

接口地址：/api/delegates/forging/enable  
请求方式：post  
支持格式：urlencoded //url必须是受托人所在服务器  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| address | string | 受托人地址 |

请求示例：

curl -k -H "Content-Type: application/json" -X POST -d '{"secret":"motion group blossom coral upper warrior pattern fragile sister misery palm detect"}' 'http://localhost:4096/api/delegates/forging/enable'

JSON返回示例：

{"success":true,"address":"16358246403719868041"}

### ****2.4.9 受托人关闭锻造****

接口地址：/api/delegates/forging/disable  
请求方式：post  
支持格式：urlencoded //url必须是受托人所在服务器  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| address | string | 受托人地址 |

请求示例：

curl -k -H "Content-Type: application/json" -X POST -d '{"secret":"motion group blossom coral upper warrior pattern fragile sister misery palm detect"}' 'http://localhost:4096/api/delegates/forging/disable'

JSON返回示例：

{"success":true,"address":"16358246403719868041"}

### ****2.4.10 受托人锻造状态查看****

接口地址：/api/delegates/forging/status  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publicKey | string | Y | 公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| enabled | string | 锻造是否开启 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/delegates/forging/status?publicKey=fafcd01f6b813fdeb3c086e60bc7fa9bfc8ef70ae7be47ce0ac5d06e7b1a8575'

JSON返回示例：

{"success":true,"enabled":false}

## ****2.5 节点peers****

### ****2.5.1 获取全网节点信息****

接口地址：/api/peers  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| state | integer | N | 节点状态,0: ,1:,2:,3: |
| os | string | N | 内核版本 |
| version | string | N | etm版本号 |
| limit | integer | N | 限制结果集个数，最小值：0,最大值：100 |
| orderBy | string | N |  |
| offset | integer | N | 步长，最小值0 |
| port | integer | N | 端口，1~65535 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| peers | Array | 节点信息json构成的数组 |
| totalCount | integer | 当前正在运行的节点个数 |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/peers?limit=1'

JSON返回示例：

{

"success": true,

"peers": [{

"ip": "45.32.19.241",

"port": 4096,

"state": 2,

"os": "linux3.13.0-87-generic",

"version": "1.0.0"

}],

"totalCount": ["54"]

}

### ****2.5.2 获取节点版本信息****

接口地址：/api/peers/version  
请求方式：get  
支持格式：无  
请求参数说明：无参数

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| version | string | 版本号 |
| build | timestamp | 构建时间 |
| net | string | 主链或者测试链 |

请求示例：

curl -k -X GET http://45.32.248.33:4096/api/peers/version

JSON返回示例：

{

"success": true,

"version": "1.0.0",

"build": "12:11:11 16/08/2016",

"net": "testnet"

}

### ****2.5.3 获取特定ip节点信息****

接口地址：/api/peers/get  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| ip | string | Y | 待查询节点ip |
| port | integer | Y | 待查询节点端口，1~65535 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| peer | json |  |

请求示例：

curl -k -X GET 'http://45.32.248.33:4096/api/peers/get?ip=45.32.248.33&port=4096'

JSON返回示例：

{

"success": true,

"peer": {

}

}

## ****2.6 同步和加载****

### ****2.6.1 查看本地区块链加载状态****

接口地址：/api/loader/status  
请求方式：get  
支持格式：无  
请求参数说明：无参数

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| loaded | boole |  |
| blocksCount | integer |  |

请求示例：

curl -k http://45.32.248.33:4096/api/loader/status -X GET

JSON返回示例：

{

"success": true,

"loaded": true,

"blocksCount": 0

}

### ****2.6.2 查看区块同步信息****

接口地址：/api/loader/status/sync  
请求方式：get  
支持格式：无  
请求参数说明：无参数

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| height | int | 区块高度 |

请求示例：

curl -k http://45.32.248.33:4096/api/loader/status/sync -X GET

JSON返回示例：

{

"success": true,

"syncing": false, // 是否在同步中，如果是则为true，目前没有数据可以同步所以为false

"blocks": 0,

"height": 111987

}

## ****2.7 二级密码signatures****

### ****2.7.1 设置二级密码****

接口地址：/api/signatures  
请求方式：put  
支持格式：json  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |
| secondSecret | string | Y | etm账户二级密码，最小长度：1，最大长度：100 |
| multisigAccountPublicKey | string | N | 多重签名账户公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transaction | json | 设置二级密码产生的交易详情 |

