

# JCC\_RPC\_JAVA

1 JCallback.....	3
2 交易接口 (JccdexExchang API).....	3
2.1 使用示例.....	3
2.2 获取钱包余额.....	3
2.3 获取历史交易.....	4
2.4 获取钱包挂单.....	8
2.5 创建挂单.....	9
2.6 撤销挂单.....	10
2.7 获取序列号.....	10
2.8 转账.....	11
2.9 获取订单详情.....	11
3 数据接口(JccdexInfo API).....	14
3.1 使用示例.....	14
3.2 获取指定币种 24 小时的行情数据.....	14
3.3 获取所有币种 24 小时的行情数据.....	15
3.4 获取市场深度.....	16
3.5 获取 K 线数据.....	17
3.6 获取分时数据(获取历史成交).....	18
3.7 获取币种间汇率.....	19

4 配置接口(JccConfig API).....	19
4.1 使用示例.....	19
4.2 获取配置数据.....	19
5 浏览器接口(JccExplore API).....	20
5.1 使用示例.....	20
5.2 获取钱包余额.....	20
5.3 根据 hash 获取详细交易信息.....	21
5.4 获取指定钱包的历史交易.....	25
5.5 获取钱包每天/月/年支付或收到的币种笔数和数量.....	27
5.6 查询某个 token 在某个时间段内的交易 hash 信息.....	28
5.7 获取所有 tokens 列表.....	29
5.8 获取某种 tokens 的 100 排名列表 .....	29
5.9 获取指定钱包指定币种的排名.....	31
5.10 获取银关地址发行过的 token.....	32

## 1 JCallback

```
/**
 * @param code code 为 0, 表示响应结果正确
 * @param response
 */
void onResponse(String code, String response);
void onFail(Exception e);
```

## 2 交易接口 (JccdexExchang API)

### 2.1 使用示例

```
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 获取钱包余额

```
exchange.requestBalance(address, callback);
```

参数:

address: 井通钱包地址

callback: 回调, 即 JCallback, 后面相同, 不在重复描述

结果如下:

```
{
  "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": "获得余额成功",
"isActive": true
}
```

结果参数:

参数		类型	说明
code		String	0 表示响应正确(以下所有接口返回格式都相同, 不再重复描述)
msg		String	接口返回的状态信息描述(以下所有接口返回格式都相同, 不再重复描述)
isActive		Boolean	当前钱包是否激活(以下所有接口返回格式都相同, 不再重复描述)
data		Array	余额数组
	value	String	余额
	currency	String	token
	issuer	String	发行方
	freezed	String	冻结金额
	reserve	String	激活钱包所需币种数量
	title	String	币种名称

### 2.3 获取历史交易

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

参数:

address: 井通钱包地址

page: 当前页

ledger: 账本号

seq: 交易号

结果如下:

