井通科技

井通标准服务接口

v2.1.0

井通科技
JINGTUM TECH

1 账号类接口

1.1 创建账号

接口：/v2/wallet/new，GET方法

例子：http://localhost/v2/wallet/new

结果如下：

```
{
  "success": true,
  "status_code": "0",
  "wallet": {
    "secret": "ssD7KN7xnoznunG4H3TF6weF9kWb4",
    "address": "jN22GEKA4kha8xGktuo1K4MpMhx6RRahQS"
  }
}
```

返回的结果信息：

参数		类型	说明
success		Boolean	请求结果
wallet		Object	井通钱包
	secret	String	井通钱包私钥
	address	String	井通钱包地址

1.2 获得账号余额

接口：/v2/accounts/{:address}/balances，GET方法

接口参数：

参数	类型	说明

address	String	井通钱包地址

可选接口参数：

参数	类型	说明
currency	String	指定返回对应货币的余额，货币区分大小写
issuer	String	指定返回对应银关发行的货币

例子：http://localhost/v2/accounts/jQNdYXxgNHY49oxDL8mrjr7J6k7tdNy1kM/balances

结果如下：


```
{
  "success": true,
  "status_code": "0",
  "balances": [
    {
      "value": "97.63958",
      "currency": "SWT",
      "issuer": "",
      "freezed": "35"
    },
    {
      "value": "99",
      "currency": "AAA",
      "issuer": "jMhLAPaNFo288PNo5HMC37kg6ULjJg8vPf",
      "freezed": "0"
    },
    {
      "value": "0",
      "currency": "CNY",
```

```
      "issuer": "jMhLAPaNFo288PNo5HMC37kg6ULjJg8vPf",
      "freezed": "0"
    }
  ],
  "sequence": 164,
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
balances	Array	余额数组
value	String	余额
currency	String	货币名称，三个字母或20字节的货币
issuer	String	货币发行方
freezed	String	冻结的金额
sequence	Integer	当前交易序列号（用于本地签名）

2 支付接口

2.1 支付请求

接口：/v2/accounts/{:source_address}/payments，POST方法

接口参数：

参数	类型	说明
source_address	String	支付方的井通地址

提交参数详情：

参数		类型	说明
secret		String	支付方的钱包私钥
client_id		String	此次请求的交易单号，交易单需要唯一
payment		Object	支付对象
	source	String	发起账号
	destination	String	目标账号
	amount	Object	支付金额
	choice	String	支付选择的key，可选
	memos	Array	支付的备注，String数组，可选

例子：http://localhost/v2/accounts/jQNdYXxgNHY49oxDL8mrjr7J6k7tdNy1kM/payments

提交的数据：

```
{
  "secret": "snxVJXMkURjrscL7gfwfWcywYzPkL",
  "client_id": "109",
  "payment": {
    "source": "jQNdYXxgNHY49oxDL8mrjr7J6k7tdNy1kM",
    "destination": "jD2RbZEpBG3T6iWDPyPiNBvpU8sAhRYbpZ",
    "amount": {
      "value": "1.00",
      "currency": "AAA",
      "issuer": "jMhLAPaNFo288PNo5HMC37kg6ULjJg8vPf"
    },
    "choice": "f53b09afcf9e1758a7b647f2f738c86426cabfc1",
   "memos": ["hello world", "hello payment"]
  }
```

```
}
```

结果：

```
{
  "success": true,
  "status_code": "0",
  "client_id": "109",
  "hash": "0FE129880597A2681A7CDEA6098C19DA0EB97787FB09F0C1966AAD640D991BB5",
  "result": "tesSUCCESS",
  "fee": "0.000012"
}
```

结果返回参数：

参数	类型	说明
success	Boolean	请求结果
client_id	String	支付交易单号
hash	String	支付交易Hash
result	String	支付交易的服务器结果，tesSUCCESS表示成功，其他类型详见错误信息
fee	String	交易费用，井通计价

2.2 获得支付信息

接口：/v2/accounts/{:address}/payments/{:id}，GET方法

接口参数：

参数	类型	说明
address	String	支付用户的井通地址

id	String	支付交易的hash或资源号

例子：

http://localhost/v2/accounts/jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834/payments/D0BF25B015A6BF94F0645FBBC17B599E546B5DBECD5BBC23C1CB64F83C80A76B

结果：


```
{
  "success": true,
  "status_code": "0",
  "date": 1427828830,
  "hash": "D0BF25B015A6BF94F0645FBBC17B599E546B5DBECD5BBC23C1CB64F83C80A76B",
  "type": "sent",
  "fee": "0.000012",
  "result": "tesSUCCESS",
  "memos": ["hello world", "hello payment"]
  "counterparty": "jsPVrt5C97gbNXx9S1DzBZvXETkWbmHgFQ",
  "amount": {
    "value": "1",
    "currency": "CNY",
    "issuer": "jsPVrt5C97gbNXx9S1DzBZvXETkWbmHgFQ"
  },
  "effects": [ ]
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
date	Integer	支付时间，UNIXTIME时间

hash	String	支付hash
type	String	支付类型，sent或received
fee	String	支付费用
result	String	支付的服务器结果
memos	Array	支付的备注，String数组
counterparty	String	交易对家
amount	Object	交易金额
effects	Array	支付的效果，详见下面交易记录效果

2.3 获得支付历史

接口：/v2/accounts/{:address}/payments，GET方法

接口参数：

参数	类型	说明
address	String	支付相关的井通地址

可选接口参数：

参数		类型	说明
results_per_page		Integer	返回的每页数据量，默认每页10项
page		Integer	页码，默认从第一页开始
marker		Object	交易记录标记，表示从当前记录继续向下查找。(注：marker优先级大于page。即当有marker和page时，表示从marker处向下查找到第page页)
	ledger	Integer	账本号
	seq	Integer	上一次查询返回的最后一条交易号
currency		String	指定返回对应货币的支付历史，货币区分大小写

例子：

http://localhost/v2/accounts/jnbsmZpkBaGGpPip2A3HujzzWcQvURNGC4/payments?results_per_page=1&page=4

结果：


```
{
  "success": true,
  "status_code": "0",
  "marker": {
      "ledger": 7657903,
      "seq": 0,
  },
  "payments": [
    {
      "date": 1430271890,
      "hash": "9600FFB5966458EB32D6C5344D3F720A59A210AA6ECF92F68D2D7C077C242375",
      "type": "sent",
      "fee": "0.000012",
      "result": "tesSUCCESS",
      "memos": ["hello world", "hello payment"]
      "counterparty": "jP659KPCEa76rf7kUCwoTk9m5ZjGiRXco7",
      "amount": {
        "value": "0.01",
        "currency": "8000000001201504081454405641668733216531",
        "issuer": "jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS"
      },
      "effects": [ ]
    }
```

```
  ]
}
```

返回的结果是下一页的数据，返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
marker	Object	交易记录标记，是当前交易记录结束标记，也是下一次查找的开始标记；当marker存在时，表示后续还有记录。
payments	Array	支付历史，同交易记录中的信息

3 挂单类接口

3.1 提交挂单

接口：/v2/accounts/{:address}/orders，POST方法

接口参数：

参数	类型	说明
address	String	用户的井通地址

提交参数详情：

参数		类型	说明
secret		String	用户的钱包私钥
order		Object	提交的挂单
	type	String	挂单的类型，sell或buy
	pair	String	交易的货币对(支付SWT不需要加issuer；货币区分大小写)

	amount	String/Integer	挂单的数量
	price	String/Integer	挂单的价格

POST需要提交的参数格式如下：

```
{
  "secret": "snUaJxp2k4WFt5LCCtEx2zjThQhpT",
  "order": {
    "type": "sell",
    "pair": "SWT/CNY:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
    "amount": "100.00",
    "price": "0.0124"
  }
}
```

