# JCC\_RPC\_JAVA

[1 JCallback 2](#_Toc195094669)

[2 Exchange API (JccdexExchang API) 3](#_Toc1896056561)

[2.1 Usage 3](#_Toc512782894)

[2.2 Get wallet balances 3](#_Toc490224047)

[2.3 Get historic transactions 4](#_Toc1448288037)

[2.4 Get orders 8](#_Toc1797382761)

[2.5 Create order 9](#_Toc2107085425)

[2.6 Cancel order 9](#_Toc1779398945)

[2.7 Get sequence 10](#_Toc500800493)

[2.8 Transfer 10](#_Toc965473258)

[2.9 Get transaction detail 11](#_Toc322610474)

[3 Info API(JccdexInfo API) 13](#_Toc1865511490)

[3.1 Usage 13](#_Toc390366230)

[3.2 Get 24-hour market data for the specified currency 13](#_Toc322686025)

[3.3 Get 24-hour market data for all currencies 14](#_Toc987813500)

[3.4 Get market depth 15](#_Toc2132903190)

[3.5 Get kline data 16](#_Toc1906878606)

[3.6 Get time-sharing data (get historical deals) 17](#_Toc2010266861)

[3.7 Get inter-currency exchange rates 18](#_Toc194914576)

[4 Config API(JccConfig API) 18](#_Toc1016717157)

[4.1 Usage 18](#_Toc437878520)

[4.2 Get config data 19](#_Toc2145311018)

[5 Explorer API(JccExplore API) 19](#_Toc2139330043)

[5.1 Usage 19](#_Toc401330980)

[5.2 Get wallet balances 19](#_Toc2071129280)

[5.3 Get transaction detail by hash 20](#_Toc907374737)

[5.4 Get historic transactions by address 23](#_Toc965827412)

[5.5 Get payment transactions by address and time 25](#_Toc1979909458)

[5.6 Get transactions by token or time 25](#_Toc1079150341)

[5.7 Get all tokens list 27](#_Toc1780382272)

[5.8 Get top-100 addresses for specified token 27](#_Toc1995191853)

[5.9 Get the ranking of the specified currency in the specified wallet 29](#_Toc232325466)

[5.10 Get the tokens issued by the issuer address 29](#_Toc568836816)

## 1 JCallback

/\*\*

\* @param code if code is 0 means request success

\* @param response

\*/

void onResponse(String code, String response);

void onFail(Exception e);

## 2 Exchange API **(JccdexE**xchang API)

### 2.1 Usage

Create JccdexUrl:

JccdexUrl jccUrl = new JccdexUrl("xxx", true);

JccdexUrl jccUrl = new JccdexUrl("xxx", false);

or

JccdexUrl jccUrl = new JccdexUrl("xxx", true, 8081);

Create JccdexExcahnge:

JccdexExchange exchange = JccdexExchange.getInstance();

exchange.setmBaseUrl(jccUrl);

### 2.2 Get wallet balances

exchange.requestBalance(address, callback);

Parameters：

address：wallet address

callback: callback function

Result：

{

"code": "0",

"data": [

{

"value": "171.9799110000",

"currency": "SWT",

"issuer": "",

"freezed": "42.0000000000",

"reserve": "20",

"title": "SWT"

},

{

"value": "0",

"currency": "CNY",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"freezed": "0.0000000000",

"title": "CNY"

},

{

"value": "0",

"currency": "JJCC",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"freezed": "0.0000000000",

"title": "JJCC"

}

],

"msg": "",

"isActive": true

}

Result parameters:：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | | Type | Description |
| code | | String | 0 means request success |
| msg | | String | message |
| isActive | | Boolean | Wallet is active or not |
| data | | Array | Balance Object |
|  | value | String | amount |
| currency | String | Token name |
| issuer | String | Issuer address |
| freezed | String | Freezed value |
| reserve | String | Value of token required to activate the wallet |
| title | String | Token mirror name |

### 2.3 Get historical transactions

exchange.requestHistoricTransactions(address,page,ledger,seq,callback);