```
{
  "code": "0",
  "data": {
    "transactions": [
      {
        "date": 1573614140,
        "hash":
"2A9D8789DC22F4CE9D6A773ED6DF67DC429A456EC54A773A75D298CA551
DFBE3",
        "type": "offercancel",
        "fee": "0.00001",
        "result": "tesSUCCESS",
        "memos": [],
        "offerseq": 38,
        "effects": [
          {
            "effect": "offer_cancelled",
            "hash":
"CB191C4C6AE3B07A077D042CB927AAB47D63B05778389C3793220198ADBF
3C27",
            "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": "获取历史交易成功",
"isActive": true
}

```

结果参数:

参数			类型	说明
data			Object	
	marker		Object	记录标记
		ledger	Number	账本号
		seq	Number	上一次查询返回的最后一条交易号
	transactions		Array	历史交易记录数据
		hash	String	交易 hash
		date	Timestamp	交易时间
		type	String	交易类型(详情如下)
		fee	String	燃料费
		result	String	结果, tesSUCCESS 表示成功
		memos	Array	转账备注
		offerseq	Number	取消单号
		effects	Array	
			effect	交易状态(详情如下)
			hash	交易 hash
			gets	Object
			value	数量
			currency	token
			issuer	发行方
			pays	Object
			value	数量
			currency	token
			issuer	发行方
			type	挂单类型 (buy/sell)
			seq	交易号
			price	价格

			deleted	Boolean	对应 effect, 交易是否成功
			gets	Object	
			value	String	数量
			currency	String	token
			issuer	String	发行方
			pays	Object	
			value	String	数量
			currency	String	token
			issuer	String	发行方

交易类型(type)详细说明:

1.sent, 用户进行支付操作, 参数如下:

参数	类型	说明
counterparty	String	支付对家, 即接收方
amount	Object	交易记录标记
	value	String 金额
	currency	String 货币
	issuer	String 货币发行方, SWT 为空
effects	Array	[], 空

2.received, 用户接受支付, 参数如下:

参数	类型	说明
counterparty	String	支付对家, 即接收方
amount	Object	交易记录标记
	value	String 金额
	currency	String 货币
	issuer	String 货币发行方, SWT 为空
effects	Array	[], 空

3.convert, 用户进行兑换操作, 参数如下:

参数	类型	说明
spent	Object	兑换支付的金额
	value	String 金额
	currency	String 货币
	issuer	String 货币发行方, SWT 为空
amount	Object	交易记录标记
	value	String 金额

	currency	String	货币
	issuer	String	货币发行方, SWT 为空
effects		Array	[], 空

4.offernew, 用户进行挂单操作, 参数如下:

参数	类型	说明
offertype	String	挂单类型, sell 或 buy
pair	String	交易的货币对
amount	String	挂单的数量
price	String	挂单的价格
effects	Array	详见下面的 effects 解释

5.offercancel, 用户进行取消挂单操作, 参数如下:

参数	类型	说明
type	String	挂单的类型, sell 或 buy
pair	String	交易的货币对
amount	String	挂单的数量
price	String	挂单的价格
effects	Array	详见下面的 effects 解释

6.offereffect, 挂单成交情况, 即被动成交的情况, 参数如下:

参数	类型	说明
effects	Array	详见下面的 effects 解释

交易状态(effect)详细说明:

offer\_funded, 交易实际成交

offer\_partially\_funded, 交易部分成交

offer\_cancelled, 被关联交易取消单子, 交易单子被取消

offer\_created, 交易单子创建

offer\_bought, 挂单买到/卖出, 成交单子的信息

## 2.4 获取钱包挂单

```
exchange.requestOrders(address, page, callback);
```

参数:

address: 井通钱包地址

page: 当前页

结果如下:

```
{
```



```

"code": "0",
"data": [
  {
    "type": "sell",
    "pair": "SWT/CNY+jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",
    "price": "0.0100000000",
    "amount": "1.0000000000",
    "sequence": 35,
    "passive": false
  }
],
"msg": "获得挂单列表成功",
"isActive": true
}

```

结果参数:

参数		类型	说明
data		Array	挂单数组
	type	String	挂单类型(sell/buy)
	pair	Sring	挂单币种
	price	String	挂单金额
	amount	String	挂单数量
	sequence	Number	挂单序列号
	passive	Boolean	

## 2.5 创建挂单

```
exchange.createOrder(signature, callback);
```

参数:

signature: 签名

结果如下:

```

{
  "code": "0",
  "data": {
    "hash":
"9364F31B5DF34BEA5F0A0039D8A0E3D7C296403C1A6CEFA7774FD9D0CFE3
430F"
  },
  "msg": "挂单提交成功",
  "isActive": true
}

```

```
}
```

结果参数:

参数		类型	说明
data		Object	
	hash	String	挂单 hash

## 2.6 撤销挂单

```
exchange.cancelOrder(signature, callback);
```

参数:

signature: 签名

结果如下:

```
{
  "code": "0",
  "data": {
    "hash":
"467FBAB2C54F0F4321ABFD045BAD7BFBFA8ED21F5BCE82997A1072F932B5
F56C"
  },
  "msg": "取消挂单提交成功",
  "isActive": true
}
```

结果参数:

参数		类型	说明
data		Object	
	hash	String	撤销订单 hash

## 2.7 获取序列号

```
exchange.requestSequence(address);
```

参数:

address: 井通钱包地址

返回结果: 最新的序列号

## 2.8 转账

```
exchange.transferToken(signature, callback);
```

参数:

signature: 签名

结果如下:

```
{
  "code": "0",
  "data": {
    "hash":
"C173D26816EC647F521441785C0B1684060A93A5142A02913468CABA93888E8
B"
  },
  "msg": "提交支付成功",
  "isActive": true
}
```

结果参数:

参数		类型	说明
data		Object	
	hash	String	转账 hash

## 2.9 获取订单详情

```
exchange.requestOrderDetail(hash, callback);
```

参数:

hash: 交易 hash

结果如下:

```
{
  "code": "0",
  "data": {
    "Account": "jU3EZPPFbKHzoUi65jcXVZDx3QE13uJFQc",
    "Amount": {
      "currency": "CSP",
      "issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",

```

```
    "value": "1"
  },
  "Destination": "jDKQVYNBwcPfcztUQsjxKkAWDQgQXXm4gL",
  "Fee": "10",
  "Flags": 0,
  "Memos": [
    {
      "Memo": {
        "MemoData":
"E696B0E794A8E688B7E6B3A8E5868CE8B5A0E98081E58CBAE59D97E5B8813
B20353238373738",
        "MemoType": "737472696E67"
      }
    }
  ],
  "Sequence": 20067,
  "SigningPubKey":
"02F60A05977987D77AD4D6C188D1C51685EC05BED50EB6DC986511907C487
3828E",
  "TransactionType": "Payment",
  "TxnSignature":
"30450221009E367C3BD7CAB7B1FAED45CAB10EF3D4F07ED02B73AEAE2B
111FEB548E5A0BA02205B62B1BEB09E8D443091C182C425E412593B2AC6E1A
BC266898245BC7D077101",
  "date": 627049150,
  "hash":
"23B46F65B855B98B24D5AB94ADE8829EE03C8840C7B58C8E09870F14773949
CC",
  "inLedger": 14388884,
  "ledger_index": 14388884,
  "meta": {
    "AffectedNodes": [
      {
        "ModifiedNode": {
          "FinalFields": {
            "Account": "jU3EZPPFbKHzoUi65jcXVZDx3QE13uJFQc",
            "Balance": "29799330",
            "Flags": 0,
            "OwnerCount": 1,
            "Sequence": 20068
          },
          "LedgerEntryType": "AccountRoot",
```

```

        "LedgerIndex":
"F59FC7650C29E645E494AFE10E126FA90355EDBC3B21976E55520C3E21B67E
06",
        "PreviousFields": { "Balance": "29799340", "Sequence": 20067 },
        "PreviousTxnID":
"06211E3643FF10AE5924985145BE3DA8CEF38F7D495EF36023B75A9CFD881F
FE",
        "PreviousTxnLgrSeq": 14388832
    }
},
"TransactionIndex": 0,
"TransactionResult": "tesSUCCESS",
"delivered_amount": {
    "currency": "CSP",
    "issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",
    "value": "1"
},
},
"validated": true
},
"msg": "获取交易详情成功",
"isActive": true
}

```

结果参数:

参数		类型	说明
Account		String	发起交易地址
Amount		Object	
	currency	String	币种名称
	issuer	String	发行方
	value	String	数量
Destination		String	交易对家
Fee		String	gas 费
Flags		Number	买卖标记
Memos		Array	备注
Sequence		Number	序列号
SigningPubKey		String	公共签名
TransactionType		String	转账类型
date		Number	时间
hash		String	转账 hash
inLedger		Numer	账本序号
ledger_index		Number	账本序号
TxnSignature		String	转账签名

## 3 数据接口(JccdexInfo API)

### 3.1 使用示例

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 获取指定币种 24 小时的行情数据