请求示例：

curl -k -H "Content-Type: application/json" -X PUT -d '{"secret":"unaware label emerge fancy concert long fiction report affair appear decide twenty","secondSecret":"fault still attack alley expand music basket purse later educate follow ride"}' 'http://45.32.248.33:4096/api/signatures'

JSON返回示例：

{

"success": true,

"transaction": {

"type": 1, //设置二级密码密码的交易类型为1

"amount": 0,

"senderPublicKey": "3b64f1833e6328043e1f2fee31e638bdaa6dfff5c7eb9c8577a5cefcf11261f2",

"requesterPublicKey": null,

"timestamp": 4872315,

"asset": {

"signature": {

"publicKey": "bd1e78c5a10fbf1eca36b28bbb8ea85f320967659cbf1f7ff1603d0a368867b9"

}

},

"recipientId": null,

"signature": "e76d9b25ec0fdaa88b19d59c5a222b7efdc04f738ee05896f55f4e6959229d9b1600ca25aa92fbea176668f3be7c12c506f2091e2b38c52ef0ece7a5d35e240a",

"id": "1614688380530105232",

"fee": 500000000, //设置二级密码密码的手续费为5 ETM

"senderId": "250438937633388106"

}

}

### ****2.7.2 获取二级密码设置费****

接口地址：/api/signatures/fee  
请求方式：get  
支持格式：无  
请求参数说明：无

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| fee | integer | 费用 |

请求示例：

curl -k http://45.32.248.33:4096/api/signatures/fee -X GET

JSON返回示例：

{

"success": true,

"fee": 500000000 //5 ETM

}

## ****2.8 多重签名multisignatures****

### ****2.8.1 设置普通账户为多重签名账户****

接口地址：/api/multisignatures  
请求方式：put  
支持格式：json  
接口说明：返回结果只是生成交易id，还需要其他人签名后该账户才能成功设置成多重签名账户。注册多重签名账户后任意一笔转账都需要多人签名，签名最少个数为min的值（含交易发起人自身）  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |
| secondSecret | string | N | etm账户二级密码，最小长度：1，最大长度：100 |
| min | integer | Y | 多重签名交易账户的任意一笔转账都需要多人签名的最少个数，如果是注册多重签名账户操作，这该值不生效（此时需要所有人都签名）。最小值：2，最大值：16,该值需要小于keysgroup.length+1 |
| lifetime | integer | Y | 多重签名交易的最大挂起时间，最小值：1，最大值：24，暂时不生效 |
| keysgroup | array | Y | 其它签名人的公钥数组，每个公钥前需要加上+或者-号，代表增加/删除多重签名账户，数组最小长度：1，数组最大长度：10 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactionId | string | 多重签名交易的id |

请求示例：

curl -k -H "Content-Type: application/json" -X PUT -d '{"secret":"vanish deliver message evil canyon night extend unusual tell prosper issue antenna","min":2,"lifetime":1,"keysgroup":["+eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97","+d5d7aa157f866c47a2a1e09e2746286ed089fd90976b54fbfa930e87d11609cb"]}' 'http://45.32.248.33:4096/api/multisignatures' //公钥为2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd

JSON返回示例：

{

"success": true,

"transactionId": "17620378998277022323" //返回结果只是生成交易id，还需要其他人签名后该账户才能成功设置成多重签名账户

}

### ****2.8.2 获取挂起的多重签名交易详情****

接口地址：/api/multisignatures/pending  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publicKey | string | Y | 公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactions | Array | 交易json组成的数组 |

请求示例：

curl -k -X GET http://45.32.248.33:4096/api/multisignatures/pending?publicKey=2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd

JSON返回示例：

{

"success": true,

"transactions": [{ //上一步中设置账户为多重签名交易的详情，transactionId: 17620378998277022323

"min": 2,

"lifetime": 1,

"signed": true,

"transaction": {

"type": 4, //4代表注册多重签名账户

"amount": 0,

"senderPublicKey": "2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd",

"requesterPublicKey": null,

"timestamp": 4879978,

"asset": {

"multisignature": {

"min": 2,

"keysgroup": ["+eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97",

"+d5d7aa157f866c47a2a1e09e2746286ed089fd90976b54fbfa930e87d11609cb"],

"lifetime": 1

}

},

"recipientId": null,

"signature": "a42feaccd9f2a4940fc0be1a1580e786b360f189db3154328f307988e75484293eae391f2f9eee489913cc6d15984eb1f5f5a0aa1bf78ea745d5c725f161af08",