Parameters：

Address

page

ledger

seq

Result：

{

"code": "0",

"data": {

"transactions": [

{

"date": 1573614140,

"hash": "2A9D8789DC22F4CE9D6A773ED6DF67DC429A456EC54A773A75D298CA551DFBE3",

"type": "offercancel",

"fee": "0.00001",

"result": "tesSUCCESS",

"memos": [],

"offerseq": 38,

"effects": [

{

"effect": "offer\_cancelled",

"hash": "CB191C4C6AE3B07A077D042CB927AAB47D63B05778389C3793220198ADBF3C27",

"gets": { "value": "1", "currency": "SWT",

"issuer": "" },

"pays": {

"currency": "JJCC",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"value": "1"

},

"type": "buy",

"seq": 37,

"price": "1",

"deleted": true

}

],

"gets":{ "value":"1","currency":"SWT","issuer":"" },

"pays": {

"currency": "JJCC",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"value": "1"

}

}

],

"marker": { "ledger": 14324732, "seq": 3 }

},

"msg": "",

"isActive": true

}

Result：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameters | | | | | Type | Description |
| data | | | | | Object |  |
|  | marker | | | | Object | Marker |
|  | ledger | | | Number | Ledger |
| seq | | | Number | The last transaction number returned by the previous query |
| transactions | | | | Array | Transaction data |
|  | hash | | | String |  |
| date | | | Timestamp |  |
| type | | | String |  |
| fee | | | String |  |
| result | | | String | If result is tesSUCCESS means successful |
| memos | | | Array |  |
| offerseq | | | Number |  |
| effects | | | Array |  |
|  | effect | | String |  |
| hash | | String |  |
| gets | | Object |  |
|  | value | String |  |
| currency | String |  |
| issuer | String |  |
| pays | | Object |  |
|  | value | String |  |
|  |  |  | currency | String |  |
| issuer | String |  |
| type | | String |  |
| seq | | Number |  |
| price | | String |  |
| deleted | | Boolean |  |
| gets | | | Object |  |
|  | value | | String |  |
| currency | | String |  |
| issuer | | String |  |
| pays | | | Object |  |
|  | value | | String |  |
| currency | | String |  |
| issuer | | String |  |

Transaction Type Description:

1.sent

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | | Type | Description |
| counterparty | | String | Received address |
| amount | | Object |  |
|  | value | String |  |
| currency | String |  |
| issuer | String |  |
| effects | | Array | Is empty array |

2.received

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| counterparty | | String | Received address |
| amount | | Object |  |
|  | value | String |  |
| currency | String |  |
| issuer | String |  |
| effects | | Array | Is empty array |

3.convert

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| spent | | Object |  |
|  | value | String |  |
| currency | String |  |
| issuer | String |  |
| amount | | Object |  |
|  | value | String |  |
| currency | String |  |
| issuer | String |  |
| effects | | Array | Is empty array |

4.offernew：

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| offertype | String | Sell or buy |
| pair | String |  |
| amount | String |  |
| price | String |  |
| effects | Array |  |

5.offercancel

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| type | String | Sell or buy |
| pair | String |  |
| amount | String |  |
| price | String |  |
| effects | Array |  |

6.offereffect

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| effects | Array |  |

Transaction state：

offer\_funded，Actual transaction

offer\_partially\_funded，Partial transaction

offer\_cancelled，Cancelled orders for related transactions, transaction orders cancelled

offer\_created，create order

offer\_bought，buy or sell order

### 2.4 Get orders

exchange.requestOrders(address, page, callback);

Parameter：

address

page

Result：

{

"code": "0",

"data": [

{

"type": "sell",

"pair": "SWT/CNY+jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"price": "0.0100000000",

"amount": "1.0000000000",

"sequence": 35,

"passive": false

}

],

"msg": "",

"isActive": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Array |  |
|  | type | String | (sell/buy) |
| pair | Sring |  |
| price | String |  |
| amount | String |  |
| sequence | Number |  |
| passive | Boolean |  |

### 2.5 Create order

exchange.createOrder(signature, callback)

Parameter：

signature

Result：

{

"code": "0",

"data": {

"hash": "9364F31B5DF34BEA5F0A0039D8A0E3D7C296403C1A6CEFA7774FD9D0CFE3430F"

},

"msg": "",

"isActive": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Object |  |
|  | hash | String |  |