```
info.requestTicker(base, counter, callBack);
```

参数:

base: 基准货币

counter: 目标货币

结果如下:

```
{
  "code": "0",
  "data": [
    1573703178,
    "0.0056000000000015",
    1.8181633060612183,
    "0.00564996696339",
    "0.0054599999999997",
    565422.5980797218,
    101822158.77275103,
    166
  ],
  "msg": "获取市场信息成功",
  "success": true
}
```

结果参数:

参数		类型	说明
data		Array	
	data[0]	Timestamp	当前时间
	data[1]	String	当前价
	data[2]	Number	日涨跌(%)
	data[3]	String	最高价
	data[4]	String	最低价
	data[5]	Number	日交易额
	data[6]	Number	日成交量
	data[7]	Number	成交笔数

### 3.3 获取所有币种 24 小时的行情数据

```
info.requestAllTickers(callBack);
```

结果如下:

```
{
  "code": "0",
  "data": {
    "ECP-SWT": [1573711152, 0.45],
    "JMOAC-SWT": [
      1573711152,
      "476",
      1.4925373134324422,
      "476.000000000000022",
      "469",
      2685353.026836,
      5662.984673403357,
      17
    ]
  },
  "msg": "获取市场信息成功",
  "success": true
}
```

结果参数:

参数		类型	说明
	Array[0]	Timestamp	当前时间
	Array[1]	String	当前价
	Array[2]	Number	日涨跌(%)
	Array[3]	String	最高价

	Array[4]	String	最低价
	Array[5]	Number	日交易额
	Array[6]	Number	日成交量
	Array[7]	Number	成交笔数

### 3.4 获取市场深度

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

参数:

base: 基准货币

counter: 目标货币

type: 类型 normal/more, normal 数据长度为 5, more 数据长度为 50

结果如下:

```
{
  "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": "获取市场信息成功",
  "success": true
}
```

结果参数:

参数	类型	说明
data	Object	



	base		String	基准货币
	counter		String	目标货币(目标货币.发行方)
	asks		Array	卖出挂单
		price	String	价格
		amount	String	数量
		total	String	到当前价的总量
		type	String	类型 sell:卖 buy:买
	bids		Array	买入挂单
		price	String	价格
		amount	String	数量
		total	String	到当前价的总量
		type	String	类型 sell:卖 buy:买

### 3.5 获取 K 线数据

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

参数:

base: 基准货币

counter: 目标货币

type: 类型 minute、5minute、15minute、30minute、hour、4hour、day、week、month

结果如下:

```
{
  "code": "0",
  "data": [
    [
      1570262400000,
      "0.002971027566921",
      "0.00296",
      "0.002958536564484",
      "0.0030400000000002",
      79113.63096444756,
      26375594.9900000002,
      42
    ]
  ],
  "msg": "获取市场信息成功",
  "success": true
}
```

```
}
```

结果参数:

参数		类型	说明
data		Array	
	Array[0]	Timestamp	挂单时间
	Array[1]	String	开盘价
	Array[2]	String	收盘价
	Array[3]	String	最低价
	Array[4]	String	最高价
	Array[5]	Number	交易额
	Array[6]	Number	交易数量
	Array[7]	Number	成交笔数

### 3.6 获取分时数据(获取历史成交)

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

参数:

base:基准货币

counter:目标货币

type: 类型 all、more、newest, 如果是 newest, 则需要 time

time: 时间戳

结果如下:

```
{
  "code": "0",
  "data": [
    [
      "18667.894616639613",
      "3357598.8601470003",
      "0.005559894256047",
      1573716050000,
      1,
      1
    ]
  ],
  "msg": "获取市场信息成功",
  "success": true
}
```

结果参数:

参数		类型	说明
data		Array	
	Array[0]	String	交易额

	Array[1]	String	成交量
	Array[2]	String	价格
	Array[3]	Timestamp	时间
	Array[4]	Number	涨跌(0:涨, 1:跌)
	Array[5]	Number	撮合标志(1:非撮合, 3:三方撮合)

### 3.7 获取币种间汇率

`info.requestTickerFromCMC(token, currency, callBack)`

参数:

Token: 支持 eth、btc

currency: 支持 cny、rub

结果如下:

```
{
  "data": {
    "quotes": { "CNY": { "price": 1304.4 } },
    "last_updated": 1570893901.299,
    "id": 3,
    "symbol": "ETH"
  }
}
```

结果参数:

price: CNY 与 ETH 之间的兑换价格(即 1ETH = 1304.4CNY)

## 4 配置接口(JccConfig API)

### 4.1 使用示例

Create JccdexUrl:

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

Create JccConfig:

```
JccConfig config = JccConfig.getInstance();
config.setmBaseUrl(jccUrl);
```

### 4.2 获取配置数据

```
config.requestConfig(callBack);
```

返回参数: 所有服务器配置信息, 参数说明如下:

exHosts: 交易服务器列表

infoHosts: 信息服务器列表

cfgHosts: 配置服务器列表

scanHosts: 浏览器服务器列表

说明:

所有 API 请求对应的 host 都是基于 exHosts、infoHosts 或 cfgHosts

交易接口请调用 exHosts 列表中服务器

信息接口请调用 infoHosts 列表中服务器

配置接口请调用 cfgHosts 列表服务器

浏览器接口请调用 scanHosts 列表服务器

## 5 浏览器接口(JccExplore API)

### 5.1 使用示例

Create JccdexUrl:

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

Create JccExplore:

```
JccExplore explore = JccExplore.getInstance();
```

```
explore.setmBaseUrl(jccUrl);
```

### 5.2 获取钱包余额

```
explore.requestBalance(uuid, address, callBack);
```

参数:

uuid: 随机生成字符串即可, 下面接口相同, 不在赘述

address: 井通钱包地址

结果如下:

```
{
  "code": "0",
  "data": {
    "_id": "jfyiDN3XrbdPuAzWSwnx49DNsdkX6jqz12",
    "SWTC": { "value": "169.979881", "frozen": "42" },
    "feeflag": 0
  },
  "msg": "成功"
}
```

结果参数:

参数		类型	说明
code		String	0 表示成功(下同, 不在重复描述)
msg		String	返回状态信息(下同, 不在重复描述)
data		Object	
_id		String	查询钱包地址
SWTC		Object	币种名称(后面可能带有发行方)
	value	String	余额(包含冻结)
	frozen	String	冻结金额

### 5.3 根据 hash 获取详细交易信息

```
explore.requestTransDetails(uuid, hash, callBack);
```

参数:

hash: 交易 hash

结果如下:

```
{
  "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": "jJBpPyFBrsevf1udWTEEn3tjiHpF2Jw6ui",
    "amount": {
      "currency": "VCC",
      "issuer": "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",
```