"id": "17620378998277022323",

"fee": 1500000000,

"senderId": "3855903394839129841"

}

}]

}

### ****2.8.3 非交易发起人对交易进行多重签名****

接口地址：/api/multisignatures/sign  
请求方式：post  
支持格式：json  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| secondSecret | string | N | etm账户二级密码，最小长度：1，最大长度：100 |
| publicKey | string | N | 公钥 |
| transactionId | string | Y | 交易id |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transactionId | string | 多重签名交易id |

请求示例：

curl -k -H "Content-Type: application/json" -X POST -d '{"secret":"lemon carpet desk accuse clerk future oyster essay seminar force live dog","transactionId":"17620378998277022323"}' 'http://45.32.248.33:4096/api/multisignatures/sign' //公钥为eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97的用户进行签名

JSON返回示例：

{

"success": true,

"transactionId": "17620378998277022323"

} // 此时再次获取pending

curl -k -X GET http://45.32.248.33:4096/api/multisignatures/pending?publicKey=2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd

{

"success": true,

"transactions": [{

"min": 2,

"lifetime": 1,

"signed": true,

"transaction": {

"type": 4,

"amount": 0,

"senderPublicKey": "2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd",

"requesterPublicKey": null,

"timestamp": 4879978,

"asset": {

"multisignature": {

"min": 2,

"keysgroup": ["+eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97",

"+d5d7aa157f866c47a2a1e09e2746286ed089fd90976b54fbfa930e87d11609cb"],

"lifetime": 1

}

},

"recipientId": null,

"signature": "a42feaccd9f2a4940fc0be1a1580e786b360f189db3154328f307988e75484293eae391f2f9eee489913cc6d15984eb1f5f5a0aa1bf78ea745d5c725f161af08",

"id": "17620378998277022323",

"fee": 1500000000,

"senderId": "3855903394839129841",

"signatures": ["b38a161264db2a23e353d3fbc4983562f6343d5ee693144543ca54e2bc67c0f73d1c761b7bfa38b2bb101ac2ab0797b674b1a9964ccd400aaa310746c3494d03"] //新生成的多重签名

}

}]

}

// 公钥为d5d7aa157f866c47a2a1e09e2746286ed089fd90976b54fbfa930e87d11609cb的账户对该注册交易进行签名

curl -k -H "Content-Type: application/json" -X POST -d '{"secret":"chalk among elbow piece badge try van round quality position simple teach","transactionId":"17620378998277022323"}' 'http://45.32.248.33:4096/api/multisignatures/sign'

{"success":true,"transactionId":"17620378998277022323"} // 此时再次获取pending,结果为空

curl -k -X GET http://45.32.248.33:4096/api/multisignatures/pending?publicKey=2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd

{"success":true,"transactions":[]} // 查看该注册交易详情（该交易已广播并写入blockchain）,此时该账户已成功注册成为多重签名账户

curl -k -X GET http://45.32.248.33:4096/api/transactions/get?id=17620378998277022323

{

"success": true,

"transaction": {

"id": "17620378998277022323", //注册账户为多重签名用户的交易id

"height": "157013",

"blockId": "4680888982781013372",

"type": 4,

"timestamp": 4879978,

"senderPublicKey": "2cef5711e61bb5361c544077aa08aebc4d962a1d656571901c48d716382ad4fd",

"senderId": "3855903394839129841",

"recipientId": "",

"amount": 0,

"fee": 1500000000,

"signature": "a42feaccd9f2a4940fc0be1a1580e786b360f189db3154328f307988e75484293eae391f2f9eee489913cc6d15984eb1f5f5a0aa1bf78ea745d5c725f161af08",

"signSignature": "",

"signatures": null,

"confirmations": "26",

"asset": {

}

}

}

### ****2.8.4 获取多重签名账户信息****

接口地址：/api/multisignatures/accounts  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| publicKey | string | Y | 多重签名参与者之一的公钥 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| accounts | Array | 多重签名账户详情 |

请求示例：

curl -k -X GET http://45.32.248.33:4096/api/multisignatures/accounts?publicKey=eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97

