

---

# **sphinxcontrib-httpdomain Documentation**

*Release 0.10.0*

**Hong Minhee**

**Jun 03, 2020**



---

## Contents

---

<b>1</b>	<b>useful links</b>	<b>3</b>
1.1	General . . . . .	3
<b>2</b>	<b>Contents</b>	<b>5</b>
2.1	Set up client . . . . .	5
2.2	Getting started with the nodejs API . . . . .	5
2.3	Changelog . . . . .	32
	<b>HTTP Routing Table</b>	<b>35</b>





Nodela provide access to elastos network. Our easy to use API and developer tools provide secure, reliable, and scalable access to elastos network.

---

**Note:** you can also construct your own network using [Node](#)

---

**Warning:** Since software is written by humans, it can have bugs. Thus, if you have questions you are always welcome to issue your problem.



---

useful links

---

## 1.1 General

- [Elastos](#)
- [Changelog](#)
- [Source Code](#)





Keyword Index, Search Page

## 2.1 Set up client

### 2.1.1 Walk Through

The code is written in java . make sure you add jdk in your environment.

1. download the code
  - git clone <https://github.com/elastos/Elastos.ORG.Wallet.Service.git>
2. modify application.properties
  - *node.prefix* change it to your own local mainchain node .
  - *did.mainChainAddress* change it to your own sidechain address in the mainchain if you want to do crosschain transfer.
3. install project
  - mvn install -Dmaven.test.skip -Dgpg.skip
4. start your project
  - java -jar base.api-\*.jar

## 2.2 Getting started with the nodela API

### 2.2.1 Introduction

Nodela has a Restful API with URL endpoints corresponding to actions that users can perform with their channels. The endpoints accept and return JSON encoded objects. The API URL path always contains the API version in order

to differentiate queries to different API versions. All queries start with: `/api/<version>/` where `<version>` is an integer representing the current API version.

## 2.2.2 Elastos Mainchain Service API

using the following api ,we can easily get a glimps of what is going on in the blockchain.

### get transaction by transaction id

check out a transaction

**GET** `/api/1/tx/(string:`txid`)`

Example request:

```
GET /api/1/tx/62637968e72b06e4fa1de91542a3b71bd2462ba1d29e9c14c2ecfd042d1937ab_
↪ HTTP/1.1
Host: localhost
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": {
    "vsize": 346,
    "locktime": 0,
    "txid": "62637968e72b06e4fa1de91542a3b71bd2462ba1d29e9c14c2ecfd042d1937ab",
    "confirmations": 6756,
    "type": 8,
    "version": 0,
    "vout": [
      {
        "outputlock": 0,
        "address": "XQd1DCi6H62NQdWZQhJCRnrPn7sF9CTjaU",
        "assetid":
↪ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
        "value": "0.10010000",
        "n": 0
      },
      {
        "outputlock": 0,
        "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
        "assetid":
↪ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
        "value": "0.50249300",
        "n": 1
      }
    ],
    "blockhash":
↪ "4021e5c0ace86221016d3aa2b114adbd84bb03692bb6ddc6034794260834c570",
    "size": 346,
    "blocktime": 1538279155,
    "payload": {
      "CrossChainAddresses": [
        "EHLhCEbwViWBPwh1VhpECzYEA7jQHZ4zLv"
      ]
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    ],
    "OutputIndexes": [
        0
    ],
    "CrossChainAmounts": [
        10000000
    ]
},
"vin": [
    {
        "sequence": 0,
        "txid":
↪ "ba7bd41aae0a1371d9689ad04508f0754bb4a5333386411bccbdec718ce61625",
        "vout": 1
    }
],
"payloadversion": 0,
"attributes": [
    {
        "data": "32323432343239353130383035363838303230",
        "usage": 0
    }
],
"time": 1538279155,
"programs": [
    {
        "code":
↪ "21021421976fdb518ca4e8b91a37f1831ee31e7b4ba62a32dfe2f6562efd57806adac",
        "parameter":
↪ "40cf6b8a18c861fcad1c23816221cc40a0d2e7d43065c070e66905ff7d6c634068542dd2a9b0bbb24de6a5a547b57"
↪ ""
    }
],
"hash": "62637968e72b06e4fa1de91542a3b71bd2462ba1d29e9c14c2ecfd042d1937ab"
},
"status": 200
}

```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

### get transactions

get a list of transactions

**POST** /api/1/tx

Example request:

```
POST /api/1/tx HTTP/1.1
Host: localhost:8090
Content-Type: application/json

[
  "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c8",
  "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c9",
  "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c8"
]
```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": [
    {
      "vsize": 288,
      "locktime": 0,
      "txid":
↪ "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c8",
      "confirmations": 13,
      "type": 2,
      "version": 0,
      "vout": [
        {
          "outputlock": 0,
          "address": "8NJ7dbKsG2NRiBqdhY6LyKMiWp166cFBiG",
          "assetid":
↪ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
          "value": "1",
          "n": 0
        },
        {
          "outputlock": 0,
          "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
          "assetid":
↪ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
          "value": "977.89999500",
          "n": 1
        }
      ],
      "blockhash":
↪ "75d78222e8f8b7622ab45902fd7a79c03edf08bceb1078335e9a8caf90cee612",
      "size": 288,
      "blocktime": 1539919032,
      "vin": [
        {
          "sequence": 0,
          "txid":
↪ "f176d04e5980828770acadcf3e2d471885ab7358cd7d03f4f61a9cd0c593d54",
          "vout": 1
        }
      ],
      "payloadversion": 0,
```

(continues on next page)

(continued from previous page)