### 2.6 Cancel order

exchange.cancelOrder(signature, callback);

Parameter：

signature

Result：

{

"code": "0",

"data": {

"hash": "467FBAB2C54F0F4321ABFD045BAD7BFBFA8ED21F5BCE82997A1072F932B5F56C"

},

"msg": "",

"isActive": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Object |  |
|  | hash | String |  |

### 2.7 Get sequence

exchange.requestSequence(address);

Parameter：

address

Result：latest sequence

### 2.8 Transfer

exchange.transferToken(signature, callback);

Parameter：

signature

Result：

{

"code": "0",

"data": {

"hash": "C173D26816EC647F521441785C0B1684060A93A5142A02913468CABA93888E8B"

},

"msg": "",

"isActive": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Object |  |
|  | hash | String |  |

### 2.9 Get transaction detail

exchange.requestOrderDetail(hash, callback);

Parameter：

Hash

Result：

{

"code": "0",

"data": {

"Account": "jU3EZPPFbKHzoUi65jcXVZDx3QE13uJFQc",

"Amount": {

"currency": "CSP",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"value": "1"

},

"Destination": "jDKQVYNBwcPfcztUQsjxKkAWDQgQXXm4gL",

"Fee": "10",

"Flags": 0,

"Memos": [

{

"Memo": {

"MemoData": "E696B0E794A8E688B7E6B3A8E5868CE8B5A0E98081E58CBAE59D97E5B8813B20353238373738",

"MemoType": "737472696E67"

}

}

],

"Sequence": 20067,

"SigningPubKey": "02F60A05977987D77AD4D6C188D1C51685EC05BED50EB6DC986511907C4873828E",

"TransactionType": "Payment",

"TxnSignature": "30450221009E367C3BD7CAB7B1FAED45CAB10EF3D4F07ED02B73AEAEE2B111FEB548E5A0BA02205B62B1BEB09E8D443091C182C425E412593B2AC6E1ABC266898245BC7D077101",

"date": 627049150,

"hash": "23B46F65B855B98B24D5AB94ADE8829EE03C8840C7B58C8E09870F14773949CC",

"inLedger": 14388884,

"ledger\_index": 14388884,

"meta": {

"AffectedNodes": [

{

"ModifiedNode": {

"FinalFields": {

"Account": "jU3EZPPFbKHzoUi65jcXVZDx3QE13uJFQc",

"Balance": "29799330",

"Flags": 0,

"OwnerCount": 1,

"Sequence": 20068

},

"LedgerEntryType": "AccountRoot",

"LedgerIndex": "F59FC7650C29E645E494AFE10E126FA90355EDBC3B21976E55520C3E21B67E06",

"PreviousFields": { "Balance": "29799340", "Sequence": 20067 },

"PreviousTxnID": "06211E3643FF10AE5924985145BE3DA8CEF38F7D495EF36023B75A9CFD881FFE",

"PreviousTxnLgrSeq": 14388832

}

}

],

"TransactionIndex": 0,

"TransactionResult": "tesSUCCESS",

"delivered\_amount": {

"currency": "CSP",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"value": "1"

}

},

"validated": true

},

"msg": "",

"isActive": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| Account | | String |  |
| Amount | | Object |  |
|  | currency | String |  |
| issuer | String |  |
| value | String |  |
| Destination | | String |  |
| Fee | | String |  |
| Flags | | Number |  |
| Memos | | Array |  |
| Sequence | | Number |  |
| SigningPubKey | | String |  |
| TransactionType | | String |  |
| date | | Number |  |
| hash | | String |  |
| inLedger | | Numer |  |
| ledger\_index | | Number |  |
| TxnSignature | | String |  |

## 3 Info API(JccdexInfo API)

### 3.1 Example

Create JccdexUrl:

JccdexUrl jccUrl = new JccdexUrl("xxx", true);// https://xxx:443

JccdexUrl jccUrl = new JccdexUrl("xxx", false);// http://xxx:80

or

JccdexUrl jccUrl = new JccdexUrl("xxx", true, 8081);

Create JccdexInfo:

JccdexInfo info = JccdexInfo.getInstance();

info.setmBaseUrl(jccUrl);