例子：http://localhost/v2/accounts/jf96oSdxU7kwfCHF2sjm9GmcvhFBcfN8Py/orders

结果：

```
{
  "success": true,
  "status_code": "0",
  "hash": "71AE74B03DE3B9A06C559AD4D173A362D96B7D2A5AA35F56B9EF21543D627F34",
  "result": "tesSUCCESS",
  "fee": "0.000012",
  "sequence": 99
}
```

返回的结果信息：

参数	类型	说明

success	Boolean	请求结果
hash	String	交易hash
result	String	交易结果
fee	String	交易费用，井通计价
sequence	Integer	交易单子序号

3.2 取消挂单

接口：/v2/accounts/**{:address}**/orders/**{:order}**，**DELETE**方法

DELETE方法请求时需设置Content-Length消息头

接口参数：

参数	类型	说明
address	String	挂单方的井通地址
order	String	订单的序号

提交参数详情：

参数	类型	说明
secret	String	井通钱包私钥

DELETE需要提交的参数格式如下：

```
{
  "secret": "snUaJxp2k4WFt5LCCtEx2zjThQhpT"
}
```

例子：http://localhost/v2/accounts/jJHPjyMfaKDVfCjuKdn8zFmYZVYXg7akeJ/orders/84

结果：

```
{
```

```
  "success": true,
  "status_code": "0",
  "hash": "31E59895D920F05A913D24808D1EE0392562B1DE3053FF72C15758476EA8A6C0",
  "fee": "0.000012",
  "sequence": 86,
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
hash	String	交易hash
fee	String	交易费用，井通计价
sequence	Integer	操作订单序列号

3.3 获取用户挂单

接口：/v2/accounts/**{:address}**/orders，**GET**方法

接口参数：

参数	类型	说明
address	String	挂单方的井通地址

可选接口参数：

参数	类型	说明
results_per_page	Integer	返回的每页数据量，默认每页10项
page	Integer	页码，默认从第一页开始
marker	Integer	账本存储索引，是交易记录的另一种标记，可从当前记录继续向下查找。(注：marker优先级大于page。即当有marker和page时，表示从marker处向下查找到第page页)

注：results_per_page与page相乘，最小值是10。

例子：http://localhost /v2/accounts/jf96oSdxU7kwfCHF2sjm9GmcvhFBcfN8Py/orders

结果：


```
{
  "success": true,
  "status_code": "0",
  "marker": "32F9FB0BC6449DE935F085E681560F59630A15B00E24FB1E97ABA60D3C3AF3A9",
  "orders": [
    {
      "type": "buy",
      "pair": "800000000000000000000000A95EFD7EC3101635:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
      "amount": "119815.98",
      "price": "0.0124",
      "sequence": 4
    },
    {
      "type": "buy",
      "pair": "800000000000000000000000A95EFD7EC3101635:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
      "amount": "10000.00",
      "price": "0.0100",
      "sequence": 33
    },
    {
      "type": "buy",
      "pair": "800000000000000000000000A95EFD7EC3101635:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
      "amount": "10000.00",
```

```
      "price": "0.0100",
      "sequence": 34
    },
    {
      "type": "sell",
      "pair": "800000000000000000000000A95EFD7EC3101635:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
      "amount": "98300",
      "price": "0.0100",
      "sequence": 242
    },
    {
      "type": "sell",
      "pair": "800000000000000000000000A95EFD7EC3101635:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
      "amount": "16400.00",
      "price": "0.0100",
      "sequence": 244
    },
    {
      "type": "sell",
      "pair": "800000000000000000000000A95EFD7EC3101635:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
      "amount": "18500.00",
      "price": "0.0100",
      "sequence": 246
    }
  ]
}
```

返回的结果信息：

参数		类型	说明
success		Boolean	请求结果
marker		String	当前交易记录标记
order		Object	单子信息
	type	String	单子类型，sell或buy
	pair	String	交易的货币对，货币区分大小写
	amount	String	挂单的数量
	price	String	挂单的价格
	sequence	Integer	交易序列号

3.4 获取挂单信息

接口：/v2/accounts/**{:address}**/orders/**{:hash}**，**GET**方法

通用接口参数：

参数	类型	说明
address	String	挂单方的井通地址
hash	String	挂单交易哈希号

例子：
http://localhost/v2/accounts/jf96oSdxU7kwfCHF2sjm9GmcvhFBcfN8Py/orders/6ECFE85BDB388AFCD2AD6BC766F5B47C26DCFCF3D05ED0226AFB849EA799577F

结果：


```
{
  "success": true,
  "status_code": "0",
```

```
  "hash": "6ECFE85BDB388AFCD2AD6BC766F5B47C26DCFCF3D05ED0226AFB849EA799577F",
  "fee": "0.000012",
  "action": "sell",
  "order": {
    "account": "jf96oSdxU7kwfCHF2sjm9GmcvhFBcfN8Py",
    "pair": "SWT/CNY:janxMdrWE2SUzTqRUtfycH4UGewMMeHa9f",
    "amount": "1.00",
    "price": "10.0000",
    "type": "buy",
    "sequence": 761
  }
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
hash	String	交易Hash
fee	String	交易费用，井通计价
action	String	交易的动作类型
order	Object	交易单子信息
account	String	交易帐号
pair	String	交易的货币对
amount	String	挂单的数量
price	String	挂单的价格
type	String	交易类型，sell或buy

	sequence	String	交易序列号

3.5 获得货币对的挂单列表

接口：/v2/order_book/**{:base}**/**{:counter}**，**GET**方法

通用接口参数：

参数	类型	说明
base	String	基准货币（currency+issuer），如果货币是基础货币SWT，issuer不填写，货币区分大小写
counter	String	目标货币（currency+issuer），如果货币是基础货币SWT，issuer不填写，货币区分大小写

可选接口参数：

参数	类型	说明
results_per_page	Integer	返回的每页数据量，默认每页买卖单各10项，最大300
page	Integer	页码，默认从第一页开始

例子：
http://localhost/v2/order_book/CNY+jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/USD+jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS

结果：


```
{
  "success": true,
  "status_code": "0",
  "pair": "CNY:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/USD:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
  "bids": [
    {
      "price": "0.66",
```

```
    "funded": 100,
    "order_maker": "jhBGoRtU9MxLeipYyvtbbZxk9cLD5c35MP",
    "sequence": 448,
    "passive": false,
    "sell": false
  }
],
"asks": [
  {
    "price": "0.82333",
    "funded": 158000,
    "order_maker": "jU7uJxzJttKdyeNenaN7KCyfcb9gy99KKc",
    "sequence": 960,
    "passive": false,
    "sell": true
  },
  {
    "price": "0.88",
    "funded": 81.84,
    "order_maker": "js46SK8GtxSeGRR6hszxozFxftEnwEK8my",
    "sequence": 12,
    "passive": false,
    "sell": true
  },
  {
    "price": "1",
    "funded": 1,
    "order_maker": "j3AxmnN7ZE4EUTiD7CTWRdEYsQk4tEpMVQ",
```

```
      "sequence": 37,
      "passive": false,
      "sell": true
    },
    {
      "price": "2",
      "funded": 1,
      "order_maker": "j3WBq56fYBaJGRhex3RJGTVFFLfhZeSkR7",
      "sequence": 10,
      "passive": false,
      "sell": true
    }
  ]
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	调用结果
pair	String	挂单货币对
bids/asks	Array	买入单
price	Object	该档的价格
funded	Integer	实际中用户可以成交的金额
order_maker	String	挂单用户
sequence	String	交易序号
passive	Boolean	交易是否是被动交易

3.6 获得货币对的买单列表

接口：/v2/order_book/bids/**{:base}**/**{:counter}**，**GET**方法

通用接口参数：

参数	类型	说明
base	String	基准货币（currency+issuer），如果货币是基础货币SWT，issuer不填写；货币区分大小写
counter	String	目标货币（currency+issuer），如果货币是基础货币SWT，issuer不填写，货币区分大小写

可选接口参数：

参数	类型	说明
results_per_page	Integer	返回的每页数据量，默认每页10项，最大300
page	Integer	页码，默认从第一页开始

例子：
http://localhost/v2/order_book/bids/SWT/CNY+jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS?results_per_page=3&page=2

结果：