```

        "attributes": [
            {
                "data": "e6b58be8af95",
                "usage": 129
            }
        ],
        "time": 1539919032,
        "programs": [
            {
                "code":
→ "21021421976fdb518ca4e8b91a37f1831ee31e7b4ba62a32dfe2f6562efd57806adac",
                "parameter":
→ "403792fa7dd7f29a810ab247e6476ca814ae51c550419f101948db6141004b364b645d84aaecdcb96790bd8cd7606
→ "
            }
        ],
        "hash":
→ "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c8"
    },
    "Unknown Transaction",
    {
        "vsize": 288,
        "locktime": 0,
        "txid":
→ "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c8",
        "confirmations": 13,
        "type": 2,
        "version": 0,
        "vout": [
            {
                "outputlock": 0,
                "address": "8NJ7dbKsG2NRiBqdhY6LyKMiWp166cFBiG",
                "assetid":
→ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
                "value": "1",
                "n": 0
            },
            {
                "outputlock": 0,
                "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
                "assetid":
→ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
                "value": "977.89999500",
                "n": 1
            }
        ],
        "blockhash":
→ "75d78222e8f8b7622ab45902fd7a79c03edf08bceb1078335e9a8caf90cee612",
        "size": 288,
        "blocktime": 1539919032,
        "vin": [
            {
                "sequence": 0,
                "txid":
→ "f176d04e5980828770acadcf3e2d471885ab7358cd7d03f4f61a9cd0c593d54",
                "vout": 1
            }
        ]
    }

```

(continues on next page)

(continued from previous page)

```

    ],
    "payloadversion": 0,
    "attributes": [
        {
            "data": "e6b58be8af95",
            "usage": 129
        }
    ],
    "time": 1539919032,
    "programs": [
        {
            "code":
↪ "21021421976fdb518ca4e8b91a37f1831ee31e7b4ba62a32dfe2f6562efd57806adac",
            "parameter":
↪ "403792fa7dd7f29a810ab247e6476ca814ae51c550419f101948db6141004b364b645d84aaecdcb96790bd8cd7606
↪ "
        }
    ],
    "hash":
↪ "64955791d225fddae4bba01547712c53f97ce3fb38252c01dbb9d6d9b7b982c8"
    }
  ],
  "status": 200
}

```

## get utxos of specific address

check out a transaction

**GET** /api/1/utxos/(string:`address`)

**Example request:**

```

GET /api/1/utxos/EHLhCEbwViWBPwhlVhpECzYEA7jQHZ4zLv HTTP/1.1
Host: localhost

```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": [
    {
      "AssetId":
↪ "a3d0eaa466df74983b5d7c543de6904f4c9418ead5ffd6d25814234a96db37b0",
      "AssetName": "ELA",
      "Utxo": [
        {
          "Txid":
↪ "4213d73e3764750cbab09593db213ac1ea222947d91e6e6d2c243d2461bd034c",
          "Value": "0.00010000",
          "Index": 0
        }
      ]
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    },
    "status": 200
  }

```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

## Check the current network block height

tells you the current block height of the network

**GET** /api/1/currHeight

Example request:

```

GET /api/1/currHeight HTTP/1.1
Host: localhost

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": 128797,
  "status": 200
}

```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

## get the balance of address

get the balance of the provided public address

**GET** /api/1/balance/(string: `public\_address`)

Example request:

```

GET /api/1/balance/EbunxcqXie6UExs5SXDbFZxr788iGGvAs9 HTTP/1.1
Host: localhost

```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": "2.11059400",
  "status": 200
}
```

**Status Codes**

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

**Get the transactions of specific height**

using height to get block contained transactions

**GET /api/1/txs/(int:`block\_height`)**  
get the transactions that the user (*block\_height*) wrote.

**Example request:**

```
GET /api/1/txs/10 HTTP/1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": {
    "Transactions": [
      "53b06e08da9362abf50003e26f8b99b38bd32b6a7dfad83203ef5bb9da2f4a05"
    ],
    "Height": 10,
    "Hash": "1166ae059fd6914a44edde9aa8a2765138da0ab868ddaeb51d20d21908c488da"
  },
  "status": 200
}
```

**create offline transaction**

create a offline transaction utxo json data , you should sign it using private key

**POST /api/1/createTx**

**Example request:**



