

# JCC\_EXCHANGE 接口文档

一、安装 .....	2
二、使用说明 .....	2
1 交易签名 .....	2
1.1 使用示例 .....	2
2 实例初始化 .....	4
2.1 使用示例 .....	4
3 创建挂单 .....	4
3.1 使用示例 .....	4
4 取消挂单 .....	5
4.1 使用示例 .....	5
5 转账.....	6
5.1 使用示例 .....	6

# 一、安装

```
npm install jcc_exchange
```

# 二、使用说明

## 1 交易签名

### 1.1 使用示例

```
import * as jingtumSignTx from "jcc_jingtum_lib/src/local_sign";
const copyTx = Object.assign({}, tx);
import { swtcSequence } from "jcc_exchange/lib/util";
const sequence = await swtcSequence.get(JCCExchange.getSequence, tx.Account);
copyTx.Sequence = sequence;
const sign = jingtumSignTx(copyTx, { seed: secret });
```

参数说明:

参数		类型	说明
secret		String	井通钱包私钥
tx		Object	需要签名的交易对象
	Account	String	井通钱包地址
	Sequence	Number	交易序列号(通过 getSequence 得到)
	Fee	Number	交易燃料费(gas 费)
	Flags	Number	交易标记(挂卖单时是 0x00080000, 其余情况是 0)
	Platform	String	挂单平台方钱包地址, 缺省为 jDXCeSHSpZ9LiX6ihckWaYDeDt5hFrd Tto(只有创建挂单时才有这个字段)
	TakerGets	Object 或 String	挂单付出的币种和数量(只有创建挂单时才有这个字段)
	TakerPays	Object 或 String	挂单得到的币种和数量(只有创建挂单时才有这个字段)
	OfferSequence	Number	交易序列号(只有取消挂单时才有这个字段)
	Amount	Object 或 String	转账的数量(只有转账时才有这个字段)



## 2 实例初始化

### 2.1 使用示例

```
const hosts = ["localhost"];
const port = 80;
const https = false;
const retry = 3;
JCCExchange.init(hosts, port, https, retry);
```

参数:

hosts: 某个交易服务器  
port: 某个交易服务器端口号  
https: true 为 https, false 为 http  
retry: 交易错误时重试次数, 缺省 3 次

只有初始化之后, 才能后续的创建挂单, 取消挂单, 转账的操作

## 3 创建挂单

### 3.1 使用示例

```
const address = "";
const secret = "";
const amount = "1";
const base = "jjcc";
const counter = "swt";
const sum = "1";
const type = "buy";
const platform = "";
const issuer;
const hash = await JCCExchange.createOrder(address, secret, amount,
base, counter, sum, type, platform, issuer);
```

参数:

参数	类型	说明
address	String	井通钱包地址
secret	String	井通钱包私钥
amount	String	挂单数量
base	String	交易币种(不区分大小写)
count	String	交易的交易区(不区分大小写),根据挂单的类型区分 (若是买单,counter 是付出的币种;若是卖单,counter

		则是得到的币种)
sum	String	挂单的总价
type	String	挂单的类型(买: buy, 卖:sell)
platform	String	挂单平台方钱包地址, 缺省为 jDXCeSHSpZ9LiX6ihckWaYDeDt5hFrdTto (威链平台)
issuer	String	交易币种的发行方, 缺省 jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or

结果如下:

```
hash: "9F1D72707EFBA863B8DAD487857E4D7B4E54E90CF6348BBEA5F32509C4390DE4"
```

结果参数:

参数	类型	说明
hash	String	交易哈希

## 4 取消挂单

### 4.1 使用示例

```
const address = "";
const secret = "";
const orderSequence = 0;
const hash = await JCCExchange.cancelOrder(address, secret,
orderSequence);
```

参数:

参数	类型	说明
address	String	井通钱包地址
secret	String	井通钱包私钥
orderSequence	Number	要取消订单的序列号

结果如下:

```
hash: "6E3D45E77EB20D73439D10958AF28A648DC49302FEE1212BB4D5E010624F554D"
```

结果参数:

参数	类型	说明
hash	String	交易哈希

## 5 转账

### 5.1 使用示例

```
const address = "jpgWGpfHz8GxqUjz5nb6ej8eZJQtIF6KhH";
const secret = "";
const amount = "1";
const memo = "test";
const to = "jKTtq57iqHoHg3cP7Rryzug9Q2puLX1kHh";
const token = "jjcc";
const issuer;
const hash = await JCCExchange.transfer(address, secret, amount,
memo, to, token, issuer);
```

参数:

参数	类型	说明
address	String	转出井通钱包地址
secret	String	转出井通钱包私钥
amount	String	转账数量
memo	String	转账备注，可以为空
to	String	转入(目标)井通钱包地址
token	String	转账币种(不区分大小写)
issuer	String	转账币种的发行方，缺省 jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or

结果如下:

```
hash: "445821D60F9294E4ACF911D95166320D2F8EFFC6B4C6C133701B5327
2684D166"
```

结果参数:

参数	类型	说明
hash	String	交易哈希