```
{
  "success": true,
  "status_code": "0",
  "pair": "SWT/CNY:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
  "bids": [
    {
      "price": "0.97",
      "order_maker": "jhBGoRtU9MxLeipYyvtbbZxk9cLD5c35MP",
      "sequence": 90,
      "passive": false,
```

```
      "sell": false,
      "funded": 0.32
    },
    {
      "price": "0.96999",
      "order_maker": "jhBGoRtU9MxLeipYyvtbbZxk9cLD5c35MP",
      "sequence": 48,
      "passive": false,
      "sell": false,
      "funded": 0.31
        },
    {
      "price": "0.95999",
      "order_maker": "jhBGoRtU9MxLeipYyvtbbZxk9cLD5c35MP",
      "sequence": 448,
      "passive": false,
      "sell": false,
      "funded": 8.999
    }
  ]
}
```


返回的结果信息：同3.5。

3.7 获得货币对的卖单列表

接口：/v2/order_book/asks/**{:base}**/**{:counter}**，**GET**方法

通用接口参数：

参数	类型	说明

base	String	基准货币（currency+issuer），如果货币是基础货币SWT，issuer不填写；货币区分大小写
counter	String	目标货币（currency+issuer），如果货币是基础货币SWT，issuer不填写；货币区分大小写

可选接口参数：

参数	类型	说明
results_per_page	Integer	返回的每页数据量，默认每页10项，最大300
page	Integer	页码，默认从第一页开始

例子：
http://localhost/v2/order_book/asks/SWT/CNY+jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/USD?results_per_page=2&page=3

结果：


```
{
  "success": true,
  "status_code": "0",
  "pair": "SWT/CNY:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
  "asks": [
    {
      "price": "0.82333",
      "order_maker": "jU7uJxzJttKdyeNenaN7KCyfcb9gy99KKc",
      "sequence": 960,
      "passive": false,
      "sell": true,
      "funded": 158000
    },
    {
      "price": "0.88",
```

```
      "order_maker": "js46SK8GtxSeGRR6hszxozFxftEnwEK8my",
      "sequence": 12,
      "passive": false,
      "sell": true,
      "funded": 81.84
    }
  ]
}
```


返回的结果信息：同3.5。

4 交易记录接口

4.1 查询交易信息

接口1：/v2/accounts/**{:address}**/transactions/**{:id}**，**GET**方法

接口2：/v2/transactions/**{:id}**，**GET**方法

接口参数：

参数	类型	说明
address	String	井通钱包地址
id	String	交易资源号或者交易hash

说明：查询交易记录可通过交易hash直接查询，也可通过钱包地址和hash查询，两者区别在于加钱包地址的可以判断该交易是否属于该地址。

例子：

http://localhost/v2/accounts/jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834/transactions/264ECE86AF9A830405B71C9A9FDB6807AC55525CA73E6DD1C61621F33D13F749

结果：


```
{
  "success": true,
  "status_code": "0",
```

```
  "transaction": {
    "date": 1427265660,
    "hash": "264ECE86AF9A830405B71C9A9FDB6807AC55525CA73E6DD1C61621F33D13F749",
    "type": "sent",
    "fee": "0.000012",
    "result": "tesSUCCESS",
    "counterparty": "j3AxmnN7ZE4EUTiD7CTWRdEYsQk4tEpMVQ",
    "amount": {
      "value": "1.3",
      "currency": "CNY",
      "issuer": "j3AxmnN7ZE4EUTiD7CTWRdEYsQk4tEpMVQ"
    },
    "ledger": 8989380,
    "effects": [ ]
  }
}
```


返回的结果信息：

参数		类型	说明
success		Boolean	请求结果
transaction		Object	具体的交易信息
	date	Integer	交易时间，UNIXTIME
	hash	Object	交易hash
	type	Object	交易类型
	fee	Object	交易费用，井通计价
	result	Object	交易结果

	counterparty	Object	交易对家
	amount	Object	交易金额
	account	String	交易发起方，也是自动补齐的接口1中的address，通过接口2查才有此字段。
	ledger	Integer	交易所在的账本号
	effects	Object	交易效果，详见下面分析

4.2 查询交易记录

接口：/v2/accounts/**{:address}**/transactions，**GET**方法

接口参数：

参数	类型	说明
address	String	支付相关的井通地址

可选接口参数：

参数		类型	说明
results_per_page		Integer	返回的每页数据量，默认每页10项
page		Integer	页码，默认从第一页开始
forward		String	按交易时间排序，固定的两个值：asc(升序)、desc(降序)
marker		Object	交易记录标记，表示从当前记录继续向下查找。(注：marker优先级大于page。即当有marker和page时，表示从marker处向下查找到第page页)
	ledger	Integer	账本号
	seq	Integer	上一次查询返回的最后一条交易号
currency		String	指定返回对应货币的交易记录，货币区分大小写

注：results_per_page与page相乘，最大值是200。

例子：

http://localhost/v2/accounts/jHZ5pAKGcpCdvU8N177eePYtMY4ZtC85o6/transactions?results_per_page=5&marker={ledger:6946767,seq:0}

结果：


```
{
  "success": true,
  "status_code": "0",
  "marker": {
    "ledger": 6946759,
    "seq": 0
    },
  "transactions": [
    {
      "date": 1464163090,
      "hash": "DCFB219EB792CAB190EFE49004D341624AA3444F99CEFD1FDABF1D2AF2716C9C",
      "type": "offereffect",
      "fee": "0.000012",
      "result": "tesSUCCESS",
      "memos": [ ],
      "effects": [
        {
          "effect": "offer_partially_funded",
          "type": "bought",
          "price": "2",
          "seq": 3,
          "pair": "830000003100002016052520320528012000000 3:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/820000003100002016052520290525012000000 3:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
          "amount": "1.00",
```

```
        "price": "2.0000",
        "counterparty": {
          "account": "jf16TmdHVXuaVNhFDEg4MVeGWiEhsCMRF4",
          "seq": 10,
          "hash": "DCFB219EB792CAB190EFE49004D341624AA3444F99CEFD1FDABF1D2AF2716C9C"
        },
        "remaining": true
      }
    ]
  },
  {
    "date": 1464160770,
    "hash": "D48B7D8B319D3EBC30CB2ED85E45A12208F199A69E51AAB2291AB94A6B79AF68",
    "type": "offernew",
    "fee": "0.000012",
    "result": "tesSUCCESS",
    "memos": [ ],
    "offertype": "buy",
    "seq": 3,
    "pair": "8300000031000020160525203205280120000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/8200000310000201605252029052501200000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
    "amount": "2.00",
    "price": "5.0000",
    "effects": [
      {
        "effect": "offer_created",
        "type": "buy",
        "pair": "8300000031000020160525203205280120000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/8200000031000020160525202905250120000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
```

```
          "amount": "5.00",
          "price": "2.0000",
          "seq": 3
        }
      ]
    },
    {
      "date": 1464158680,
      "hash": "2AE79F8EC6B71BC10F148F9DB6791B9B363C91D9D8F0B2BB45177A761998D4F3",
      "type": "offercancel",
      "fee": "0.000012",
      "result": "tesSUCCESS",
      "memos": [ ],
      "offertype": "buy",
      "pair": "8300000022000020151029201910310120000002:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/8200000310000201605252029052501200000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
      "amount": "1.00",
      "price": "2.0000",
      "seq": 2,
      "effects": [
        {
          "effect": "offer_cancelled",
          "pair": "8300000022000020151029201910310120000002:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/8200000031000020160525202905250120000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
          "amount": "1.00",
          "price": "2.0000",
          "seq": 1,
          "hash": "DB1B6CEDADAACFFEDDEC48FCD12A841F858ABACE3575B46D96E1EA05C86091E0",
          "deleted": true
```