```
POST /api/1/createTx HTTP/1.1
Host: localhost

{
  "inputs" : ["EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV"],
  "outputs" : [{
    "addr": "EPzxJrHefvE7TCWmEGQ4rcFgxGeGBZFSHw",
    "amt" : 1000
  }]
}
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": {
    "Transactions": [
      {
        "UTXOInputs": [
          {
            "address": "EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV",
            "txid":
→ "fa9bcb8b2f3a3a1e627284ad8425faf70fa64146b88a3aceac538af8bfef9d1",
            "index": 1
          }
        ],
        "Fee": 100,
        "Outputs": [
          {
            "amount": 1000,
            "address": "EPzxJrHefvE7TCWmEGQ4rcFgxGeGBZFSHw"
          },
          {
            "amount": 99997800,
            "address": "EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV"
          }
        ]
      }
    ],
    "status": 200
  }
}
```

**create offline vote transaction**

create a offline transaction utxo json data , you should sign it using private key

**POST /api/1/createVoteTx**

**Example request:**

```
POST /api/1/createVoteTx HTTP/1.1
Host: localhost

{
```

(continues on next page)

(continued from previous page)

```

    "inputs": [
        "ELbKQrj8DTYn2gU7KBejcNWb4ix4EAGDmy"
    ],
    "outputs": [
        {
            "addr": "EQNJEA8XhraX8a6SBq98ENU5QSW6nvgSHJ",
            "amt": 1300
        }
    ]
}

```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
    "result": {
        "Transactions": [
            {
                "UTXOInputs": [
                    {
                        "address": "ELbKQrj8DTYn2gU7KBejcNWb4ix4EAGDmy",
                        "txid":
→ "498b1fd7c9e455163b79a46b68ac272a7ec4736995c82deab975848221031981",
                        "index": 0
                    },
                    {
                        "address": "ELbKQrj8DTYn2gU7KBejcNWb4ix4EAGDmy",
                        "txid":
→ "498b1fd7c9e455163b79a46b68ac272a7ec4736995c82deab975848221031981",
                        "index": 1
                    }
                ],
                "Fee": 4860,
                "Outputs": [
                    {
                        "amount": 1300,
                        "address": "EQNJEA8XhraX8a6SBq98ENU5QSW6nvgSHJ"
                    },
                    {
                        "amount": 688546,
                        "address": "ELbKQrj8DTYn2gU7KBejcNWb4ix4EAGDmy"
                    }
                ]
            }
        ],
        "status": 200
    }
}

```

**create offline cross chain transaction**

create a cross chain offline transaction utxo json data , you should sign it using private key

**POST /api/1/createCrossTx**

**Example request:**

```
POST /api/1/createCrossTx HTTP/1.1
Host: localhost

{
  "inputs": [
    "EZEKtpUwDXyKLvq5JPRfezqA5k5G5rWa3Z",
    "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
  ],
  "outputs": [
    {
      "addr": "ELag7vYvKcUBVKJkWosBQw73HSx8madjcP",
      "amt": 120001
    }
  ]
}
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": {
    "Transactions": [
      {
        "UTXOInputs": [
          {
            "address": "EZEKtpUwDXyKLvq5JPRfezqA5k5G5rWa3Z",
            "txid":
            ↪ "a7f06797be1d354d27173045c63426d20f9085c5a175ea5fcc38a771c7148406",
            "index": 1
          },
          {
            "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "txid":
            ↪ "9e17bb1448f5b5805c682afff28226193884e84a72d8680a35ce0fcf2ec70d97",
            "index": 1
          }
        ],
        "CrossChainAsset": [
          {
            "amount": 120001,
            "address": "ELag7vYvKcUBVKJkWosBQw73HSx8madjcP"
          }
        ],
        "Fee": 20000,
        "Outputs": [
          {
            "amount": 130001,
            "address": "XKUh4GLhFJiqAMTF6HyWQrV9pK9HcGUdfJ"
          },
          {
            "amount": 34198299,
            "address": "EZEKtpUwDXyKLvq5JPRfezqA5k5G5rWa3Z"
          }
        ]
      }
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```
        }
    ],
    "status": 200
}
```

## send offline transaction

send raw transaction

**POST /api/1/sendRawTx**

**Example request:**

```
POST /api/1/sendRawTx HTTP/1.1
Host: localhost

{
  "data":
  ↳ "0200010013313637333832373132343538363832353937350191FDEFBFF88A53ACCE3A8AB84641A60FF7FA2584AD8
  ↳ "
}
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": "1f4432635bcf8c347f2bc20b7906c8c6c195f51beb3426e5f8d6a9e4cc073cf3",
  "status": 200
}
```

## get address transaction history

check history of Address

---

**Note:** this api provided by [Elastos.ORG.API.Misc](#), so you need to deploy it if you want to use it.

---

**GET /api/1/history/(string:`addr`)**

**Example request:**

```
GET /api/1/history/EQoascGFzdQ1rLKfNEavJKTm3hMRhBaXBT HTTP/1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": {
    "History": [
```

(continues on next page)