### 3.2 Get 24-hour market data for the specified currency

info.requestTicker(base, counter, callBack);

Parameter：

base: token name

counter: token name

Result：

{

"code": "0",

"data": [

1573703178,

"0.005600000000015",

1.8181633060612183,

"0.00564996696339",

"0.005459999999997",

565422.5980797218,

101822158.77275103,

166

],

"msg": "",

"success": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Array |  |
|  | data[0] | Timestamp | Current time |
| data[1] | String | Current price |
| data[2] | Number | Daily change (%) |
| data[3] | String | Highest price |
| data[4] | String | Lowest price |
| data[5] | Number | Daily turnover |
| data[6] | Number | Daily volume |
| data[7] | Number | Number of transactions |

### 3.3 Get 24-hour market data for all currencies

info.requestAllTickers(callBack);

Result：

{

"code": "0",

"data": {

"ECP-SWT": [1573711152, 0.45],

"JMOAC-SWT": [

1573711152,

"476",

1.4925373134324422,

"476.0000000000022",

"469",

2685353.026836,

5662.984673403357,

17

]

},

"msg": "",

"success": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
|  | Array[0] | Timestamp | Current time |
| Array[1] | String | Current price |
| Array[2] | Number | Daily change (%) |
| Array[3] | String | Highest price |
| Array[4] | String | Lowest price |
| Array[5] | Number | Daily turnover |
| Array[6] | Number | Daily volume |
| Array[7] | Number | Number of transactions |

### 3.4 Get market depth

info.requestDepth(base, counter, type, callBack);

Parameter：

base: token name

counter: token name

type: normal/more，normal data length is 5，more data length is 50

Result：

{

"code": "0",

"data": {

"base": "SWT",

"counter": "CNY.jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"asks": [

{

"price": "0.0055",

"amount": "0.204754",

"total": "0.204754",

"type": "sell"

}

],

"bids": [

{

"price": "0.005480011",

"amount": "1500000",

"total": "1500000",

"type": "buy"

}

]

},

"msg": "",

"success": true

}

Result：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | | | Type | Description |
| data | | | Object |  |
|  | base | | String | Token name |
| counter | | String | Token name |
| asks | | Array | sell |
|  | price | String |  |
| amount | String |  |
| total | String |  |
| type | String |  |
| bids | | Array | buy |
|  | price | String |  |
| amount | String |  |
| total | String |  |
| type | String |  |

### 3.5 Get kline data

info.requestKline(base, counter, type, callBack);

Parameter：

base: token name

counter: token name

type: minute、5minute、15minute、30minute、hour、4hour、day、week、mouth

Result：

{

"code": "0",

"data": [

[

1570262400000,

"0.002971027566921",

"0.00296",

"0.002958536564484",

"0.003040000000002",

79113.63096444756,

26375594.990000002,

42

]

],

"msg": "",

"success": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Array |  |
|  | Array[0] | Timestamp | time |
| Array[1] | String | Opening price |
| Array[2] | String | Closing price |
| Array[3] | String | Lowest price |
| Array[4] | String | Highest price |
| Array[5] | Number | turnover |
| Array[6] | Number | volume |
| Array[7] | Number | Number of transactions |

### 3.6 Get time-sharing data (get historical deals)

info.requestHistory(base, counter, type, time, callBack);

Parameter：

base: token name

counter: token name

type: all、more、newest，if type is newest，need time

time

Result：

{

"code": "0",

"data": [

[

"18667.894616639613",

"3357598.8601470003",

"0.005559894256047",

1573716050000,

1,

1

]

],

"msg": "",

"success": true

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| data | | Array |  |
|  | Array[0] | String | turnover |
| Array[1] | String | volume |
| Array[2] | String | price |
| Array[3] | Timestamp | time |
|  | Array[4] | Number | change(0:rise，1:fall) |
|  | Array[5] | Number | Match flag(1:not，3: three match) |

### 3.7 Get inter-currency exchange rates

info.requestTickerFromCMC(token, currency, callBack)

Parameter：

Token: support eth、btc

currency: support cny、rub

Result：

{

"data": {

"quotes": { "CNY": { "price": 1304.4 } },

"last\_updated": 1570893901.299,

"id": 3,

"symbol": "ETH"