```

        "value": "5"
    },
    "memos": [ { "Memo": { "MemoData": "00" } } ],
    "past": 121
},
    "msg": "成功"
}

```

结果参数:

data		Object	
	_id	String	交易 hash
	hashType	Number	Hash 类型(1:区块 hash,2:交易 hash)
	upperHash	String	所属区块 hash
	block	Number	区块高度
	time	Number	交易时间(time*1000+946684800 = unix 的时间戳)
	type	String	交易类型(Payment, OfferCreate, OfferCancel)
	account	String	交易发起放钱包地址
	seq	Number	交易发起方钱包所有交易的序号
	fee	Number	交易 gas 费
	succ	String	交易结果, tesSUCCESS 表示成功
	dest	String	交易对家
	amount	Obiect	转账数据
	currency	String	币种
	issuer	String	发行方
	value	String	数量
	memos	Array	备注
	past	Number	交易距现在的秒数

当 type 不同时, 返回字段可能不同

1). 当 type=Payment 时

dest: 转账目的钱包地址, 34 位字符串

amount: 转账币种和数量, json 类型, 如: {"currency": "CNY", "issuer": "\*\*\*\*", "value": "100" }

memos: 转账备注, json 数组类型, 如: [ { "Memo": { "MemoData": "7B227472616E735F74797065223A226C6F67696E5F726577617264227D", "MemoType": "737472696E67" } } ], 取 memos[0]即可, 客户端需要转码显示 (2 位分割, ascII 转换即可)

2). 当 type=OfferCreate 时

flag: 买/卖, 整型 (1:买; 2:卖; 0:未知)

takerGets: 挂单付出币种和数量, json 对象, 格式如:  
{"currency":"CNY","issuer":"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or","value":  
"100" }

takerPays: 挂单得到币种和数量, json 对象, 格式如上

realGets: 实际挂单付出币种和数量 (即扣除立即成交之后形成 Offer 的那部分), json 对象, 格式如上 (若挂单立即全部成交则该字段不存在)

realPays: 实际挂单得到币种和数量 (即扣除立即成交之后形成 Offer 的那部分), json 对象, 格式如上 (若挂单立即全部成交则该字段不存在)

matchGets: 实际成交付出币种和数量, json 对象, 格式如上 (若没有实际成交则该字段不存在)

matchPays: 实际成交得到币种和数量, json 对象, 格式如上 (若没有实际成交则该字段不存在)

matchFlag: 撮合标志, 整型 (若没有撮合, 则该字段不存在; 数字: 表示多方撮合, 比如 3 表示三方撮合)

affectedNodes: 挂单立即成交部分 (以被动成交钱包的角度), json 数组对象, 格式如: [ { "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" } } }, ..... ], 根据该数组分析出 matchGets、matchPays 以及 matchFlag。

a). account: 被动成交的钱包地址, 34 位字符串

b). seq: 该被动成交的挂单的序号, 整型

c). flag: 该被动成交的挂单的性质 (买/卖), 整型 (1:买; 2:卖; 0:未知)

d). previous: 被动成交前的交易对币种和数量, json 对象 (该字段可能没有, 若没有该字段, 表示这个被动成交记录是撤消自己的反向挂单, 这种情况在自己新的挂单会吃掉自己以前的反向挂单时会发生, 就是说不允许自己吃掉自己的挂单, 一旦要出现这种情况时, 会先把自己以前的反向挂单撤消, 然后再把新单挂上去)

e). final: 被动成交后的数量, json 对象

3). type=OfferCancel 时

offerSeq: 取消的那个挂单的序号, 整型

takerGets: 取消的那个挂单付出币种和数量, json 对象, 格式如:  
{"currency":"CNY","issuer":"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or","value":  
"100" }

takerPays: 取消的那个挂单得到的币种和数量

另: 当查询 hash 为区块 hash 时,  
结果如下:

```
{  
  "code": "0",  
  "data": {  
    "info": {
```

```

    "_id": "****",
    "hashType": 1,
    "upperHash": "",
    "block": 8488996,
    "time": 568047820,
    "past": "****",
    "transNum": 1,
    "parentHash": "****",
    "totalCoins": "5999999999999999460713"
  },
  "list": [
    {
      "_id": "****",
      "index": 0,
      "type": "OfferCreate",
      "account": "****",
      "seq": 634,
      "fee": 0.01,
      "succ": "tesSUCCESS"
    }
  ],
  "count": 16
},
"msg": ""
}

```

参数		类型	说明
info		Object	区块信息
	_id	String	区块 hash
	hashType	Number	hash 类型
	upperHash	String	所属区块 hash, 为""
	block	Number	区块高度
	time	Number	区块关闭时间
	past	String	区块距现在过去的秒数
	transNum	Number	区块中交易数量



	parentHash	String	上一区块 hash
	totalCoins	String	SWTC 总量, 客户端除以 1000000 后所得即是真实的 SWTC 总量
list		Array	区块中的交易列表
	_id	String	交易 hash
	index	Number	区块内编号
	type	String	交易类型
	account	String	发起方钱包地址
	seq	Number	发起方钱包交易中的序号
	fee	Number	交易 gas 费
	succ	String	交易结果
count		Number	区块中的交易总数

## 5.4 获取指定钱包的历史交易

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

参数:

page: 页数(从 0 开始)

size: 每页条数(10/20/50/100,缺省 20)

begin: 开始日期(格式:2019-10-11)

end: 结束日期

type: 交易类型(多个类型以逗号分隔, 可以不传值, 不传值表示查询所有类型, 可能的类型有: Send、Receive)

currency: 币种(可以不传值, 不传值表示币种不作为查询条件。在 t=Send 或 Receive 时, 传值必须 2<=长度<=8, 如 JJCC)

address: 钱包地址  
结果如下:

```
{
  "code": "0",
  "data": {
    "list": [
      {
        "type": "Receive",
        "time": 627017440,
        "hash":
"C173D26816EC647F521441785C0B1684060A93A5142A02913468CABA93888E8
B",
        "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": "成功"
}
```

结果参数:

参数	类型	说明
type	String	交易类型(Send: 支付 Receive: 收到)
time	Number	交易时间
hash	String	交易 hash
block	Number	区块高度
fee	String	交易 gas 费
seq	Number	交易序号
account	String	交易对方钱包地址
memos	Array	备注
past	Number	交易距离查询时过去的

		秒数
amount	Object	
currency	String	币种
issuer	String	发行方
value	String	数量

## 5.5 获取钱包每天/月/年支付或收到的币种笔数和数量

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

参数:

address: 钱包地址

dateType: 查询日期类型(日=2, 月=3, 年=4)

begin: 表示查询的具体某一天/月/年的日期 若 dt=2, 则必须形如 2019-07-01, 若 dt=3, 则必须形如 2019-07, 若 dt=4, 则必须形如 2019 (该参数必须传值)

end: 格式要求同 b 参数。若指定 e 的值, 则累加统计 b~e 时间段范围内相同 token 的值, 另外 e 的值必须大于 b 的值。对于 dt=2 或 3 时, b 和 e 之间天数之差不能超过一年的时间; dt=4 时, b 和 e 之间的间隔没有限制

type: 交易类型, Send/Receive

token: 该参数由 token 名称和发行方中间用下划线“\_”连接组成, 其中对于 SWTC 由于没有发行方所以 c = SWTC\_。

结果如下:

```
{
  "code": "0",
  "data": [
    {
      "wallet": "jKBCwv4EcyvYtD4PafP17PLpnnZ16szQsC",
      "token": "SWTC_",
      "type": "Send",
      "num": 1,
      "amount": 100
    }
  ],
  "msg": "成功"
}
```

结果参数:

参数	类型	说明
wallet	String	钱包地址
token	String	币种
type	String	类型(Send/Receive)
num	Number	

amount	Number	数量
--------	--------	----

## 5.6 查询某个 token 在某个时间段内的交易 hash 信息

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

参数:

page: 查询页数, 从 0 开始  
size: 每页条数, 缺省 20, 10/20/50/100 四种选择  
begin: 查询的开始日期, 形如 2019-10-13  
end: 查询的结束日期, 形如 2019-10-13  
type: 交易类型, Payment  
currency: token 名称

结果如下:

```
{
  "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": "成功"
}
```

结果参数:

参数	类型	说明
<u>_id</u>	String	交易 hash

hashType	Number	Hash 类型
time	Number	交易时间
index	Number	区块内编号
type	String	交易类型
account	String	交易发起方钱包
seq	Number	交易发起方钱包的序列号
fee	Number	交易 gas 费(swtc)
dest	String	交易对家
succ	String	交易状态, tesSUCCESS 表示成功
amount	Object	
currency	String	币种名称
issuer	String	发行方
value	Strign	数量

### 5.7 获取所有 tokens 列表