(continued from previous page)

```

        {
            "Txid":
↪ "71eedccf5a45acc17036294b7a1e70c5756dd4c72032cc76d5089c522509096b",
            "Type": "income",
            "Value": 24,
            "CreateTime": 1544101570,
            "Height": 175548,
            "Fee": 100,
            "Inputs": [
                "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
            ],
            "Outputs": [
                "EQoascGFzdQ1rLKfNEavJKTm3hMRhBaXBT",
                "EQoascGFzdQ1rLKfNEavJKTm3hMRhBaXBT",
                "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
            ]
        },
        {
            "TotalNum": 1
        },
        {
            "status": 200
        }
    ]
}

```

**GET** /api/1/history/(string:`addr`)?pageSize=(int:`pageSize`)&pageNum=(int:`pageNum`)

Example request:

```

GET /api/1/history/EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA?pageSize=4&pageNum=1 HTTP/1.
↪ 1
Host: localhost

```

Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
    "result": {
        "History": [
            {
                "Txid":
↪ "58e11ffffafad7456835dc3eff8b28ec0d0b2aa39855d79bfb3e5c52b4cc308f",
                "Type": "income",
                "Value": 8990000,
                "CreateTime": 1538114443,
                "Height": 38666,
                "Fee": 10000,
                "Inputs": [
                    "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
                ],
                "Outputs": [
                    "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
                ]
            },
            {
                "Txid":
↪ "b04714ff7076e18efeb9963f99af2f10bd870d39a1774e0821b71640d6b14e52",

```

(continues on next page)

(continued from previous page)

```

        "Type": "spend",
        "Value": 100,
        "CreateTime": 1538115578,
        "Height": 38667,
        "Fee": 100,
        "Inputs": [
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ],
        "Outputs": [
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ]
    },
    {
        "Txid":
        ↪ "49eb7f1335d89a9c43446613850d1b763a5f84cad0d1d039ab5726125d2eab47",
        "Type": "spend",
        "Value": 100,
        "CreateTime": 1538203592,
        "Height": 39046,
        "Fee": 100,
        "Inputs": [
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ],
        "Outputs": [
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ]
    },
    {
        "Txid":
        ↪ "d7d8b34944790cf2e1fa47851612ddb4eb7e3f5bd75a4af2227d3786b6f4c122",
        "Type": "spend",
        "Value": 100,
        "CreateTime": 1539238227,
        "Height": 44755,
        "Fee": 100,
        "Inputs": [
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ],
        "Outputs": [
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ]
    }
],
    "TotalNum": 55295
},
    "status": 200
}

```

## get dpos producer vote statistics

producer's vote statistics

**Note:** this api provided by [Elastos.ORG.API.Misc](#), so you need to deploy it if you want to use it.

**GET** /api/1/dpos/producer/ (string: `producer\_public\_key`)

**Example request:**

```
GET /api/1/dpos/producer/
↪ 03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800 HTTP/1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": [
    {
      "Producer_public_key":
↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
      "Vote_type": "Delegate",
      "Txid":
↪ "2638f858000dd118015daa7b1ee23c86e1c0738b5e641265d52f6612c527c672",
      "N": 0,
      "Value": "4999",
      "Outputlock": 0,
      "Address": "EbeD11dua88L9VQtNmJuEez8aVYX294CML",
      "Block_time": 1551800055,
      "Height": 233745
    },
    {
      "Producer_public_key":
↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
      "Vote_type": "Delegate",
      "Txid":
↪ "82fce02fb0e835102eb37633e513e78c825a534d46146962391866e25bf8005c",
      "N": 0,
      "Value": "9999",
      "Outputlock": 0,
      "Address": "EKmp4dqTSMVW2f2H3x5H2A6vQf7FJV8Frj",
      "Block_time": 1551838308,
      "Height": 234056
    },
    {
      "Producer_public_key":
↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
      "Vote_type": "Delegate",
      "Txid":
↪ "74f2beb77f15fcc6f36e43533aec254fc17b84edbb7e2b3a625c9ac2867a7435",
      "N": 0,
      "Value": "123",
      "Outputlock": 0,
      "Address": "EWHEoukFBK6AyMjuS9ucxhQ2twS7BKQEv8",
      "Block_time": 1551838618,
      "Height": 234058
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