JSON返回示例：

{

"success": true,

"accounts": [{

"address": "3855903394839129841", //多重签名账户地址

"balance": 18500000000, //多重签名账户余额

"multisignatures": ["eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97",

"d5d7aa157f866c47a2a1e09e2746286ed089fd90976b54fbfa930e87d11609cb"], //多重签名账户公钥

"multimin": 2, //最少签名个数

"multilifetime": 1,

"multisigaccounts": [{ //签名者账户详情

"address": "13542769708474548631",

"publicKey": "eb48b9ab7c9a34a9b7cdf860265d65b31af774355cabf1b3a387d14a1925dc97",

"balance": 0

},

{

"address": "4100816257782486230",

"publicKey": "d5d7aa157f866c47a2a1e09e2746286ed089fd90976b54fbfa930e87d11609cb",

"balance": 0

}]

}]

}

## ****2.9 点对点传输tansport[安全的api]****

### ****2.9.1 说明****

/peer相关的api，在请求时都需要设置一个header

* key为magic，value为594fe0f3
* key为version，value为’’

### ****2.9.2 交易****

etm系统的所有写操作都是通过发起一个交易来完成的。  
交易数据通过一个叫做etm-js的库来创建，然后再通过一个POST接口发布出去

POST接口规格如下：  
payload为etm-js创建出来的交易数据  
接口地址：/peer/transactions  
请求方式：post  
支持格式：json

#### ****2.9.2.1 设置二级支付密码****

请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| transaction | json | Y | etm-js.signature.createSignature生成的交易数据 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功 |

请求示例：

var etm = require('etm-js');

var transaction = etm.signature.createSignature('measure bottom stock hospital calm hurdle come banner high edge foster cram','erjimimashezhi001') console.log(JSON.stringify(transaction))

{"type":1,"amount":0,"fee":500000000,"recipientId":null,"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","timestamp":5328943,"asset":{"signature":{"publicKey":"27116db89cb5a8c02fb559712e0eabdc298480d3c79a089b803e35bc5ef7bb7b"}},"signature":"71ef98b1600f22f3b18cfcf17599db3c40727c230db817f610e86454b62df4fb830211737ff0c03c6a61ecfd4a9fcb68a30b2874060bb33b87766acf800e820a","id":"15605591820551652547"}

// 将上面生成的设置二级密码的交易数据通过post提交给etm server

curl -H "Content-Type: application/json" -H "magic:594fe0f3" -H "version:''" -k -X POST -d '{"transaction":{"type":1,"amount":0,"fee":500000000,"recipientId":null,"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","timestamp":5328943,"asset":{"signature":{"publicKey":"27116db89cb5a8c02fb559712e0eabdc298480d3c79a089b803e35bc5ef7bb7b"}},"signature":"71ef98b1600f22f3b18cfcf17599db3c40727c230db817f610e86454b62df4fb830211737ff0c03c6a61ecfd4a9fcb68a30b2874060bb33b87766acf800e820a","id":"15605591820551652547"}}' http://45.32.248.33:4096/peer/transactions

JSON返回示例：

{

"success":true //二级密码设置成功

}

#### ****2.9.2.2 转账****

请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| transaction | json | Y | etm-js.transaction.createTransaction生成的交易数据 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功 |

请求示例：

var etm = require('etm-js');

var targetAddress = "16358246403719868041";

var amount = 100\*100000000; //100 ETM

var password = 'measure bottom stock hospital calm hurdle come banner high edge foster cram';var secondPassword = 'erjimimashezhi001';

// 其中password是在用户登录的时候记录下来的，secondPassword需要每次让用户输入// 可以通过user.secondPublicKey来判断用户是否有二级密码，如果没有，则不必输入，以下几个交易类型类似var transaction = etm.transaction.createTransaction(targetAddress, amount, password, secondPassword || undefined); JSON.stringify(transaction)'{"type":0,"amount":10000000000,"fee":10000000,"recipientId":"16358246403719868041","timestamp":5333378,"asset":{},"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","signature":"2d47810b7d9964c5c4d330a53d1382769e5092b3a53639853f702cf4a382aafcff8ef8663c0f6856a23f41c249944f0c3cfac0744847268853a62af5dd8fc90a","signSignature":"dfa9b807fff362d581170b41c56a2b8bd723c48d1f100f2856d794408723e8973016d75aeff4705e6837dcdb745aafb41aa10a9f1ff8a77d128ba3d712e90907","id":"16348623380114619131"}'