}

}

Result：

price: (1ETH = 1304.4CNY)

## 4 Config API(JccConfig API)

### 4.1 Example

Create JccdexUrl:

JccdexUrl jccUrl = new JccdexUrl("weidex.vip", true);

Create JccConfig:

JccConfig config = JccConfig.getInstance();

config.setmBaseUrl(jccUrl);

### 4.2 Get config data

config.requestConfig(callBack);

Result:

exHosts：exchange api hosts

infoHosts：info api hosts

cfgHosts： config api hosts

scanHosts：explorer api hosts

## 5 Explorer API(JccExplore API)

### 5.1 Example

Create JccdexUrl:

JccdexUrl jccUrl = new JccdexUrl("expjia1b6719b6e6.jccdex.cn", true);

Create JccExplore:

JccExplore explore = JccExplore.getInstance();

explore.setmBaseUrl(jccUrl);

### 5.2 Get wallet balances

explore.requestBalance(uuid, address, callBack);

Parameter：

uuid：unique id

address：wallet address

Result：

{

"code": "0",

"data": {

"\_id": "jfyiDN3XrbdPuAzWSwnx49DNsdkX6jqz12",

"SWTC": { "value": "169.979881", "frozen": "42" },

"feeflag": 0

},

"msg": ""

}

Result：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| code | | String | 0 means succssful |
| msg | | String | message |
| data | | Object |  |
| \_id | | String | Wallet address |
| SWTC | | Object | Token name |
|  | value | String | amount(include freezed) |
| frozen | String | Freezed amount |

### 5.3 Get transaction detail by hash

explore.requestTransDetails(uuid, hash, callBack);

Parameter：

hash： transaction hash

Result：

{

"code": "0",

"data": {

"\_id": "2D19BDF2AC03554087C9C3CFB3005958DAF5748C9EF473FA1C4E74EE1C6C8932",

"hashType": 2,

"upperHash": "69F0D9A6F18413A8F18120FB4E70CC3ADA1FD4D8E59AEC16FA9978ACF506E0FF",

"block": 14387372,

"time": 627034030,

"type": "Payment",

"account": "jhAiUKdoEXD3HBBMbsfmmeRynExhp355NM",

"seq": 55,

"fee": 0.01,

"succ": "tesSUCCESS",

"dest": "jJBPpjyFBrsevf1udWTEn3tjiHpF2Jw6ui",

"amount": {

"currency": "VCC",

"issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"value": "5"

},

"memos": [{ "Memo": { "MemoData": "00" } }],

"past": 121

},

"msg": ""

}

Result：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data | | | Object |  |
|  | \_id | | String | Transaction hash |
| hashType | | Number | Hash type(1:block hash,2:transaction hash) |
| upperHash | | String | Owned block hash |
| block | | Number | Block height |
| time | | Number | (time\*1000+ 946684800 = unix timstamp) |
| type | | String | Transaction type(Payment, OfferCreate, OfferCancel) |
| account | | String | Wallet address |
| seq | | Number | sequence |
| fee | | Number | gas |
| succ | | String | result，tesSUCCESS means success |
| dest | | String | Counterparty |
| amount | | Obiect | Transfer amount |
|  | currency | String | Token name |
| issuer | String | Issuer address |
| value | String | amount |
| memos | | Array |  |
| past | | Number | Seconds from transaction |

If type is different，the response may be different

1). when type is Payment

dest：destination address

amount：transfer token and amount，json object，examplee：{"currency": "CNY", "issuer": "\*\*\*", "value" : "100" }

memos：remarks，array，example：[ { "Memo" : { "MemoData" : "7B227472616E735F74797065223A226C6F67696E5F726577617264227D", "MemoType" : "737472696E67" } } ] (MemoData is hex, need convert to string)

