
sphinxcontrib-httpdomain Documentation

Release 0.10.0

Hong Minhee

Dec 17, 2020

Contents

1	useful links	3
1.1	General	3
2	Contents	5
2.1	Set up client	5
2.2	Getting started with the nodejs API	5
2.3	Changelog	23
	HTTP Routing Table	25



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.DID.Service.git>
2. modify application.properties
 - *node.didPrefix* change it to your own local sidechain node .
 - *did.address* change it to your own fee collection address .
3. install project
 - mvn install -Dmaven.test.skip -Dgpg.skip
4. start your project
 - java -jar did.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 Centralized 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": [
```

(continues on next page)

(continued from previous page)

```

        "EHLhCEbwViWBPwh1VhpECzYEA7jQHZ4zLv"
    ],
    "OutputIndexes": [
        0
    ],
    "CrossChainAmounts": [
        10000000
    ]
},
"vin": [
    {
        "sequence": 0,
        "txid":
↪ "ba7bd41aae0a1371d9689ad04508f0754bb4a5333386411bccbdec718ce61625",
        "vout": 1
    }
],
"payloadversion": 0,
"attributes": [
    {
        "data": "32323432343239353130383035363838303230",
        "usage": 0
    }
],
"time": 1538279155,
"programs": [
    {
        "code":
↪ "21021421976fdbe518ca4e8b91a37f1831ee31e7b4ba62a32dfe2f6562efd57806adac",
        "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

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": [
```

(continues on next page)

(continued from previous page)

```

        {
            "address": "EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV",
            "txid":
↪ "fa9bcb8b2f3a3a1e627284ad8425faf70fa64146b88a3aceac538af8bfeffd91",
            "index": 1
        }
    ],
    "Fee": 100,
    "Outputs": [
        {
            "amount": 1000,
            "address": "EPzxJrHefvE7TCWmEGQ4rcFgxGeGBZFShw"
        },
        {
            "amount": 99997800,
            "address": "EU3e23CtozdSvrtPzk9A1FeC9iGD896DdV"
        }
    ]
}
},
"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": [

```

(continues on next page)

(continued from previous page)

```

{
  "UTXOInputs": [
    {
      "address": "EZEktpUwDXyKLvq5JPRfezqA5k5G5rWa3Z",
      "txid":
        ↪ "a7f06797be1d354d27173045c63426d20f9085c5a175ea5fcc38a771c7148406",
      "index": 1
    },
    {
      "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
      "txid":
        ↪ "9e17bb1448f5b5805c682afff28226193884e84a72d8680a35ce0fcf2ec70d97",
      "index": 1
    }
  ],
  "CrossChainAsset": [
    {
      "amount": 120001,
      "address": "00000000000000000000000000000000"
    }
  ],
  "Fee": 20000,
  "Outputs": [
    {
      "amount": 130001,
      "address": "XKU4GLhFJiqAMTF6HyWQrV9pK9HcGUdfJ"
    },
    {
      "amount": 34198299,
      "address": "EZEktpUwDXyKLvq5JPRfezqA5k5G5rWa3Z"
    }
  ]
}
]
},
"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 transaction 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": [
      {
        "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
↪ 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": [
          ],
        "Outputs": [
          "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA"
        ]
      },
      {
        "Txid":
↪ "b04714ff7076e18efeb9963f99af2f10bd870d39a1774e0821b71640d6b14e52",
        "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",
```

(continues on next page)

(continued from previous page)

```

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

```

get did information

check did key's value

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

GET /api/1/did/(string: `did`)/(string: `key`)

Example request:

```

GET /api/1/did/iYnguKQcpeVyrpN6edamSkky1brvQvCW6/clark HTTP/1.1
Host: localhost

```

Example response:

```

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

{
    "result": {
        "Did": "iYnguKQcpeVyrpN6edamSkky1brvQvCW6",
        "Did_status": 1,
        "Public_key":
        ↪ "0354E3AE040052CD61A38DF72E189EDCBBB8BA81599DED25019E286F2E013A3726",
        "Property_key": "clark",
        "Property_value": "hello,world",
        "Txid": "c39dd1463678146467ea1b43d8905e75ac34e727a4c36824410089a6682b43c8
        ↪ ",
        "Block_time": 1543902641,
    }
}

```

(continues on next page)

(continued from previous page)

```

        "Height": 66480
    },
    "status": 200
}

```

2.2.3 Local Service API

If you are running code locally, you can use the following API.

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",
    "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",

```

(continues on next page)