// 将上面生成的转账操作的交易数据通过post提交给etm server

curl -H "Content-Type: application/json" -H "magic:594fe0f3" -H "version:''" -k -X POST -d '{"transaction":{"type":0,"amount":10000000000,"fee":10000000,"recipientId":"16358246403719868041","timestamp":5333378,"asset":{},"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","signature":"2d47810b7d9964c5c4d330a53d1382769e5092b3a53639853f702cf4a382aafcff8ef8663c0f6856a23f41c249944f0c3cfac0744847268853a62af5dd8fc90a","signSignature":"dfa9b807fff362d581170b41c56a2b8bd723c48d1f100f2856d794408723e8973016d75aeff4705e6837dcdb745aafb41aa10a9f1ff8a77d128ba3d712e90907","id":"16348623380114619131"}}' http://45.32.248.33:4096/peer/transactions

JSON返回示例：

{

"success":true //转账成功

}

#### ****2.9.2.3 注册受托人****

请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| transaction | json | Y | etm-js.delegate.createDelegate生成的交易数据 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功 |

请求示例：

var etm = require('etm-js');

var password = 'measure bottom stock hospital calm hurdle come banner high edge foster cram';

var secondPassword = 'erjimimashezhi001';var userName = 'zhenxi\_test';

var transaction = etm.delegate.createDelegate(password, userName, secondPassword || undefined); JSON.stringify(transaction) '{"type":2,"amount":0,"fee":10000000000,"recipientId":null,"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","timestamp":5334485,"asset":{"delegate":{"username":"zhenxi\_test","publicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f"}},"signature":"a12ce415d2d21ab46e4c1b918b8717b1d351dd99abd6f2f94d9a1a7e1f32b697f843a05b1851cb857ea45a2476dce592f5ddd612c00cd44488b8b610c57d7f0a","signSignature":"35adc9f1f37d14458e8588f9b4332eedf1151c02480159f64a287a4b0cbb59bfe82040dfec96a4d9560bae99b8eaa1799a7023395db5ddc640d95447992d6e00","id":"12310465407307249905"}'

// 将上面生成的注册受托人的交易数据通过post提交给etm server

curl -H "Content-Type: application/json" -H "magic:594fe0f3" -H "version:''" -k -X POST -d '{"transaction":{"type":2,"amount":0,"fee":10000000000,"recipientId":null,"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","timestamp":5334485,"asset":{"delegate":{"username":"zhenxi\_test","publicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f"}},"signature":"a12ce415d2d21ab46e4c1b918b8717b1d351dd99abd6f2f94d9a1a7e1f32b697f843a05b1851cb857ea45a2476dce592f5ddd612c00cd44488b8b610c57d7f0a","signSignature":"35adc9f1f37d14458e8588f9b4332eedf1151c02480159f64a287a4b0cbb59bfe82040dfec96a4d9560bae99b8eaa1799a7023395db5ddc640d95447992d6e00","id":"12310465407307249905"}}' http://45.32.248.33:4096/peer/transactions

JSON返回示例：

{

"success":true //注册受托人成功

}

#### ****2.9.2.4 投票 & 取消投票****

请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| transaction | json | Y | etm-js.vote.createVote生成的交易数据 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功 |

请求示例：

var etm = require('etm-js');

var password = 'measure bottom stock hospital calm hurdle come banner high edge foster cram';

var secondPassword = 'erjimimashezhi001';// 投票内容是一个列表，列表中的每一个元素是一个符号加上所选择的受托人的公钥，符号为+表示投票，符号为-表示取消投票

var voteContent = [

'-ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7',

'+c292db6ea14d518bc29e37cb227ff260be21e2e164ca575028835a1f499e4fe2'

];

var transaction = etm.vote.createVote(password, voteContent, secondPassword || undefined);

JSON.stringify(transaction)

{"type":3,"amount":0,"fee":10000000,"recipientId":null,"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","timestamp":5334923,"asset":{"vote":{"votes":["-ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7","+c292db6ea14d518bc29e37cb227ff260be21e2e164ca575028835a1f499e4fe2"]}},"signature":"6036c2066a231c452a1c83aafd3bb9db3842ee05d5f17813f8264a4294cdec761faa89edf4a95f9b2e2451285807ab18aa9f989ad9a3165b95643179b8e4580f","signSignature":"a216ca739112e6f65986604b9467ccc8058138a7077faf134d6c4d673306cd1c514cc95bd54a036f7c602a56c4b4f2e4e59f6aa7c376cb1429e89054042e050b","id":"17558357483072606427"}