```
curl stats.jccdex.cn/sum/all/:uuid
curl stats.jccdex.cn/sum/all/:uuid?t=C
```

参数:  
t: 查询 token 条件, 可指定 token 包含信息  
结果如下:

```
{
  "code": "0",
  "data": [
    {
      "J": [
        "JSLASH_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or"
      ]
    }
  ],
  "msg": "成功"
}
```

参数	类型	说明
	Array	token 名称首字母
	String	token 名称+"_"+发行方

## 5.8 获取某种 tokens 的 100 排名列表

curl

```
stats.jccdex.cn/sum/list/:uuid?t=JSLASH_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or&p=4
```

参数:

t: 某种 token, 必选参数

p: 页数, 从 0 开始, 最多显示某 token 前 100 名

s: 每页显示条数, 默认 20, 可选 10/20/50/100

b: 开始日期, 用于查询历史排名

e: 结束日期

结果如下:

```
{
  "code": "0",
  "data": [
    {
      "tokens": "JSLASH_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or"
    },
    {
      "holders": 0
    },
    {
      "totalsupply": 199999412
    },
    {
      "circulation": 136375233.93999976
    },
    [
      {
        "address": "jsk45ksJZUB7durZrLt5e86Eu2gtiXNRN4",
        "time": 627096607,
        "amount": "63618238.06000023"
      }
    ],
    [
      "jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd",
      "j4TVCSesxU2s6tjYpsDoH4Pc81c4WMvNHPo",
      "jUq1KR6YHHAWxTb8MApaeCLfkBS6a8Y4nX",
      "jnoqBzpAdWixYzwcwFj8Q3SDBrR6QKRgK",
      "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or",
      "jsk45ksJZUB7durZrLt5e86Eu2gtiXNRN4",
      "jG9ntUTuBKqDURPUqbGYZRuRDVzPY6bpxL",
      "jMCJrXRmycsT5tsVuge7Y65v9MrQi9r11E",

```

```

        "japp9xxt2VHpRwHsoa76GWOqj1VdsjcZQJ",
        "jDu7umDxKxeaHoj7eNdUn8YsGWTHZSuEGL",
        "jQs5cAcZrKmyWSQgkmUtXsdeFMzwSYcBA4",
        "jhAvDrBjDV4HFRmRKZ3ZnL77Sg4PCezAwE"
    ],
    {
        "flag": 1
    },
    {
        "issueDate": 594797250
    }
],
    "msg": "成功"
}

```

结果参数:

参数		类型	说明
tokens		String	token 名称
totalsupply		Number	币种发行量
circulation		Number	流通量
		Array	排名数组
	address	String	钱包地址
	time	Number	时间戳，以天为单位
	amount	String	持有数量
		Array	储备钱包

## 5.9 获取指定钱包指定币种的排名

```

curl
stats.jccdex.cn/sum/self/:uuid?t=CSP_jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or&
w=jKectkVmsURkNbrJ7EEd1DQPngPCmDn1sQ

```

参数:

t: token

w: 钱包地址

结果如下:

```

{
    "code": "0",
    "data": 422415,
    "msg": "成功"
}

```

结果参数:

参数	类型	说明
data	Number	个人排行, 未持有则返回空

## 5.10 获取银关地址发行过的 token

get 方法:

[https://stats.jccdex.cn/wallet/fingate\\_tokenlist/:uuid?w=jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd](https://stats.jccdex.cn/wallet/fingate_tokenlist/:uuid?w=jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd)

参数:

w: 银关钱包地址

结果如下:

```
{
  "code": "0",
  "data": {
    "jhyXdawKCnhCVLkgqzTnNRrdMvGt1yqzwd": [
      { "currency": "YYT", "issuer":
"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or" },
      { "currency": "HPT", "issuer":
"jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or" }
    ]
  },
  "msg": "成功"
}
```

结果参数:

参数	类型	说明
currency	String	token 名称
issuer	String	发行方