        "Producer_public_key":
        ↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
        "Vote_type": "Delegate",
        "Txid":
        ↪ "1a71b89c5e6c1b9baf31884f075f5e3ea159d8edfe5d665a2f5182d0c715ff91",
        "N": 0,
        "Value": "9999",
        "Outputlock": 0,
        "Address": "EYZt2Xk76NNFEHiihqkyBhyzuwlabcheXF",
        "Block_time": 1551850832,
        "Height": 234161
    },
    {
        "Producer_public_key":
        ↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
        "Vote_type": "Delegate",
        "Txid":
        ↪ "71083736e824c73e4b327a8b958dbbd00aec879768a96963cbdfc5008e1bd393",
        "N": 0,
        "Value": "0.01111111",
        "Outputlock": 0,
        "Address": "ELbKQrj8DTYn2gU7KBejcNWb4ix4EAGDmy",
        "Block_time": 1551851053,
        "Height": 234163
    },
    {
        "Producer_public_key":
        ↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
        "Vote_type": "Delegate",
        "Txid":
        ↪ "fbc81da6db6db5cb09c76fe405cf238353a8e837dda5acacd137ba43a9da1d02",
        "N": 0,
        "Value": "9999",
        "Outputlock": 0,
        "Address": "ENaaqePNBtrZsNbs9uc35CPqTbvn8oaYL9",
        "Block_time": 1551853616,
        "Height": 234180
    },
    {
        "Producer_public_key":
        ↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
        "Vote_type": "Delegate",
        "Txid":
        ↪ "82529a764fd1bbdd4ae39e9bb791d029ecb3010b7db48a7b5d1edfe8be71f36e",
        "N": 0,
        "Value": "9999",
        "Outputlock": 0,
        "Address": "Ea3XHVqFiAjYA4sSCTQSmrWQafGkboxaYxe",
        "Block_time": 1551853616,
        "Height": 234180
    }
],
"status": 200
}

```



## get dpos voter's statistics

voter's statistics

**Note:** this api provided by [Elastos.ORG.API.Misc](#), so you need to deploy it if you want to use it.

**GET** /api/1/dpos/address/(string:`address`)

**Example request:**

```
GET /api/1/dpos/address/ENaaqePNBtrZsNbs9uc35CPqTbvn8oaYL9 HTTP/1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": [
    {
      "Producer_public_key":
        ↪ "03330ee8520088b7f578a9afabaef0c034fa31fe1354cb3a14410894f974132800",
      "Vote_type": "Delegate",
      "Txid":
        ↪ "fbc81da6db6db5cb09c76fe405cf238353a8e837dda5acacd137ba43a9da1d02",
      "N": 0,
      "Value": "9999",
      "Outputlock": 0,
      "Address": "ENaaqePNBtrZsNbs9uc35CPqTbvn8oaYL9",
      "Block_time": 1551853616,
      "Height": 234180
    }
  ],
  "status": 200
}
```

## create a ELA wallet

generate a elastos wallet

**GET** /api/1/createWallet

**Example request:**

```
GET /api/1/createWallet HTTP/1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": {
    "privateKey":
      ↪ "492F67D441F563AA4746497EB77C89906A3D3C06B242030BA966BC5604482EF7",

```

(continues on next page)

(continued from previous page)

```
    "publicKey":  
    ↪ "035EBC0D70C9E34006C932D7BB47474159C136A8944C92416A94481212379751CB",  
    "address": "EJonBz8U1gYnANjSafRF9EAJW9KTwrKd6x"  
  },  
  "status": 200  
}
```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

## generate mnemonic phrases

please copy your mnemonic to somewhere safe

**GET /api/1/eng/mnemonic**

Example request:

```
GET /api/1/eng/mnemonic HTTP/1.1  
Host: localhost
```

Example response:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
  
{  
  "result": "obtain pill nest sample caution stone candy habit silk husband give_  
  ↪net",  
  "status": 200  
}
```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

**GET /api/1/cn/mnemonic**

Example request:

```
GET /api/1/cn/mnemonic HTTP/1.1  
Host: localhost
```

Example response:

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": "          ",
  "status": 200
}

```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

## using mnemonic to retrieve wallet

Get wallet of index 1

POST /api/1/hd

Example request:

```

POST /api/1/hd HTTP/1.1
Host: localhost:8090
Content-Type: application/json

{
  "mnemonic": "obtain pill nest sample caution stone candy habit silk husband_
↪give net",
  "index": 1
}

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": {
    "privateKey":
↪"7A87D1C43FBDF76689A5A66A369B34E92391748F64D2952BCE3E6D5E06A8D8CD",
    "publicKey":
↪"02345363ACEA3A744DC149193171A87B6888F4CD108821CC1F9AD689CCA53489AC",
    "publicAddress": "EXoaGjh6H9afjDX7DUBY1MpsdLz4Vo16Qa"
  },
  "status": 200
}

```

Get wallet from 1 to 10

POST /api/1/hd

Example request:

```
POST /api/1/hd HTTP/1.1
Host: localhost:8090
Content-Type: application/json

{
  "mnemonic": "obtain pill nest sample caution stone candy habit silk husband_
↪give net",
  "start": 1,
  "end": 10
}
```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": [
    {
      "privateKey":
↪"7A87D1C43FBDF76689A5A66A369B34E92391748F64D2952BCE3E6D5E06A8D8CD",
      "publicKey":
↪"02345363ACEA3A744DC149193171A87B6888F4CD108821CC1F9AD689CCA53489AC",
      "publicAddress": "EXoaGjh6H9afjDX7DUBY1MpsdLz4Vo16Qa"
    },
    {
      "privateKey":
↪"D0E8FB87B32EC69EC8527486AF6780DB06983F395F8394372B88F9F29F738A91",
      "publicKey":
↪"02371040B81C28B3A194826E1F8905687E06E94D5AEB1292C72051BF6ED314888D",
      "publicAddress": "EfPZc1VdGHztzeRag9ayb4MLPZA7psGuGs"
    },
    {
      "privateKey":
↪"5892859E60028872E63AEC018217F6A9A38AFC05F4FA4DB2FA2D07455B6C46E1",
      "publicKey":
↪"0371EAD623897AEA29CBFB2ABC9A361E5862DCAEF68A265C6D4F8A506D3353CFC8",
      "publicAddress": "EPc6Mmx78YN8MwallqjLgSagPjAWU7LJZu"
    },
    {
      "privateKey":
↪"9F8A564E7CC0E4880006B8029A1FA220A60AF426D2CD90DEB52F4CCD5E89087C",
      "publicKey":
↪"0323671B0FB55471D885300445E25893A1EF304484C73BD6F380959BF90987A0A9",
      "publicAddress": "EPgRBCe4BwNVpWFdNhLhTEebQG35b9MoFz"
    },
    {
      "privateKey":
↪"CDA64875FED6B901595C732BD71F925C6CACF5DBC503F7C472D950975A90EF05",
      "publicKey":
↪"027B4AEFD9208E25BD27F0416729E73187CB22B58859D40E756451D3B73079FD54",
      "publicAddress": "EKi96gzJSQj3zQ6LybXwNPmEguj66H8SjN"
    },
    {
      "privateKey":
↪"1F3AE27B2E070E8A95257295FE602113D81A30B7554C3F77378A1CD04AEBBE53",

```

(continues on next page)

(continued from previous page)

```

        "publicKey":
        ↪ "032063D96363FE153F85DCC0550C2753EEC407334DBACCDD933F3DC45CBC7BC1E0",
        "publicAddress": "EVVEw7p4wcfDXPMRzZhcd7TEGR1EXBG291"
    },
    {
        "privateKey":
        ↪ "9C6E82523939E11A455E2962EB0DE49211E7F4216D76836D557D9D395C59C4DB",
        "publicKey":
        ↪ "02622D033371BFF4DB1512F655A22B1D8783B02E703477E35B76974EA2D9C89B9",
        "publicAddress": "EbYdmcS99kCo8D44AphCBjtE4SmGLEGsWw"
    },
    {
        "privateKey":
        ↪ "8A69ADD6A8534C2E7D52067D8929D65354F8D218EE0FF1FA1C631A334821E85E",
        "publicKey":
        ↪ "024A4751A73AA5E186D83C171449FB8AF699C6EE0BC24DB34B85EEF5FA6EEDBD27",
        "publicAddress": "EcBVvLx31eN69KS3oWNLxxdaGK2UKUxg3g"
    },
    {
        "privateKey":
        ↪ "4E2C9E4C320AF08C943AADA5BC9106E053DFD2AFC8A3F229213DB6BE36DB71D8",
        "publicKey":
        ↪ "0354674BFA392313760DD66936A2BEC9CDD8DA7AB264D920FC4AF0B14704646BA5",
        "publicAddress": "EPbCbYYCwbyxFQgU9XNAWwAFRbcJtDC7Tt"
    },
    {
        "privateKey":
        ↪ "31E5DB750E2BAC324A0C7F59B2BAB9C574232B811F7A4650352AAD31A3C2A941",
        "publicKey":
        ↪ "036A2ED5B9F57636460B4CB38A6103E03A08F8A6E5B4F0326CD303B6407E7CC303",
        "publicAddress": "ESzmxv7R4Tkt1SakscXRlyxTYtPwuXtvZP"
    }
],
"status": 200
}

```

## Signing message using private key

### POST /api/1/sign

Example request:

```

POST /api/1/sign HTTP/1.1
Host: localhost:8090
Content-Type: application/json

{
    "privateKey":
    ↪ "0D5D7566CA36BC05CFF8E3287C43977DCBB492990EA1822643656D85B3CB0226",
    "msg": ""
}

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept

```

(continues on next page)

(continued from previous page)

```

Content-Type: application/json

{
  "result": {
    "msg": "E4BDA0E5A5BDEFBC8CE4B896E7958C",
    "pub": "02C3F59F337814C6715BBE684EC525B9A3CFCE55D9DEEC53E1EDDB0B352DBB4A54
    ↪",
    "sig":
    ↪"E6BB279CBD4727B41F2AA8B18E99B3F99DECBB8737D284FFDD408B356C912EE21AD478BCC0ABD65246938F17DDE64
    ↪"
  },
  "status": 200
}

```

### verify message signed by a public address's private key

#### POST /api/1/verify

Example request:

```

POST /api/1/verify HTTP/1.1
Host: localhost
Content-Type: application/json

{
  "msg": "E4BDA0E5A5BDEFBC8CE4B896E7958D",
  "pub": "02C3F59F337814C6715BBE684EC525B9A3CFCE55D9DEEC53E1EDDB0B352DBB4A54",
  "sig":
  ↪"E6BB279CBD4727B41F2AA8B18E99B3F99DECBB8737D284FFDD408B356C912EE21AD478BCC0ABD65246938F17DDE64
  ↪"
}

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": true,
  "status": 200
}