// 将上面生成的投票的交易数据通过post提交给etm server

curl -H "Content-Type: application/json" -H "magic:594fe0f3" -H "version:''" -k -X POST -d '{"transaction":{"type":3,"amount":0,"fee":10000000,"recipientId":null,"senderPublicKey":"3e6e7c90571b9f7dabc0abc2e499c2fcee8e436af3a9d5c8eadd82ac7aeae85f","timestamp":5334923,"asset":{"vote":{"votes":["-ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7","+c292db6ea14d518bc29e37cb227ff260be21e2e164ca575028835a1f499e4fe2"]}},"signature":"6036c2066a231c452a1c83aafd3bb9db3842ee05d5f17813f8264a4294cdec761faa89edf4a95f9b2e2451285807ab18aa9f989ad9a3165b95643179b8e4580f","signSignature":"a216ca739112e6f65986604b9467ccc8058138a7077faf134d6c4d673306cd1c514cc95bd54a036f7c602a56c4b4f2e4e59f6aa7c376cb1429e89054042e050b","id":"17558357483072606427"}}' http://45.32.248.33:4096/peer/transactions

JSON返回示例：

{

"success":true //投票&取消投票 成功

}

# ****2.10 矿机****

### ****2.10.1 获取矿工列表****

接口地址：/api/miner   
请求方式：get  
支持格式：urlencoded  
接口说明：如果不加参数则会返回全网注册矿机列表  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| address | string | N | 矿工地址 |
| limit | int | N | 限制返回结果数据集的个数 |
| offset | integer | N | 步长，最小值：0 |
| orderBy | string | N | 排序字段:排序规则，如:desc |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| miners | Array | 矿工详情列表 |

请求示例：

curl -k -X GET 'http://127.0.0.1:4096/api/miner?limit=20' //取出前20名

JSON返回示例：

{

"success": true,

"miners": [{

"minerNo": "no10000000", //矿机编号

"minerName": "超级矿机", //矿机名称

"address": "14636456069025293113", //矿机拥有者地址

"publicKey": "ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7", //矿机拥有者公钥

"vote": 15600500, //得票数

"producedblocks": 1371, //生成的区块数

"missedblocks": 6, //丢失的区块数

"fees": 12588514990, //手续费

"rewards": 276850000000, //已经得到的奖励

"rate": 1,

"approval": 98.54, //得票率

"productivity": 99.56, //生产率

"forged": "289438514990", //挖矿产生的所有奖励

"cheat": 0, //作弊次数

"averagepower":”400.13 MH/s”, //平均算力

"alreadyruntime": 234450, //已经运行时间

"certified": Y //是否认证

},

{

"minerNo": "no10000001", //矿机编号

"minerName": "罗马矿池",

"address": "9961157415582672274",

"publicKey": "c292db6ea14d518bc29e37cb227ff260be21e2e164ca575028835a1f499e4fe2",

"vote": 9891994035600500,

"producedblocks": 1370,

"missedblocks": 8,

"fees": 12355148480,

"rewards": 275100000000,

"rate": 2,

"approval": 98.44,

"productivity": 99.42,

"forged": "287455148480" ,

"cheat": 2, //作弊次数

"averagepower":”400.13 MH/s”, //平均算力

"alreadyruntime": 100000, //已经运行时间(min)

"certified": N //是否认证

}],

"totalCount": 233

}

### ****2.10.2 投选矿工****

接口地址：/api/miner   
请求方式：put  
支持格式：json  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |
| secondSecret | string | N | 账户二级密码，最小长度：1，最大长度：100 |
| miners | Array | 矿机编号数组，投票数 |  |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| transaction | json | 投票交易详情 |

请求示例：

curl -k -H "Content-Type: application/json" -X PUT -d '{"secret":"call scissors pupil water friend timber spend brand vote obey corn size","publicKey":"3ec1c9ec08c0512641deba37c0e95a0fe5fc3bdf58424009f594d7d6a4e28a2a","miners":[{minerNo:”9961157415582672274”,vote:1000}，{minerNo:”9961157415582672275”,vote:1000}]}' 'http://127.0.0.1:4096/api/miners/vote'

JSON返回示例：

{ "success": true }

### ****2.10.3 获取矿机属性****

接口地址：/api/miner/prop  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| minerNo | string | Y | 矿机唯一编号,最小长度：1 |
| props | array | n | 属性列表，目前支持basic, doubledig, register,power |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| miner | json | 矿工信息 |