2). when type is OfferCreate

flag：buy or sell，（1:buy；2:sell；0:unknown）

takerGets：paid token，example：{"currency":"CNY","issuer":"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or", "value" : "100" }

takerPays：got token，json object，liken takerGets

realGets：Actual paid token

realPays：actual got token

matchGets：actual deal paid token

matchPays：

matchFlag：

affectedNodes：json ，example：[ { "account" : "\*\*\*", "seq" : \*\*\*, "flag" : \*, "previous" : { "takerGets" : { "currency" : "CNY", "issuer" : "\*\*\*", "value" : "39" }, "takerPays" : { "currency" : "SWTC", "issuer" : "", "value" : "1000" } }, "final" : { "takerGets" : { "currency" : "CNY", "issuer" : "\*\*\*", "value" : "0" }, "takerPays" : { "currency" : "SWTC", "issuer" : "", "value" : "0" } } }, ...... ]

a). account：

b). seq：

c). flag：（1:buy；2:sell；0:unknown）

d). previous：

e). final

3). when type is OfferCancel

offerSeq

takerGets：example：{"currency":"CNY","issuer":"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or", "value" : "100" }

takerPays：

when hash is block hash，

Result：

{

"code": "0",

"data": {

"info": {

"\_id": "\*\*\*",

"hashType": 1,

"upperHash": "",

"block": 8488996,

"time": 568047820,

"past": "\*\*\*",

"transNum": 1,

"parentHash": "\*\*\*",

"totalCoins": "599999999999460713"

},

"list": [

{

"\_id": "\*\*\*",

"index": 0,

"type": "OfferCreate",

"account": "\*\*\*",

"seq": 634,

"fee": 0.01,

"succ": "tesSUCCESS"

}

],

"count": 16

},

"msg": ""

}

### 5.4 Get historic transactions by address

explore.requestHistoricTransWithAddr(uuid, page, size, begin, end, type, currency, address, callBack);

Parameter：

page: (begin 0)

size: (10/20/50/100,default 20)

begin: (format:2019-10-11)

end：like begin

type： (Send、Receive)

currency：

address：

Result：

{

"code": "0",

"data": {

"list": [

{

"type": "Receive",

"time": 627017440,

"hash": "C173D26816EC647F521441785C0B1684060A93A5142A02913468CABA93888E8B",

"block": 14385713,

"fee": "",

"success": "tesSUCCESS",

"seq": 43,

"account": "jfyiDN3XrbdPuAzWSwnx49DNsdkX6jqz12",

"amount": { "currency": "SWTC", "issuer": "", "value": "2" },

"memos": [

{ "Memo": { "MemoData": "74657374", "MemoType": "737472696E67" } }

],

"past": 18413

}

],

"count": 302

},

"msg": ""

}

Parameter：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| type | | String | (Send,Receive) |
| time | | Number |  |
| hash | | String |  |
| block | | Number |  |
| fee | | String |  |
| seq | | Number |  |
| account | | String |  |
| memos | | Array |  |
| past | | Number |  |
| amount | | Object |  |
|  | currency | String |  |
| issuer | String |  |
| value | String |  |

### 5.5 Get payment transactions by address and time

explore.requestPaymentSummary(uuid, address, dateTpye, begin, end, type, token, callBack);

Parameter：

address：

dateType：(day=2, month=3, year=4)

begin：

type：Send/Receive

token：

Result：

{

"code": "0",

"data": [

{

"wallet": "jKBCwv4EcyvYtD4PafP17PLpnnZ16szQsC",

"token": "SWTC\_",

"type": "Send",

"num": 1,

"amount": 100

}

],

"msg": ""

}

Parameter：

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| wallet | String |  |
| token | String |  |
| type | String |  |
| num | Number |  |
| amount | Number |  |

### 5.6 Get transactions by token or time

explore.requestHistoricTransWithCur(uuid, page, size, begin, end, type, currency, callBack);

Parameter：

page：begin 0

size：default 20，10/20/50/100

begin：

end

type：Payment

currency：token name

Result：

{

"code": "0",

"data": {

"list": [

{

"\_id": "5935EB56AF33524C300A6892406F93AA061CC94D2FED4E3D5A0E2F88407A0D40",

"hashType": 2,

"time": 620668790,

"index": 1,

"type": "Payment",

"account": "jpJX5RFCMV1F6p1zUnGTkaQjXoqWc4CKGR",

"seq": 4868,

"fee": 0.0001,

"succ": "tesSUCCESS",

"dest": "jKj7choWbwaAMiLo4dCL1pEn9JqvgAc2Nh",

"amount": { "currency": "SWTC", "issuer": "", "value": "30" }

}

],

"count": 7166

},

"msg": ""