```
      }
    ]
  },
  {
    "date": 1464161640,
    "hash": "F0DF19E6809909744DC07E5B4CC8F9B294846658912C3DBC5DAD8343316C39B8",
    "type": "received",
    "fee": "0.000012",
    "result": "tesSUCCESS",
    "memos": [ ],
    "counterparty": "jfpZh2698t7RcAVpboyfx5bDL29dGyp5Ve",
    "amount": {
      "value": "1",
      "currency": "8200000031000020160525202905250120000003",
      "issuer": "jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS"
    },
    "effects": [ ]
  },
  {
    "date": 1464160770,
    "hash": "D48B7D8B319D3EBC30CB2ED85E45A12208F199A69E51AAB2291AB94A6B79AF68",
    "type": "offernew",
    "fee": "0.000012",
    "result": "tesSUCCESS",
    "client_resource_id": "",
    "memos": [ ],
    "offertype": "buy",
    "seq": 3,
```

```
      "pair": "830000002200002015102920191031012000000 2:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/8200000310000201605252029052501200000 03:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
      "amount": "5.00",
      "price": "2.0000",
      "effects": [
        {
          "effect": "offer_created",
          "type": "buy",
          "pair": "830000002200002015102920191031012000000 2:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS/8200000031000020160525202905250120000003:jBciDE8Q3uJjf111VeiUNM775AMKHEbBLS",
          "amount": "5.00",
          "price": "2.0000",
          "seq": 3
        }
      ]
    }
  ]
}
```


返回的结果信息：

参数		类型	说明
success		Boolean	调用结果
marker		Object	交易记录标记，是当前交易记录结束标记，也是下一次查找的开始标记；当marker存在时，表示后续还有记录。
transactions		Array	具体的交易信息数组
	date	Integer	交易时间，UNIXTIME
	hash	String	交易hash
	type	String	交易类型

	fee	String	交易费用，井通计价
	result	String	交易结果
	memos	Array	交易备注
	counterparty	String	交易对家，支付交易才有
	amount	String	交易金额/挂单数量，支付交易或者挂单交易才有
	offertype	String	挂单类型，sell或者buy，挂单交易才有
	pair	String	交易的货币对，挂单交易才有
	price	String	挂单的价格，挂单交易才有
	method	String	固定的两个值：deploy（创建合约）、call（调用合约），合约交易才有
	payload	String	合约内容，创建合约才有
	destination	String	合约账号，调用合约才有
	effects	Object	交易效果，详见如下

4.3 交易记录信息

用户提交的交易类型主要有Payment、OfferCreate和OfferCancel；

交易信息存储在系统中，查询交易信息时，系统解析交易信息，将交易信息解析为主要有如下信息：

date 交易时间UNIXTIME

hash 交易hash

fee 交易费用

result 交易结果

memos 交易的备注信息

type 交易类型

type有如下几种：

1. sent，用户进行支付操作，在交易信息中包含的信息有：

字段		类型	说明
counterparty		String	支付对家，即接收方
amount		Object	
	value	String	金额
	currency	String	货币
	issuer	String	货币发行方，SWT为空字符串
effects		Array	[]，空

2. received，用户接受支付，在交易信息中包含的信息有：

字段		类型	说明
counterparty		String	支付对家，即支付方
amount		Object	
	value	String	金额
	currency	String	货币
	issuer	String	货币发行方，SWT为空字符串
effects		Array	[]，空

3. convert，用户进行兑换操作，在交易信息中包含的信息有：

字段		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为空字符串
amount		Object	兑换获得的金额
	value	String	金额
	currency	String	货币

	issuer	String	货币发行方
effects		Array	详见下面的effects解释

4. offernew，用户进行挂单操作，在交易信息中包含的信息有：

字段	类型	说明
offertype	String	挂单类型，sell或buy
seq	Integer	单子号
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，在交易信息中包含的信息有：

字段	类型	说明
type	String	挂单的类型，sell或buy
offerseq	Integer	单子号
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，在交易信息中包含的信息有：

字段	类型	说明
effects	Array	详见下面的effects解释

4.4 交易效果effects

effects是每个用户交易记录信息里面的交易效果，是个JSON数组，数组可以包含多项，每

项内容都包含效果类型effect字段，根据effect的不同里面的内容也不同：

1. offer_funded，交易实际成交；**交易提示信息建议**：交易成交，您以 XXX 价格买了/卖了 XXX 卖了/买了 XXX，价格是 XXX；其中包含的信息有：

字段		类型	说明
effect		String	offer_funded
type		String	交易类型，sell或buy
pair		String	交易的货币对
amount		String	挂单的数量
got		Object	用户获得的金额
	value	String	金额
	currency	String	货币
	issuer	String	货币发行方，SWT为空字符串
paid		Object	用户付出的金额
	value	String	金额
	currency	String	货币
	issuer	String	货币发行方，SWT为空字符串
price		String	价格，4位小数
seq		Integer	挂单序号，表示被成交的单子
deleted		Boolean	单子是否被删除了，被吃了的单子会被删除掉
counterparty		Object	对家信息
	account	String	对家账号
	seq	Integer	对家单子序号
	hash	String	对家交易hash

2. offer_partially_funded，交易部分成交；交易提示信息建议：交易部分成交，您以 XXX 价格买了/卖了 XXX 卖了/买了 XXX，价格是 XXX，剩余挂单由于金额不足被取消（可选，根据

cancelled），还剩 XXX 单子（可选，根据 remaining）；其中包含的信息有：

字段	类型	说明
effect	String	offer_partially_funded
type	String	交易类型，sold或bought
got	Object	用户获得的金额
value	String	金额
currency	String	货币
issuer	String	货币发行方，SWT为空字符串
paid	Object	用户付出的金额
value	String	金额
currency	String	货币
issuer	String	货币发行方，SWT为空字符串
seq	Integer	挂单序号，表示被部分成交的单子
cancelled	Boolean	剩余的单子是否被取消了，可选
remaining	Boolean	是否有剩余的单子，可选
pair	String	交易的货币对，remaining为true才有
amount	String	挂单的数量，remaining为true才有
price	String	挂单的价格，remaining为true才有
counterparty	Object	对家信息
account	String	对家账号
seq	Integer	对家单子序号
hash	String	对家交易hash

3. offer_cancelled，被关联交易取消单子，交易单子被取消；交易提示信息建议：由于缺少金额单子 XXX 被取消；其中包含的信息有：

字段	类型	说明

effect	String	offer_cancelled
type	String	交易类型，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
seq	Integer	被取消单子的序号
hash	String	被取消单子的hash
deleted	Boolean	单子是否被删除，取消单子为true

4. offer_created，交易单子创建；交易提示信息建议：您创建了一个买/卖单，以 XXX 交易 XXX；其中包含的信息有：

字段	类型	说明
effect	String	offer_created
type	String	交易类型，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
seq	Integer	新建的单子序号

5. offer_bought，挂单买到/卖出，成交的单子信息；交易提示信息建议：您以 XXX 价格买了/卖了 XXX 卖了/买了 XXX；其中包含的信息有：

字段		类型	说明
effect		String	offer_bought
type		String	交易类型，sell或buy
got		Object	用户获得的金额
	value	String	金额
	currency	String	货币

	issuer	String	货币发行方，SWT为空字符串
paid		Object	用户付出的金额
	value	String	金额
	currency	String	货币
	issuer	String	货币发行方，SWT为空字符串
price		String	价格，4位小数
pair		String	交易的货币对
amount		String	挂单的数量
counterparty		Object	对家信息
	account	String	对家账号
	seq	Integer	对家单子序号
	hash	String	对家交易hash

5 智能合约(暂未对外开放)

5.1 部署合约

接口：/v2/accounts/**{:address}**/contract/deploy，POST方法

接口参数：

参数	类型	说明
address	String	井通钱包地址

提交参数详情：

参数	类型	说明
secret	String	用户的钱包私钥
amount	Integer	支付金额
payload	String	智能合约代码

可选接口参数：

参数	类型	说明
params	Array	合约参数

POST需要提交的参数格式如下：