```

### transfer ELA using private key

using private key to send transaction

#### POST /api/1/transfer

Example request:

```

POST /api/1/transfer HTTP/1.1
Host: localhost

{
  "sender": [

```

(continues on next page)

(continued from previous page)

```

        {
            "address": "EHLhCEbwViWBPwh1VhpECzYEA7jQHZ4zLv",
            "privateKey":
↪ "C740869D015E674362B1F441E3EDBE1CBCF4FE8B709AA1A77E5CCA2C92BAF99D"
        },
        {
            "address": "EbunxcqXie6UExs5SXDbFZxr788iGGvAs9",
            "privateKey":
↪ "FABB669B7D2FF2BEBBED1C3F1C9A9519C48993D1FC9D89DCB4C7CA14BDB8C99F"
        }
    ],
    "memo": "",
    "receiver": [
        {
            "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "amount": "2.4"
        }
    ]
}

```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
    "result": "7bcb5fbf7d6e8f673d50999709d695030dbd6d60a00281221540498cf2830f59",
    "status": 200
}

```

**Dpos Voting using private key**

using private key to vote your producers

**Note:** the receiver address must be one of the senders address, for now the cr votes are evenly divided to the cr candidates. will be modified to each cr candidates can have their own specific votes .

**POST /api/1/dpos/vote****Example request:**

```

POST /api/1/dpos/vote HTTP/1.1
Host: localhost

{
    "sender": [
        {
            "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "privateKey":
↪ "C740869D015E674362B1F441E3EDBE1CBCF4FE8B709AA1A77E5CCA2C92BAF99D"
        }
    ],
    "receiver": [
        {

```

(continues on next page)

(continued from previous page)

```

        "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
        "amount": "0.00001234",
        "candidatePublicKeys": [
            ↪ "02fe6d1f9e1e03ef7e41bf0cb65eea4f7c6e8f5faa902455b723b997a7e6ded40b",
            "028b6e4d5c85ce549cefb8987f61f3e50ae385737eadbefc3f90f70d6d1d530472",
            "030f74c89df33e19d0d657eead78ff1fe39f4e83ba56837d2e40b12a5e61cda8f5"
        ], "candidatesCrcDids": [
            "ih1kcYd76SvHPsUrBM72xkAk5Htkhz7xv8",
            "iXuss6uqEdaSB6aho9SpnLouu3EEaiBfsE"
        ]
    }
}

```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
    "result": "abd0943a279723f0feefd14535161a1d955c139c5791ae4e1a677532b7d3b212",
    "status": 200
}

```

**Mainchain to Sidechain transfer**

using this api you can transfer money from mainchain to did sidechain.

**POST /api/1/cross/m2d/transfer**

**Example Request:**

```

POST /api/1/cross/m2d/transfer HTTP/1.1
Host: localhost

{
    "sender": [
        {
            "address": "EHLhCEbwViWBPwh1VhpECzYEA7jQHZ4zLv",
            "privateKey":
            ↪ "0D5D7566CA36BC05CFF8E3287C43977DCBB492990EA1822643656D85B3CB0226"
        },
        {
            "address": "EbunxcqXie6UExs5SXDbFZxr788iGGvAs9",
            "privateKey":
            ↪ "FABB669B7D2FF2BEBBED1C3F1C9A9519C48993D1FC9D89DCB4C7CA14BDB8C99F"
        }
    ],
    "receiver": [
        {
            "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
            "amount": "0.01"
        }
    ]
}

```



**Example Response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": "B125D3DE38E6F3D17F9DC565996FDB00282BDD46A20F3B25C8EEDA99FC56EABB",
  "status": 200
}
```

## 2.2.3 Elastos FriendChain Service API

First add friend chain configuration in the configuration file:

```
friendchain.config.${friendChain_Symbol}_prefix     =${friendChain_Node_Restful_URL}
friendchain.config.${friendChain_Symbol}_assetId    =${friendChain_Asset_Id}
friendchain.config.${friendChain_Symbol}_fee        =${friendChain_Tx_Fee}
friendchain.config.${friendChain_Symbol}_satoshis   =${friendChain_Minimum_Unit}
```

**Example:**

```
friendchain.config.ioex_prefix      =http://54.92.80.93:20334
friendchain.config.ioex_assetId     =
↪=61ccbf9f8ce9660a71321041917139cb72cbb85bd105e92f0ed32cb1d1298f
friendchain.config.ioex_fee         =0.000001
friendchain.config.ioex_satoshis    =100000000
```

### Check the current network block height

tells you the current block height of the network

**GET** /api/1/\${friendChain\_Symbol}/currHeight

**Example request:**

```
GET /api/1/${friendChain_Symbol}/currHeight HTTP/1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": 128797,
  "status": 200
}
```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error

- **10001** – process error

### get the balance of address

get the balance of the provided public address

**GET** /api/1/\${friendChain\_Symbol}/balance/(string:`public\_address`)

**Example request:**