请求示例：

curl -k -X GET http://127.0.0.1:4096/api/miner/view?minerNo=no167234734

JSON返回示例：

{

"success": true,

“data”: {

"basic": {

"minerNo": "no10000000", //矿机编号

"minerName": "超级矿机", //矿机名称

"address": "14636456069025293113", //矿机拥有者地址

"publicKey": "ae256559d06409435c04bd62628b3e7ea3894c43298556f52b1cfb01fb3e3dc7", //矿机拥有者公钥

"vote": 15600500, //得票数

"producedblocks": 1371, //生成的区块数

"missedblocks": 6, //丢失的区块数

"fees": 12588514990, //手续费

"rewards": 276850000000, //已经得到的奖励

"rate": 1,

"approval": 98.54, //得票率

"productivity": 99.56, //生产率

"forged": "289438514990", //挖矿产生的所有奖励

"cheat": 0, //作弊次数

"averagepower":”400.13 MH/s”, //平均算力

"alreadyruntime": 234450, //已经运行时间

"certified": Y //是否认证

},

"doubledig":{ //双挖信息

“enable” : ”Y”, //开启双挖

“currency” : ”ETH”, //eth:以太坊 btc:比特币

}，

“register” : [{title:”升级近况”,msg:”最新更新时间2018.8.20，预计矿机各升级”},

{title:”认证信息”,msg:”本矿机是Inter专属定制”},

{title:”GPS地址”,msg:”广东省深圳市南山区文心五路2308号”}],

“power”: [

{time: 11, power: 1928844},

{time: 12, power: 19384757},

{time: 13, power: 293884}

]

}

}

### ****2.10.4 获取矿工SCV列表****

接口地址：/api/miner/scv  
请求方式：get  
支持格式：urlencoded  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| status | string | Y | RUN(正式);WAIT(候补);EMPTY(空置) |
| orderBy | string | N | 排序字段:排序规则 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| miners | Array | 矿工详情列表 |

请求示例：

curl -k -X GET 'http://127.0.0.1:4096/api/miner/scv?status=RUN

JSON返回示例：

{

"success": true,

"miners": [{

"minerNo": "no10000000", //矿机编号

"minerName": "超级矿机", //矿机名称

"address": "14636456069025293113", //矿机拥有者地址

"vote": 15600500, //得票数

},

{

"minerNo": "no10000001", //矿机编号

"minerName": "罗马矿池",

"address": "9961157415582672274",

"vote": 9891994035600500,

}],

"totalCount": 2

}

### ****2.10.6 获取选取矿工记录****

接口地址：/api/miner/vote  
请求方式：get  
支持格式：urlencoded  
接口说明：如果不加参数则会返回全网注册矿机列表  
请求参数说明：

| **名称** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| secret | string | Y | etm账户密码 |
| publicKey | string | N | 公钥 |
| secondSecret | string | N | 账户二级密码，最小长度：1，最大长度：100 |

返回参数说明：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| success | boole | 是否成功获得response数据 |
| miners | Array | 矿工详情列表 |

请求示例：

curl -k -X GET 'http://127.0.0.1:4096/api/miner/vote/my?secret=”unaware label emerge fancy concert long fiction report affair appear decide twenty

”'

JSON返回示例：

{

"success": true,

"miners": [{

"minerNo": "no10000000", //矿机编号

"minerName": "超级矿机", //矿机名称

"vote": 15600500, //得票数

"producedblocks": 1371, //生成的区块数

"missedblocks": 6, //丢失的区块数

"approval": 98.54, //得票率

"productivity": 99.56, //生产率

"forged": "289438514990", //挖矿产生的所有奖励

"cheat": 0, //作弊次数

"averagepower":”400.13 MH/s”, //平均算力

"alreadyruntime": 234450, //已经运行时间

"certified": Y //是否认证

"status": "RUN" //矿机状态

},

{

"minerNo": "no10000001", //矿机编号

"minerName": "罗马矿池",

"vote": 9891994035600500,

"producedblocks": 1370,

"missedblocks": 8,

"approval": 98.44,

"productivity": 99.42,

"forged": "287455148480" ,

"cheat": 2, //作弊次数

"averagepower":”400.13 MH/s”, //平均算力

"alreadyruntime": 100000, //已经运行时间(min)

"certified": N //是否认证

"status": "WAIT" //矿机状态

}],

"totalCount": 2

}