```
{
  "secret": "snUaJxp2k4WFt5LCCtEx2zjThQhpT",
  "amount": 10
  "payload": "function Init(...) a={} for k,v in ipairs({...}) do a[k]=v end b=a[1] return account
info(b) end;  function foo(...) a={} for k,v in ipairs({...}) do a[k]=v end b=a[1] return accounti
nfo(b) end"
}
```


例子：

http://localhost/v2/accounts/jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834/contract/deploy

结果：


```
{
  "success": true,
  "status_code": "0",
  "ContractState": "contract:j43X58GWHGqWqQXM6b2BNr43AG8TjsmfYb",
  "engine_result": "tesSUCCESS",
  "engine_result_code": 0,
  "engine_result_message": "The transaction was applied. Only final in a validated ledger.",
  "tx_blob": "12001E22000000000240000001C2F215B24A1202400000000614000000000989680684000000000000000A
73210330E7FC9D56BB25D6893BA3F317AE5BCF33B3291BD63DB32654A313222F7FD0207447304502210 0A009376F5DCFA0
56F452C1B0703004A1D181D8479F908E353B2086867F4E89C20220100CCB09EDFEF416D1B611456BBEF4260E130E48E6AC
719EE8A3A89FE65AFE127FC066756E6374696F6E20496E6974282E2E2E2920613D7B7D20666F72206B2C7620696E206970
61697273287B2E2E2E7D2920646F20615B6B5D3D7620656E6420623D615B315D2072657475726E206163636F756E74696E
666F28622920656E643B202066756E6374696F6E20666F6F282E2E2E2920613D7B7D20666F72206B2C7620696E20697061
697273287B2E2E2E7D2920646F20615B6B5D3D7620656E6420623D615B315D2072657475726E206163636F756E74696E66
6F28622920656E648114B5F762798A53D543A014CAF8B297CFF8F2F937E8",
```

```
  "tx_json": {
    "Account": "jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834",
    "Amount": "10000000",
    "Fee": "10",
    "Flags": 0,
    "Method": 0,
    "Payload": "66756E6374696F6E20496E6974282E2E2E2920613D7B7D20666F72206B2C7620696E206970616972732
87B2E2E2E7D2920646F20615B6B5D3D7620656E6420623D615B315D2072657475726E206163636F756E74696E666F28622
920656E643B202066756E6374696F6E20666F6F282E2E2E2920613D7B7D20666F72206B2C7620696E20697061697273287
B2E2E2E7D2920646F20615B6B5D3D7620656E6420623D615B315D2072657475726E206163636F756E74696E666F2862292
0656E64",
    "Sequence": 28,
    "SigningPubKey": "0330E7FC9D56BB25D6893BA3F317AE5BCF33B3291BD63DB32654A313222F7FD020",
    "Timestamp": 559621281,
    "TransactionType": "ConfigContract",
    "TxnSignature": "3045022100A009376F5DCFA056F452C1B0703004A1D181D8479F908E353B2086867F4E89C20220
100CCB09EDFEF416D1B611456BBEF4260E130E48E6AC719EE8A3A89FE65AFE12",
    "hash": "4E00320F4D0FB5567ADB938A6463D62BA93ACF94C049972E2C4DCB01C6B57540",
  }
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
ContractState	String	生成的合约地址
engine_result	String	合约结果
engine_result_code	String	合约结果码
engine_result_message	String	合约结果说明

tx_blob		String	请求合约的blob
tx_json		Object	请求信息
	Account	String	请求的井通地址
	Amount	String	支付金额
	Fee	String	交易费用，井通计价
	Flags	String	交易标识
	Method	String	交易方法：0表示部署，1表示调用
	Payload	String	16进制智能合约代码
	Sequence	String	单子号
	SigningPubKey	String	签名公钥
	Timestamp	Integer	交易时间，UNIXTIME
	TransactionType	String	交易类型
	TxnSignature	String	交易签名
	hash	String	交易hash

5.2 调用合约

接口：/v2/accounts/**{:address}**/contract/call，POST方法

接口参数：

参数	类型	说明
address	String	井通钱包地址

提交参数详情：

参数	类型	说明
secret	String	用户的钱包私钥

destination	String	合约地址
foo	String	合约函数名

可选接口参数：

参数	类型	说明
params	Array	合约参数

POST需要提交的参数格式如下：

```
{
  "secret": "snUaJxp2k4WFt5LCCtEx2zjThQhpT",
  "destination": "jKotgzRHyoa7dywd7vf6LgFBXnv3K66zEg",
  "params":  ["jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834"],
  "foo": "foo"
}
```

例子：

http://localhost/v2/accounts/jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834/contract/call

结果：

```
{
  "success": true,
  "status_code": "0",
  "ContractState": {
    "Account": "jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834",
    "Balance": "599999999829999910",
    "Flags": 0,
    "LedgerEntryType": "AccountRoot",
    "OwnerCount": 0,
    "PreviousTxnID": "4E00320F4D0FB5567ADB938A6463D62BA93ACF94C049972E2C4DCB01C6B57540",
```

```
    "PreviousTxnLgrSeq": 153616,
    "Sequence": 30,
    "index": "2B6AC232AA4C4BE41BF49D2459FA4A0347E1B543A4C92FCEE0821C0201E2E9A8",
  },
  "engine_result": "tesSUCCESS",
  "engine_result_code": 0,
  "engine_result_message": "The transaction was applied. Only final in a validated ledger.",
  "tx_blob": "12001E220000000024000000001E2F215B2D3720240000000168400000000000000A73210330E7FC9D56BB
25D6893BA3F317AE5BCF33B3291BD63DB32654A313222F7FD020744730450221 00A2E6877B386732AB1857259705C38350
9812B60CAF8BF0615521BB06272625C6022052A3E44E6CAD493DED8A0ADF804BFE5B90BD68AD2B268CD42076FC410512FE
8B701103666F6F8114B5F762798A53D543A014CAF8B297CFF8F2F937E88314CE508BEB2DC80614229CB1E64F51A6537322
5835FAEB7012226A486239434A41577942346A7239315652576E3936446B756B4734627764747954 68E1F1",
  "tx_json": {
    "Account": "jsqRs9BDCjyTuRWEPZk3yHa4MFmRi9D834",
    "Args":  [
        {
            "Arg": {
              "Parameter": "6A486239434A41577942346A7239315652576E3936446B756B473462776474795468",
            }
        }
    ],
    "ContractMethod": "666F6F",
    "Destination": "jKotgzRHyoa7dywd7vf6LgFBXnv3K66zEg",
   "Fee": "10",
   "Flags": 0,
   "Method": 1,
   "Sequence": 30,
   "SigningPubKey": "0330E7FC9D56BB25D6893BA3F317AE5BCF33B3291BD63DB32654A313222F7FD020",
   "Timestamp": 559623479,
```

```
  "TransactionType": "ConfigContract",
  "TxnSignature": "3045022100A009376F5DCFA056F452C1B0703004A1D181D8479F908E353B2086867F4E89C20220
100CCB09EDFEF416D1B611456BBEF4260E130E48E6AC719EE8A3A89FE65AFE12",
  "hash": "4E00320F4D0FB5567ADB938A6463D62BA93ACF94C049972E2C4DCB01C6B57540",
 }
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
ContractState	Object	调用方信息
engine_result	String	合约结果
engine_result_code	String	合约结果码
engine_result_message	String	合约结果说明
tx_blob	String	调用合约的blob
tx_json	Object	调用信息
Account	String	井通地址
Args	Array	合约传入的参数
ContractMethod	String	合约函数名
Destination	String	调用的合约地址
Fee	String	交易费用，井通计价
Flags	String	交易标识
Method	String	交易方法：0表示部署，1表示调用
Sequence	String	单子号

	SigningPubKey	String	签名公钥
	Timestamp	Integer	交易时间，UNIXTIME
	TransactionType	String	交易类型
	TxnSignature	String	交易签名
	hash	String	交易hash

6 本地签名

6.1 提交本地签名

接口：/v2/blob，POST方法

注：此接口只将签名结果提交到底层，不提供签名方法。

提交参数详情：

参数	类型	说明
blob	String	签名后的交易信息

POST需要提交的参数格式如下：

