

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
6 设置挂单手续费	7
6.1 使用示例	7

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), 同下方 Amount
	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), 同下方 Amount
	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(只有转账和设置手续费时才有这个字段)

	currency	String	货币种类(不区分大小写)
	value	String	转账数量或是“0”(设置手续费时)
	issuer	String	币种发行方(缺省为 jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or)
	FeeAccountID	String	收取手续费的账号(井通钱包地址), 必须为激活账号(只有设置手续费时 才有这个字段)
	OfferFeeRateNum	Number	手续费率分子(0 和正整数)(只有 设置手续费时才有这个字段)
	OfferFeeRateDen	Number	手续费率分母(正整数)(只有设置 手续费时才有这个字段)
	Destination	String	转账目标钱包地址(只有转账时才有 这个字段)
	Memos	Array 或 String	转账备注(只有转账时才有这个字 段, 可以不传值)
	TransactionType	String	交易类型 OfferCreate (创建挂单) OfferCancel (取消挂单) Payment (转账) Brokerage(设置手续费)

参数示例: (已创建挂单的为列)

```
const tx = {
  Flags: 0,
  Fee: 0.00001,
  Sequence: 0,
  Account: 'jpgWGpfHz8GxqUjz5nb6ej8eZJQtIF6KhH',
  TransactionType: 'OfferCreate',
  TakerGets: {
    value: '0.03',
    currency: 'CNY',
    issuer: 'jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or'
  },
  TakerPays: '1',
  Platform: "jDXCeSHSpZ9LiX6ihckWaYDeDt5hFrdTto"
}
```

结果如下:

```
120007220000000006440000000000F424065D40AA87BEE538000000000000000000
0000000434E5900000000000A582E432BFC48EEDEF852C814EC57F3CD2D41596684000
000000000000A732102C13075B18C87A032226CE383AEFD748D7BB719E02CD7F5A8C1F
2C7562DE7C12A74473045022100DDAAA2406910BCDF7355ABDBFC8863099BA1575956
5E771C0E4396E37843880D02203D296D5BAB6966545E9AE5F809D149B2A33EE29551A
```

9F3AC7F61CBD40D94BD4D81141270C5BE503A3A22B506457C0FEC97633B44F7DD8D14
896E3F7353697ECE52645D9C502F08BB2EDC5717

结果参数:

参数	类型	说明
	String	签名后的交易数据

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: **sequence** 失效后重试次数, 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:"445821D60F9294E4ACF911D95166320D2F8EFFC6B4C6C133701B53272684D166"
```

结果参数：

参数	类型	说明
hash	String	交易哈希

6 设置挂单手续费

6.1 使用示例

```
const platformAccount = "jJSEWTMsB3WFsZyoGGi977wQdTqTmkBFwV";
const platformSecret = "sxxxxxxx";
const feeAccount = "j98XXLZUCqvP4x7rgFeeYekNgZCotffBH5";
const rateNum = "2";
const rateDen = "1000";
const token = "jjcc";
const issuer;
const hash = await JCCExchange.setBrokerage (platformAccount,
platformSecret, feeAccount, rateNum, rateDen, token, issuer);
```

参数:

参数	类型	说明
platformAccount	String	平台方井通钱包地址
platformSecret	String	平台方井通钱包私钥
feeAccount	String	收费井通钱包地址
rateNum	Number	费率分子(0 和正整数)
rateDen	Number	费率分母(正整数)
token	String	收取手续费的币种(不区分大小写)
issuer	String	币种的发行方, 缺省 jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or

结果如下:

```
hash: "95A3F55A7DD81695D511CF7F257A9B86FA9DF391B000517A38A8AE303A0CB200"
```

结果参数:

参数	类型	说明
hash	String	交易哈希