```
GET /api/1/${friendChain_Symbol}/balance/EbunxcqXie6UExs5SXDbFZxr788iGGvAs9 HTTP/
↪1.1
Host: localhost
```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json

{
  "result": "2.11059400",
  "status": 200
}
```

### Status Codes

- 200 OK – no error
- 400 Bad Request – bad request
- 404 Not Found – not found request
- 500 Internal Server Error – internal error
- **10001** – process error

### create offline transaction

create a offline transaction utxo json data , you should sign it using private key

**POST** /api/1/\${friendChain\_Symbol}/createTx

**Example request:**

```
POST /api/1/${friendChain_Symbol}/createTx HTTP/1.1
Host: localhost

{
  "inputs" : ["EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV"],
  "outputs" : [{
    "addr": "EPzxJrHefvE7TCWmEGQ4rcFgxGeGBZFShw",
    "amt" : 1000
  }]
}
```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": {
    "Transactions": [
      {
        "UTXOInputs": [
          {
            "address": "EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV",
            "txid":
↪ "fa9bcb8b2f3a3a1e627284ad8425faf70fa64146b88a3aceac538af8bfeffd91",
            "index": 1
          }
        ],
        "Fee": 100,
        "Outputs": [
          {
            "amount": 1000,
            "address": "EPzxJrHefvE7TCWmEGQ4rcFgxGeGBZFSHw"
          },
          {
            "amount": 99997800,
            "address": "EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV"
          }
        ]
      }
    ],
    "status": 200
  }
}

```

## send offline transaction

send raw transaction

**POST /api/1/\${friendChain\_Symbol}/sendRawTx**

**Example request:**

```

POST /api/1/${friendChain_Symbol}/sendRawTx HTTP/1.1
Host: localhost

{
  "data":
↪ "0200010013313637333832373132343538363832353937350191FDEFBFF88A53ACCE3A8AB84641A60FF7FA2584AD8
↪ "
}

```

**Example response:**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "result": "1f4432635bcf8c347f2bc20b7906c8c6c195f51beb3426e5f8d6a9e4cc073cf3",

```

(continues on next page)

(continued from previous page)

```
"status": 200
}
```

## 2.2.4 Offline transaction samples

how to send an offline transaction or cross chain transaction

### offline transaction

three steps.

1.create an offline transaction

The API *CreateTx*

the result of response is the parameter for step2.

2.sign the offline transaction

The API *generateRawTransaction*

this api will return the signed raw transaction. the return value(if not empty) is the parameter for step 3

3.send offline raw transaction

The API *SendRawTx*

### offline cross chain transaction

1.create an offline cross chain transaction

The API *CreateCrossTx*

the result of response is the parameter for step2.

2.sign the offline transaction

The API *generateRawTransaction*

this api will return the signed raw transaction. the return value(if not empty) is the parameter for step 3

3.send offline raw transaction

The API *SendRawTx*

## 2.3 Changelog

- **:release:'0.0.1 <2018-09-30>'**
- **:feature:'1'** add did creation api.
- **:feature:'1'** add did retrieve api using private key.
- **:feature:'2'** add api to sign message .
- **:feature:'2'** add api to verify message if signed by public address .
- **:feature:'3'** add Did setting info api .

- **:feature:'3'** add Did getting info from transaction .
- **:feature:'5'** add asset transfer support.
- **:feature:'6'** add offline transaction sample.



---

## HTTP Routing Table

---

### /api

```
GET /api/1/${friendChain_Symbol}/balance/ (string: `public_address`),
    30
GET /api/1/${friendChain_Symbol}/currHeight,
    29
GET /api/1/balance/ (string: `public_address`),
    11
GET /api/1/cn/mnemonic, 22
GET /api/1/createWallet, 21
GET /api/1/currHeight, 11
GET /api/1/dpos/address/ (string: `address`),
    21
GET /api/1/dpos/producer/ (string: `producer_public_key`),
    19
GET /api/1/eng/mnemonic, 22
GET /api/1/history/ (string: `addr`), 16
GET /api/1/history/ (string: `addr`) ?pageSize=(int: `pageSize`) &pageNum=(int: `pageNum`),
    17
GET /api/1/tx/ (string: `txid`), 6
GET /api/1/txs/ (int: `block_height`), 12
GET /api/1/utxos/ (string: `address`), 10
POST /api/1/${friendChain_Symbol}/createTx,
    30
POST /api/1/${friendChain_Symbol}/sendRawTx,
    31
POST /api/1/createCrossTx, 14
POST /api/1/createTx, 12
POST /api/1/createVoteTx, 13
POST /api/1/cross/m2d/transfer, 28
POST /api/1/dpos/vote, 27
POST /api/1/hd, 23
POST /api/1/sendRawTx, 16
POST /api/1/sign, 25
POST /api/1/transfer, 26
POST /api/1/tx, 7
POST /api/1/verify, 26
```