```
{
    "blob": "120000220000000024000002846140000000000007A1206840000000000000000A73210224445F6980BBC7F34F5042893C419E536468F92A9034177C0CB786CC7836025B7447304502210 0FC7EA9B7200CA4D3F2C4948E86140F14D5C1FA1CE68682288B51928A7C7256ED02204D3993571B4EEA50A64A3CDB2A38D61EA28F46C3563CF64B48A211E44456738D81147A44B90BCADB1F585D590DC31AB83245E049BB668314B9DFBBDC029B81C608497CE3D61C70D79BCCA955"
}
```

例子：

http://localhost/v2/blob

结果：

```
{
  "success": true,
  "status_code": "0",
```

```
  "engine_result": "tesSUCCESS",
  "engine_result_code": 0,
  "eengine_result_message": "The transaction was applied. Only final in a validated ledger.",
  "tx_blob": "1200002200000000240000028461400000000007A120684000000000000000A73210224445F6980BBC7F34F5042893C419E536468F92A9034177C0CB786CC7836025B74473045022100FC7EA9B7200CA4D3F2C4948E86140F14D5C1FA1CE68682288B51928A7C7256ED02204D3993571B4EEA50A64A3CDB2A38D61EA28F46C3563CF64B48A211E44456738D81147A44B90BCADB1F585D590DC31AB83245E049BB668314B9DFBBDC029B81C608497CE3D61C70D79BCCA955",
  "tx_json": {
    "Account": "jU9VmMHKk1jakMKpjGhWYibnmz5sDw3ZNt",
    "Amount": "500000",
    "Destination": "jHAFp8CrBM48QJyqB8V7NWH9XBsHRQcUu8",
    "Fee": "10",
    "Flags": 0,
    "Sequence": 644,
    "SigningPubKey": "02FE64E0C20F0058F22F3742EDC15F49F318C04F88B130742C68BAF3B1C89FD167",
    "TransactionType": "Payment",
    "TxnSignature": "3045022100948A033830881A3BCF90EBCF79CAD2000149CA347BF3C19997083DD247FBB45D02203977E0AB7B59F3982719F2A38C6363649620CF4F441C8FA7E4FC3CDFAEFA879C",
    "hash": "E663D510536DCE4DEE48B1C6F958B102386E462F80CAA77720921341AFD0A3DC"
  }
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
engine_result	String	交易结果
engine_result_code	String	交易结果码
engine_result_mess	String	交易结果说明

age			
tx_blob		String	请求交易的blob
tx_json		Object	请求信息
	Account	String	请求的井通地址
	Amount	String	支付金额*1000000
	Fee	String	交易费用，井通计价*1000000
	Flags	String	交易标识
	Sequence	String	单子号
	SigningPubKey	String	签名公钥
	TransactionType	String	交易类型
	TxnSignature	String	交易签名
	hash	String	交易hash

7 账本

7.1 获得最新账本号

接口：/v2/ledger/index，GET方法

接口参数：无

例子：

http://localhost/v2/ledger/index

结果：

```
{
  "success": true,
  "status_code": "0",
  "ledger_hash": "B6EC1E9526DBEB2C746DE76366FBF82F9890BA6813B5AEDAD7A27A0B18226B45",
  "ledger_index": 9590468
}
```

返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
ledger_hash	String	账本hash
ledger_index	Integer	账本号/区块高度

7.2 通过账本号获得某一账本信息及交易信息

接口：/v2/ledger/index/**{:index}**，GET方法

接口参数：

参数	类型	说明
index	String	账本号/区块高度

例子：

http://localhost/v2/ledger/index/9590468

结果：


```
{
  "success": true,
  "status_code": "0",
  "accepted": true,
  "account_hash": "DA4D9D7626DAACA4E42C7155E75D3873F492D3DB47B0359F42E336D710C1878B",
  "close_time": 579062540,
  "close_time_human": "2018-May-08 02:42:20",
  "close_time_resolution": 10,

  "closed": true,
  "hash": "B6EC1E9526DBEB2C746DE76366FBF82F9890BA6813B5AEDAD7A27A0B18226B45",
```

```
  "ledger_hash": "B6EC1E9526DBEB2C746DE76366FBF82F9890BA6813B5AEDAD7A27A0B18226B45",
  "ledger_index": "9590468",
  "parent_hash": "042F30BB6FC77424D5A49B12E3F60C2309F350C9B95622E76FFF307B916BACE8",
  "seqNum": "9590468",
  "totalCoins": "599999999999460713",
  "total_coins": "599999999999460713",
  "transaction_hash": "55B92B8C95B1C4AF8F46A65084F70A049565B3926E6E95410ED53000BE6DEBAB",
  "transactions": [
     "432709B5DAA873132819DED7F71B6AF3E89E05AE35268DC7AE13BAB1CF685FCC"
  ]
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
accepted	Boolean	区块是否已经产生
account_hash	String	状态hash树根
close_time	Integer	关闭时间
close_time_human	String	关闭时间
close_time_resolution	Integer	关闭周期
closed	Boolean	账本是否已经关闭
hash	String	账本hash
ledger_hash	String	账本hash
ledger_index	String	账本高度/区块高度
parent_hash	String	上一区块hash值

seqNum	String	账本高度/区块高度
totalCoins	String	swt总量
total_coins	String	swt总量
transaction_hash	String	交易hash树根
transactions	Array	该账本里的交易列表

7.3 通过账本hash获得某一账本信息及交易信息

接口：/v2/ledger/hash/**{:hash}**，GET方法

接口参数：

参数	类型	说明
hash	String	账本hash

例子：

http://localhost/v2/ledger/hash/C08847C320D4F9E136BFCBC58D1F39C630C0F5421A06C739AF30BAF05F94714F

结果：


```
{
  "success": true,
  "status_code": "0",
  "accepted": true,
  "account_hash": "DA4D9D7626DAACA4E42C7155E75D3873F492D3DB47B0359F42E336D710C1878B",
  "close_time": 579062540,
  "close_time_human": "2018-May-08 02:42:20",
  "close_time_resolution": 10,

  "closed": true,
  "hash": "B6EC1E9526DBEB2C746DE76366FBF82F9890BA6813B5AEDAD7A27A0B18226B45",
```

```
  "ledger_hash": "B6EC1E9526DBEB2C746DE76366FBF82F9890BA6813B5AEDAD7A27A0B18226B45",
  "ledger_index": "9590468",
  "parent_hash": "042F30BB6FC77424D5A49B12E3F60C2309F350C9B95622E76FFF307B916BACE8",
  "seqNum": "9590468",
  "totalCoins": "599999999999460713",
  "total_coins": "599999999999460713",
  "transaction_hash": "55B92B8C95B1C4AF8F46A65084F70A049565B3926E6E95410ED53000BE6DEBAB",
  "transactions": [
     "432709B5DAA873132819DED7F71B6AF3E89E05AE35268DC7AE13BAB1CF685FCC"
  ]
}
```


返回的结果信息：

参数	类型	说明
success	Boolean	请求结果
accepted	Boolean	区块是否已经产生
account_hash	String	状态hash树根
close_time	Integer	关闭时间
close_time_human	String	关闭时间
close_time_resolution	Integer	关闭周期
closed	Boolean	账本是否已经关闭
hash	String	账本hash
ledger_hash	String	账本hash
ledger_index	String	账本高度/区块高度
parent_hash	String	上一区块hash值

seqNum	String	账本高度/区块高度
totalCoins	String	swt总量
total_coins	String	swt总量
transaction_hash	String	交易hash树根
transactions	Array	该账本里的交易列表

8 订阅功能

为了减少前端软件的复杂度，井通标准接口（REST API）提供井通地址的websocket长连接服务。前端软件可以通过websocket连接向标准接口订阅和井通地址相关的交易信息。任何和该地址相关的交易信息将会通过相应的websocket连接传出。

REST API订阅后台是API一个服务地址，测试环境的接口是wss://tapi.jingtum.com:5443

8.1 连接服务

连接Websocket服务之后，后台返回订阅ID，如下:

```
{
  "success": true,
  "type": "connection",
  "id": "03afb5a3-0bdb-4a6b-bc6a-4ff72cce05bc"
}
```

8.2 发起订阅

客户端在连接上服务之后，通过发送订阅请求进行订阅，订阅请求如下：

```
{
  "command": "subscribe",
  "type": "account",
  "account": "jDUjqoDZLhzx4DCf6pvSivjkjgtRESY62c"
}
```

订阅请求中，必须将订阅用户的地址和类型一起提交上来，参数type表示所订阅的类型，包含3种：account(订阅某个账号的交易记录)、transactions(订阅所有交易记录)、ledger(订阅账本消息)，目前一个客户端只支持订阅这三种中的一种消息。

订阅成功之后，返回：

```
{
   "success": true,
   "command": "subscribe",
   "type": "account",
   "account": "jDUjqoDZLhzx4DCf6pvSivjkjgtRESY62c"
}
```

当请求的JSON格式不对时，失败返回

```
{
  "success": false,
  "error": "xxx"
}
```

当请求的参数有问题时，返回：

```
{
  "success": false,
  "command": "subscribe",
  "type": "account",
  "account": "jDUjqoDZLhzx4DCf6pvSivjkjgtRESY62c",
  "error": "XXX"
}
```

订阅所有交易记录，参数如下：

```
{
  "command": "subscribe",
  "type": "transactions"
}
```

订阅账本信息，参数如下：

```
{
  "command": "subscribe",
  "type": "ledger"
}
```

error包括缺少account，type，account格式不对，type格式不对等，分别是：

1. missing account
2. account is not valid jingtum address
3. invalid type

8.3 取消订阅

客户端在连接上服务之后，通过发送取消订阅请求进行取消订阅，取消订阅请求如下：

```
{
  "command": "unsubscribe",
  "type": "account",
  "account": "jDUjqoDZLhzx4DCf6pvSivjkjgtRESY62c"
}
```

取消订阅成功时候，返回：

```
{
  "success": true,
  "command": "unsubscribe",
  "type": "account"
  "account": "jDUjqoDZLhzx4DCf6pvSivjkjgtRESY62c"
}
```

当请求的JSON格式有问题时，返回：

```
{
  "success": false,
  "error": "xxx"
}
```

当请求的参数不正确时，返回：

```
{
  "success": false,
  "command": "unsubscribe",
  "type": "account",
  "account": "jDUjqoDZLhzx4DCf6pvSivjkjgtRESYdfad62c",
  "error": "XXX"
}
```

取消订阅，参数不正确主要是account缺失或者格式不正确：

a) account is missing
b) Invalid type
c) 等

8.4 接收消息

用户在做完异步操作之后，且订阅了用户的交易消息，用户可以收到如下的交易信息：

```
{
  "success": true,
  "account": "jsGn5UAxwTuYfN5aa6TBC2KZyDDoFJfM8b",
  "type": "Payment",
  "transaction": {
    "date": 1449484040,
    "hash": "599424E57016194C987F3F8A5F9723251C1458E7F33C4D88BC7624088238313F",
    "type": "sent",
    "fee": "0.000012",
```

```
    "result": "tesSUCCESS",
    "counterparty": "jfEPgtxzezSPievdVNpdyEJndkQki2xEZ6",
    "amount": {
      "value": "0.1",
      "currency": "CNY",
      "issuer": "j3nH2JYBPupQwHLz5UDSD2i7i4nnSsVWb5"
  },
  "effects": [ ]
  }
}
```


基本格式为account: 帐号，type: 交易类型，transaction: 交易信息，交易信息里面的内容和API里面的信息一致，交易类型主要有：Payment 支付类；OfferCreate 创建挂单；OfferCancel 取消挂单；TrustSet 设置信任；AccountSet 设置账号信息；SetRegularKey设置regularkey；RelationSet 设置关系等。

在用户被动成交时，用户也会收到交易相关的信息。

当交易失败时，会有如下的格式的消息：


```
{
  "success": false,
  "account": "jsGn5UAxwTuYfN5aa6TBC2KZyDDoFJfM8b",
  "type": "Payment",
  "transaction": {
    "result": "tecNO_DST_INSUF_SWT",
    "message": "xxx"
  }
}
```


通过success可以判断返回的交易结果是成功的还是失败的，其他的失败记录均即时反馈回给用户，无需在订阅中获得。

8.5 关闭订阅

通过发送如下命令，关闭订阅