(continued from previous page)

```

        "publicAddress": "EXoaGjh6H9afjDX7DUBY1MpsdLz4Vo16Qa",
        "did": "iczhaJiyRj4LFsw4YT6CqtQET8mFWi29Pj"
    },
    {
        "privateKey":
↪ "D0E8FB87B32EC69EC8527486AF6780DB06983F395F8394372B88F9F29F738A91",
        "publicKey":
↪ "02371040B81C28B3A194826E1F8905687E06E94D5AEB1292C72051BF6ED314888D",
        "publicAddress": "EfPZc1VdGHztzeRag9ayb4MLPZA7psGuGs",
        "did": "iz91XWKGTABjyXhZR5WrxWUCy7Xn7NvN5G"
    },
    {
        "privateKey":
↪ "5892859E60028872E63AEC018217F6A9A38AFC05F4FA4DB2FA2D07455B6C46E1",
        "publicKey":
↪ "0371EAD623897AEA29CBFB2ABC9A361E5862DCAEF68A265C6D4F8A506D3353CFC8",
        "publicAddress": "EPc6Mmx78YN8MwallqjLgSagPjAWU7LJZu",
        "did": "idpYiBkpRb6dWxbqXZsUkshoWyJNs76wdR"
    },
    {
        "privateKey":
↪ "9F8A564E7CC0E4880006B8029A1FA220A60AF426D2CD90DEB52F4CCD5E89087C",
        "publicKey":
↪ "0323671B0FB55471D885300445E25893A1EF304484C73BD6F380959BF90987A0A9",
        "publicAddress": "EPgRBCe4BwNVpWFdNhLhTEebQG35b9MoFz",
        "did": "iXzouXufWtHNvi6y9HyC4iV8eapeW7Tt jL"
    },
    {
        "privateKey":
↪ "CDA64875FED6B901595C732BD71F925C6CACF5DBC503F7C472D950975A90EF05",
        "publicKey":
↪ "027B4AEFD9208E25BD27F0416729E73187CB22B58859D40E756451D3B73079FD54",
        "publicAddress": "EKi96gzJSQj3zQ6LybXwNPmEguj66H8SjN",
        "did": "idRzqGSfQaQRA4LyjJH411DSS1VjHKW395"
    },
    {
        "privateKey":
↪ "1F3AE27B2E070E8A95257295FE602113D81A30B7554C3F77378A1CD04AEBBE53",
        "publicKey":
↪ "032063D96363FE153F85DCC0550C2753EEC407334DBACCDD933F3DC45CBC7BC1E0",
        "publicAddress": "EVVEw7p4wcfDXPMRzZhcd7TEGR1EXBG291",
        "did": "imcWdJX19HDpnCf6ErAaooEg9s2ibtnD3s"
    },
    {
        "privateKey":
↪ "9C6E82523939E11A455E2962EB0DE49211E7F4216D76836D557D9D395C59C4DB",
        "publicKey":
↪ "02622D033371BFF4DB1512F655A22B1D8783B02E703477E35B76974EA2D9C89B9",
        "publicAddress": "EbYdmcS99kCo8D44AphCBjtE4SmGLEsWw",
        "did": "iddpXhK28cvrLGNsbSzP78SkCz14gC1mHY"
    },
    {
        "privateKey":
↪ "8A69ADD6A8534C2E7D52067D8929D65354F8D218EE0FF1FA1C631A334821E85E",
        "publicKey":
↪ "024A4751A73AA5E186D83C171449FB8AF699C6EE0BC24DB34B85EEF5FA6EEDBD27",
        "publicAddress": "EcBVvLx31eN69KS3oWNLxxdaGK2UKUxg3g",

```

(continues on next page)

(continued from previous page)

```

        "did": "iqAakUhsWoRALTF1jeMxDtu4MEqLiRGZDt"
    },
    {
        "privateKey":
        ↪ "4E2C9E4C320AF08C943AADA5BC9106E053DFD2AFC8A3F229213DB6BE36DB71D8",
        "publicKey":
        ↪ "0354674BFA392313760DD66936A2BEC9CDD8DA7AB264D920FC4AF0B14704646BA5",
        "publicAddress": "EPbCbYYCwbyxFQgU9XNAWwAFRbcJtDC7Tt",
        "did": "ijVq9RQveWNeTSDfkjMwL5ksCaSuqCNsoJ"
    },
    {
        "privateKey":
        ↪ "31E5DB750E2BAC324A0C7F59B2BAB9C574232B811F7A4650352AAD31A3C2A941",
        "publicKey":
        ↪ "036A2ED5B9F57636460B4CB38A6103E03A08F8A6E5B4F0326CD303B6407E7CC303",
        "publicAddress": "ESzmxv7R4Tkt1SakscXRlyxTYtPwuXtvZP",
        "did": "iopfp7FCQ9123MXHd4jxM4hRiwxmMxyhKj"
    }
],
"status": 200
}

```

Create DID

create a did with the correspond private key.