}

Parameter：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| \_id | | String | hash |
| hashType | | Number |  |
| time | | Number |  |
| index | | Number |  |
| type | | String |  |
| account | | String |  |
| seq | | Number |  |
| fee | | Number |  |
| dest | | String |  |
| succ | | String |  |
| amount | | Object |  |
|  | currency | String |  |
| issuer | String |  |
| value | Strign |  |

### 5.7 Get all tokens list

curl [stats.jccdex.cn/sum/all/:uuid](http://stats.jccdex.cn/sum/all/:uuid)

curl [stats.jccdex.cn/sum/all/:uuid](http://stats.jccdex.cn/sum/all/:uuid)?t=C

Parameter：

t: token name

Result：

{

"code": "0",

"data": [

{

"J": [

"JSLASH\_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or"

]

}

],

"msg": ""

}

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
|  | | Array |  |
|  |  | String |  |

### 5.8 Get top-100 addresses for specified token

curl [stats.jccdex.cn/sum/list/:uuid?t=JSLASH\_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or&p=4](http://stats.jccdex.cn/sum/list/:uuid?t=JSLASH_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or&p=4)

Parameter：

t：token name

p：

s：default 20，10/20/50/100

b：

e：

Result：

{

"code": "0",

"data": [

{

"tokens": "JSLASH\_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or"

},

{

"holders": 0

},

{

"totalsupply": 199999412

},

{

"circulation": 136375233.93999976

},

[

{

"address": "jsk45ksJZUB7durZrLt5e86Eu2gtiXNRN4",

"time": 627096607,

"amount": "63618238.06000023"

}

],

[

"jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd",

"j4TVCesxU2s6tjYpsDoH4Pc81c4WMvNHPo",

"jUq1KR6YHHAWxTb8MApaeCLfkBS6a8Y4nX",

"jnoqBzpAdWixYzwcwFj8Q3SDBrpR6QKRgK",

"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

"jsk45ksJZUB7durZrLt5e86Eu2gtiXNRN4",

"jG9ntUTuBKqDURPUqbGYZRuRDVzPY6bpxL",

"jMCJrXRmycsT5tsVuge7Y65v9MrQi9r11E",

"japp9xxt2VHpRwHsoa76GWoQj1VdsjcZQJ",

"jDu7umDxKxeaHoj7eNdUn8YsGWTHZSuEGL",

"jQs5cAcZrKmyWSQgkmUtXsdeFMzwSYcBA4",

"jhAvDrBjDV4HFRmRKZ3ZnL77Sg4PCezAwE"

],

{

"flag": 1

},

{

"issueDate": 594797250

}

],

"msg": ""

}

Parameter：

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | | Type | Description |
| tokens | | String |  |
| totalsupply | | Number |  |
| circulation | | Number |  |
|  | | Array |  |
|  | address | String |  |
| time | Number |  |
| amount | String |  |
|  | | Array |  |
| flag | | Number |  |
| issueDate | | Number |  |

### 5.9 Get the ranking of the specified currency in the specified wallet

curl  [stats.jccdex.cn/sum/self/:uuid?t=CSP\_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or&w=jKectkVmsURkNbrJ7EEd1DQPngPCmDn1sQ](http://stats.jccdex.cn/sum/self/:uuid?t=CSP_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or&w=jKectkVmsURkNbrJ7EEd1DQPngPCmDn1sQ)

Parameter：

t：token

w：wallet addreess

result：

{

"code": "0",

"data": 422415,

"msg": ""

}

Parameter：

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| data | Number |  |

### 5.10 Get the tokens issued by the issuer address

get method：<https://stats.jccdex.cn/wallet/fingate_tokenlist/:uuid?w=jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd>

Parameters：

w：wallet address

Result：

{

"code": "0",

"data": {

"jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd": [

{ "currency": "YYT", "issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or" },

{ "currency": "HPT", "issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or" }

]

},

"msg": ""

}

Result parameters：

|  |  |  |
| --- | --- | --- |
| Parameters | Type | Description |
| currency | String | Token name |
| issuer | String | Issuer address |