```
{
  "command": "close"
}
```

关闭订阅之后，后台返回订阅关闭结果

```
{
  "success": true,
  "command": "close"
}
```

9 错误信息

9.1 客户端错误

ClientError，此错误主要是客户端请求参数错误，包括井通地址格式不对，私钥格式不对，货币格式不对等以及根据每个接口提交的参数格式不对等导致的错误；

9.2 网络错误

NetworkError，此错误主要是网络错误，包括链接井通网络没有连上，请求服务超时等；

9.3 交易错误

TransactionError，此错误主要是重复资源号的错误，即DuplicateTransactionError；

9.4 服务端错误

ServerError，此错误主要是后台程序错误，包括代码BUG、代码实现问题等；

9.5 status_code常见编码附录

编码	说明
0	success
1000	client error
1001	Invalid parameter: address, it is not a valid jingtum address.
1002	Invalid parameter: secret
1003	Invalid parameter: currency
1004	Invalid parameter: issuer
1005	Invalid parameter: order, it is an object composed of four attributes: type, pair, amount and price.
1006	Invalid parameter: order.type, it must be sell or buy
1007	Invalid parameter: order.price, it is a number.
1008	Invalid parameter: order.sequence, it must be a number.
1009	Invalid parameter: hash
1010	not an order transaction
1011	Transaction specified did not affect the given account
1012	Invalid parameter: base, it consists of two fields separated by \'+\' : [currency + counterparty].
1013	Invalid parameter: counter, it consists of two fields separated by \'+\' : [currency + counterparty].
1014	Invalid parameter: destination_address, it is not a valid jingtum address.
1015	Invalid parameter: amount, it is a string composed of three fields separated by \'+\': [value + currency + issuer],and if currency is SWT, the issuer is not needed.
1016	Invalid payment type, it must be an object type.
1017	Invalid parameter: payment.source, it is not a valid jingtum address.
1018	Invalid parameter: payment.destination, it is not a valid jingtum address.
1019	Invalid parameter: payment.amount, it is an object composed of three attributes: value, currency and issuer. And if the currency is SWT, the issuer must be empty string.

1020	Invalid parameter: memos,it must be an array.
1021	Invalid parameter: choice,it must be a string.
1022	Invalid parameter: client_id,must be a number.
1023	Invalid parameter: client_id, already exits.
1024	Invalid parameter: choice, not exist.
1025	Not a payment transaction.
1026	Account not found.
1027	Invalid parameter:page, it is a positive integer.
1028	Invalid parameter:results_per_page, it is a positive integer.
1029	Invalid parameter: order.pair, it consists of two groups of currencies separated by \'/\'.
1030	Invalid order.amount type, it is a number.
1031	Missing parameters
1032	Invalid parameter: method, it must be 0 or 1
1033	Invalid parameter: payload, it must be string
1034	Invalid parameter: amount,it must be a number greater than zero
1035	Invalid parameter: params,it must be an array
1036	Invalid parameter: blob, it must be a string
1037	Invalid parameter: memos, each item in the array must be a string
1038	Missing parameter secret.
1039	Missing parameter client_id.
1040	Missing parameter payment.
1041	Missing parameter payment.source.
1042	Missing parameter payment.destination.
1043	Missing parameter payment.amount.
1044	Invalid parameter: order.pair, base consists of two fields separated by \':\' : [currency : issuer].

1045	Invalid parameter: order.pair, counter consists of two fields separated by \':\' : [currency : issuer].
1046	Invalid parameter: order.pair, Notify developers to view log checking errors.
1047	Invalid parameter: destination, it is not a valid jingtum address.
1048	The product of two numbers(results_per_page * page) must be an integer less than 300, page\'s default value is 1.
1049	Missing parameter:type, Choose from the following three types : trust, authorize or freeze.
1050	Invalid parameter: counterparty, it is not a valid jingtum address.
1051	Invalid parameter: relation amount, it is an object composed of three attributes: limit, currency and issuer. And if the currency is SWT, the issuer must be empty string.
1052	Invalid parameter: type, it must be unfreeze.
1053	Invalid parameter: marker.
1054	The product of two numbers(results_per_page * page) must be an integer greater than 10, page\'s default value is 1.
1055	The product of two numbers(results_per_page * page) must be an integer less than 200, page\'s default value is 1.
1056	Invalid parameter: issuer, It\'s a fixed value.
1057	Invalid parameter: base, if the currency is SWT, the issuer must be empty string.
1058	Invalid parameter: counter, if the currency is SWT, the issuer must be empty string.
1059	Invalid parameter: forward.
1060	Invalid parameter: validated.
1061	Invalid parameter type: foo, it must be string.
1062	Invalid parameter: index, it must be an number.
2000	Server error
3000	Transaction error
3001	Could not generate wallet
3002	Get validated transaction error

3003	Enter into the account book failed
4000	Network error
4001	Remote is disconnected
4002	Time out
4003	Bad gateway

9.6 底层常见错误附录

错误名称	说明
tecCLAIM	Fee claimed. Sequence used. No action.
tecDIR_FULL	Can not add entry to full directory.
tecFAILED_PROCESSING	Failed to correctly process transaction.
tecINSUF_RESERVE_LINE	Insufficient reserve to add trust line.
tecINSUF_RESERVE_OFFER	Insufficient reserve to create offer.
tecNO_DST	Destination does not exist. Send SWT to create it.
tecNO_DST_INSUF_SWT	Destination does not exist. Too little SWT sent to create it.
tecNO_LINE_INSUF_RESERVE	No such line. Too little reserve to create it.
tecNO_LINE_REDUNDANT	Can't set non-existent line to default.
tecPATH_DRY	Path could not send partial amount.
tecPATH_PARTIAL	Path could not send full amount.
tecMASTER_DISABLED	Master key is disabled.
tecNO_REGULAR_KEY	Regular key is not set.
tecUNFUNDED	One of _ADD, _OFFER, or _SEND. Deprecated.
tecUNFUNDED_ADD	Insufficient SWT balance for WalletAdd.
tecUNFUNDED_OFFER	Insufficient balance to fund created offer.
tecUNFUNDED_PAYMENT	Insufficient SWT balance to send.
tecOWNERS	Non-zero owner count.
tecNO_ISSUER	Issuer account does not exist.

tecNO_AUTH	Not authorized to hold asset.
tecNO_LINE	No such line.
tecINSUFF_FEE	Insufficient balance to pay fee.
tecFROZEN	Asset is frozen.
tecNO_TARGET	Target account does not exist.
tecNO_PERMISSION	No permission to perform requested operation.
tecNO_ENTRY	No matching entry found.
tecINSUFFICIENT_RESERVE	Insufficient reserve to complete requested operation.
tecNEED_MASTER_KEY	The operation requires the use of the Master Key.
tecDST_TAG_NEEDED	A destination tag is required.
tecINTERNAL	An internal error has occurred during processing.
tefALREADY	The exact transaction was already in this ledger.
tefBAD_ADD_AUTH	Not authorized to add account.
tefBAD_AUTH	Transaction's public key is not authorized.
tefBAD_LEDGER	Ledger in unexpected state.
tefCREATED	Can't add an already created account.
tefEXCEPTION	Unexpected program state.
tefFAILURE	Failed to apply.
tefINTERNAL	Internal error.
tefMASTER_DISABLED	Master key is disabled.
tefMAX_LEDGER	Ledger sequence too high.
tefNO_AUTH_REQUIRED	Auth is not required.
tefPAST_SEQ	This sequence number has already past.
tefWRONG_PRIOR	This previous transaction does not match.
telLOCAL_ERROR	Local failure.
telBAD_DOMAIN	Domain too long.
telBAD_PATH_COUNT	Malformed: Too many paths.

telBAD_PUBLIC_KEY	Public key too long.
telFAILED_PROCESSING	Failed to correctly process transaction.
telINSUF_FEE_P	Fee insufficient.
telNO_DST_PARTIAL	Partial payment to create account not allowed.
telBLKLIST	Tx disable for blacklist.
telINSUF_FUND	Fund insufficient.
temMALFORMED	Malformed transaction.
temBAD_AMOUNT	Can only send positive amounts.
temBAD_AUTH_MASTER	Auth for unclaimed account needs correct master key.
temBAD_CURRENCY	Malformed: Bad currency.
temBAD_EXPIRATION	Malformed: Bad expiration.
temBAD_FEE	Invalid fee, negative or not SWT.
temBAD_ISSUER	Malformed: Bad issuer.
temBAD_LIMIT	Limits must be non-negative.
temBAD_QUORUM	Quorums must be non-negative.
temBAD_WEIGHT	Weights must be non-negative.
temBAD_OFFER	Malformed: Bad offer.
temBAD_PATH	Malformed: Bad path.
temBAD_PATH_LOOP	Malformed: Loop in path.
temBAD_SEND_SWT_LIMIT	Malformed: Limit quality is not allowed for SWT to SWT.
temBAD_SEND_SWT_MAX	Malformed: Send max is not allowed for SWT to SWT.
temBAD_SEND_SWT_NO_DIRECT	Malformed: No Skywell direct is not allowed for SWT to SWT.
temBAD_SEND_SWT_PARTIAL	Malformed: Partial payment is not allowed for SWT to SWT.
temBAD_SEND_SWT_PATHS	Malformed: Paths are not allowed for SWT to SWT.
temBAD_SEQUENCE	Malformed: Sequence is not in the past.
temBAD_SIGNATURE	Malformed: Bad signature.

temBAD_SRC_ACCOUNT	Malformed: Bad source account.
temBAD_TRANSFER_RATE	Malformed: Transfer rate must be >= 1.0
temDST_IS_SRC	Destination may not be source.
temDST_NEEDED	Destination not specified.
temINVALID	The transaction is ill-formed.
temINVALID_FLAG	The transaction has an invalid flag.
temREDUNDANT	Sends same currency to self.
temREDUNDANTSIGN	Add self as additional sign.
temSKYWELL_EMPTY	PathSet with no paths.
temUNCERTAIN	In process of determining result. Never returned.
temUNKNOWN	The transaction requires logic that is not implemented yet.
temDISABLED	The transaction requires logic that is currently disabled.
temMULTIINIT	contract code has multi init function
terRETRY	Retry transaction.
terFUNDS_SPENT	Can't set password, password set funds already spent.
terINSUF_FEE_B	Account balance can't pay fee.
terLAST	Process last.
terNO_SKYWELL	Path does not permit rippling.
terNO_ACCOUNT	The source account does not exist.
terNO_AUTH	Not authorized to hold IOUs.
terNO_LINE	No such line.
terPRE_SEQ	Missing/inapplicable prior transaction.
terOWNERS	Non-zero owner count.
tesSUCCESS	The transaction was applied. Only final in a validated ledger.