GET /api/1/gen/did

Example Request:

```

GET /api/1/gen/did HTTP/1.1
Host: localhost

```

Example Response:

```

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

{
  "result": {
    "privateKey":
    ↪ "7C6FE5BE1014198AB84FF9F9027C44F38A1E6A0B6F02E3077F538AEB63B1661C",
    "publicKey":
    ↪ "02EB78F1AC5465819AB82898C666C542D811813E5E9F6190E68B601C0D92FF6EE8",
    "publicAddr": "EJewutbVgYHKzfFpMndFZnHNaesLjahgzR",
    "did": "iUcizvgpUDHKcNWDyPohnc3m6gp3ww8NDS"
  },
  "status": 200
}

```

Retrive DID

using private key to retrive did .

GET /api/1/did/(string: `private_key`)

Example Request:

```
GET /api/1/did/6E228664BE94833BB18DF6C66BE09173A8F42856E27CCF3DDEADE5785C16FDF7_
↪HTTP/1.1
Host: localhost
```

Example Response:

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

{
  "result": "iXMBmDBXtqTEyiKEVga9dUNqhJBvE74Ln9",
  "status": 200
}
```

Set DID information

setting information into did. the first private key is used to pay the miner fee. the second private key is the private key of did ,used to sign the info. and we give the recommended format to write a did property. the first Status is the status of the did , it can be 1 or 0, 0 stand for discard the DID, 1 stand for DID status normal ,the second Status is the did key status , it also can be 1 or 0 , 0 stand for discard the DID key , 1 stand for Did key status normal

POST /api/1/setDidInfo**Example Request:**

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

{
  "privateKey":
  ↪"C740869D015E674362B1F441E3EDBE1CBCF4FE8B709AA1A77E5CCA2C92BAF99D",
  "settings":{
    "privateKey":
    ↪"E763239857B390502289CF75FF06EEEDC3252A302C50E1CBB7E5FAC8A703486F",
    "info":{
      "Tag":"DID Property",
      "Ver":"1.0",
      "Status":1,
      "Properties": [
        {
          "Key":"clark",
          "Value":"hello,world",
          "Status": 1
        }
      ]
    }
  }
}
```

Example Response:

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

{
  "result": "1170c18870d2547207b85bd9859dc97886ca8f570399da0bcfb40a0bdc2a1b20",

```

(continues on next page)

(continued from previous page)

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

transfer DID asset using private key

using private key to send transaction

POST /api/1/did/transfer

Example request:

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

{
  "sender": [
    {
      "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": "EB291AA789D542ECE75BCF806A375BD905EDD150046FF52D51DF7D149E37FFF5",
  "status": 200
}
```

Sidechain to Mainchain transfer

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

POST /api/1/cross/d2m/transfer

Example Request:

```
POST /api/1/cross/d2m/transfer HTTP/1.1
Host: localhost
```

(continues on next page)

(continued from previous page)

```

{
  "sender": [
    {
      "address": "EbxU18T3M9ufnrkRY7NLt6sKyckDW4VAsA",
      "privateKey":
↪ "C740869D015E674362B1F441E3EDBE1CBCF4FE8B709AA1A77E5CCA2C92BAF99D"
    }
  ],
  "receiver": [
    {
      "address": "EHLhCEbwViWBPwh1VhpECzYEA7jQHZ4zLv",
      "amount": "0.089698"
    }
  ]
}

```

Example Response:

```

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

{
  "result": "3CDEB61D4CC4541591CDE4B15EB391385715C713D6709FE84381481558C2B69A",
  "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
Content-Type: application/json

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

```

(continues on next page)

(continued from previous page)

```

    },
    "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
}

```

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.

HTTP Routing Table

/api

```
GET /api/1/balance/(string:'public_address'),  
    8  
GET /api/1/cn/mnemonic, 16  
GET /api/1/createWallet, 15  
GET /api/1/currHeight, 7  
GET /api/1/did/(string:'did')/(string:'key'),  
    14  
GET /api/1/did/(string:'private_key'),  
    19  
GET /api/1/eng/mnemonic, 15  
GET /api/1/gen/did, 19  
GET /api/1/history/(string:'addr'), 12  
GET /api/1/history/(string:'addr')?pageSize=(int:'pageSize')&pageNum=(int:'pageNum'),  
    12  
GET /api/1/tx/(string:'txid'), 6  
GET /api/1/txs/(int:'block_height'), 9  
POST /api/1/createCrossTx, 10  
POST /api/1/createTx, 9  
POST /api/1/cross/d2m/transfer, 21  
POST /api/1/did/transfer, 21  
POST /api/1/hd, 17  
POST /api/1/sendRawTx, 11  
POST /api/1/setDidInfo, 20  
POST /api/1/sign, 22  
POST /api/1/verify, 23
```