**基于井通APIv2的SDK说明文档**

V1.0.0

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **简介** | **作者** | **日期** |
| 1.0.0 | 1.0.0创建文档 | Foreso | 2018/4/24 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1. 账号类交易 3](#_Toc17181)

[1.1. 创建账号 3](#_Toc19874)

[1.2. 账号余额 3](#_Toc21356)

[2. 支付类交易 5](#_Toc29551)

[2.1. 提交支付 5](#_Toc23277)

[2.2. 查询支付选择 6](#_Toc31440)

[2.3. 查询支付历史 7](#_Toc20802)

[3. 挂单类交易 8](#_Toc16396)

[3.1. 提交挂单 8](#_Toc8188)

[3.2. 取消挂单 9](#_Toc13109)

[3.3. 查询挂单信息 9](#_Toc30880)

[3.4. 查询市场深度 10](#_Toc8841)

[4. 记录类信息 12](#_Toc4095)

[4.1. 查询交易信息 12](#_Toc14589)

[4.2. 查询交易历史 12](#_Toc4370)

# 账号类交易

## 创建账号

井通账号是非对称密码学上的公钥和私钥的编码，用户可以随时生成新的账号。但是在井通系统中，账号需要有在井通系统登记后才能使用系统功能。这一过程是通过井通系统内的已有账号支付一定量的井通（SWT）实现新账号的登记，也称之为“激活”。

商户可以通过商户银关进行账号的激活，银关需先配置具有一定数量的SWT来完成多个账号的激活。系统默认的最小激活值是25 SWT， 银关可以设置比这个值大的激活值。

账号是用户在区块链系统的操作对象，用户可以通过静态方法创建一个新的井通账号，也可以通过已有的账号的密钥生成一个账号，又称为钱包。 通过已经激活的井通账号可以直接生成账号对象，从而实现井通系统中一系列相应的操作，如支付数字资产，挂单数组资产，以及查询账号的。生成账号时必需提供账号私钥。公钥则可以省略， 由SDK通过私钥计算生成。如果同时提供私钥和公钥，那么输入时私钥在前，公钥在后。

Example

Jingtum.API.Wallet newWallet = Jingtum.API.Wallet.New();

## 账号余额

每个账号容纳用户的资产，包括SWT资产和用户通资产。余额是一个资产列表，每个项目包括资产编码、资产数量以及资产对应的发行银关， 同时根据用户的操作每个用户的资产余额里面还有冻结项目，例如用户挂单卖出CNY，则部分的CNY被冻结了。

在井通系统中，SWT还作为用户资源占用的计数，包括账号资源占用，市场单子资源占用，关系资源占用等。例如每个账号占用10个SWT， 每个提交的货币挂单占用1SWT，每个关系资源占用1SWT等。对于初始账号，一般是占用11个SWT。

余额的查询可以使用用户通的符号进行查找，仅返回相应用户通的余额。

Example

string address = this.textBox1.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

List<Jingtum.API.Balance> balances = wallet.GetBalanceList();

# 支付类交易

## 提交支付

支付是区块链系统的一个主要功能，实现账号之间点对点的电子资源转移。

在井通系统中，资源的转移通过账号地址即可实现，无需用户的业务信息。在井通系统中，用户不仅可以支付自己拥有的资源， 也可以选择支付自己账号中没有的资源。当用户支付自己账号中没有的资源时，需要提交支付路径查询并选择相应支付路径。 系统根据用户账号中含有资源，同时在系统中选择对应的支付路径进行资产替换，实现目标资产的支付。 支付路径的实现依赖于多资产的撮合算法。具体请参考查询支付路径。如果支付路径不存在，则支付无法进行。

支付操作对象初始化时需要一个井通账号对象，必须提交的参数有对方的井通地址和支付的金额。 支付操作还可以设定操作模式（同步或异步）和一个操作单号。

支付操作中也可以使用setMemo设定一些注释字段。

井通系统中支付交易收取相应的SWT手续费，即使交易失败。

目前，井通环境中只有一个井通银关，所以资产的支付是一个银关之间资产的交易， 一个银关下的同一种资产的支付可以实现直接支付，无需路径。 同时，在井通系统中SWT没有发行银关，支付时无需指定发行银关

Example

string address = this.textBox12.Text;

string secret = this.textBox13.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address, secret);

Jingtum.API.Payment payment = new Jingtum.API.Payment();

Jingtum.API.Amount amount = new Jingtum.API.Amount();

amount.Value = int.Parse(this.textBox14.Text);

amount.Currency = this.textBox11.Text;

amount.Issuer = string.Empty;

payment.Amount = amount;

List<string> memos = new List<string>();

memos.Add("Test: " + DateTime.Now.ToString("o"));

string distinationAddress = this.textBox15.Text;

Jingtum.API.SetPaymentResponse res = wallet.SetPayment(payment, distinationAddress, string.Empty, memos);

this.ShowPropertyValue<Jingtum.API.SetPaymentResponse>(res);

## 查询支付选择

在井通系统中进行支付时，用户不仅可以支付自己拥有的资源，也可以选择支付自己账号中没有的资源。

当用户支付自己账号中没有的资源时，需要提交支付选择查询并选择相应支付选择。支付选择这里是指一种用户通 兑换另一种用户通的寻径方法。如果井通系统中存在相应的资源挂单，则井通系统会自动使用系统中的挂单来计算支付选择。 两种不同的用户通之间可以有多种支付选择。

Example

string address = this.textBox7.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

string distinationAddress = this.textBox8.Text;

int amount = int.Parse(this.textBox9.Text);

string currency = this.textBox10.Text;

string issue = "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or";

List<Jingtum.API.PaymentChoice> choices = wallet.GetPaymentChoices(distinationAddress, amount, currency, issue);

## 查询支付历史

当井通账号的支付交易完成后，所有的记录都存储在井通区块链上。 用户可以查询账号的支付历史，包括收款方，时间，通的种类和数量。 用户即可以通过交易完成后的HASH值来查询单一交易的情况，也可以根据条件来过滤所查询的支付历史。 参数包括支付方地址，对方地址，是否移除失败的支付历史，支付方向，返回的每页数据量、返回第几页的数据。 返回的结果信息里面的每个对象的字段解释信息请见数据格式的说明部分。 请注意如果不设置选项的话，默认返回的支付历史是第一页的10条数据。

Example

string address = this.textBox2.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

string hash = this.textBox3.Text;

Jingtum.API.Payment payment = wallet.GetPayment(hash);

string address = this.textBox4.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

List<Jingtum.API.Payment> payments = new List<Jingtum.API.Payment>();

payments = wallet.GetPaymentList(int.Parse(this.textBox5.Text), int.Parse(this.textBox6.Text));

# 挂单类交易

## 提交挂单

挂单是用户资产交换的意愿表达，用于用一种资产置换另一种资产。在挂单中，需要标明挂单的资产对，例如AAA/BBB，其中AAA为基准资产，BBB为目标资产，基准资产是用户挂单的操作资产，目标资产是用户挂单的被动资产。

用户在挂单的时候，需要表达挂单的类型，分别是买单或者卖单，挂单类型指的是对基准货币的操作，例如买单AAA/BBB， 表示买AAA；卖单AAA/BBB，表示卖AAA。

用户挂单之后，系统根据用户的挂单信息，自动撮合用户的挂单。如果不能立即成交则用户的单子，用户的挂单处于未成交状态。

Example

string address = this.textBox31.Text;

string secret = this.textBox32.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address, secret);

Jingtum.API.Order order = new Jingtum.API.Order();

order.Type = this.radioButton6.Checked ? "sell" : "buy";

order.Amount = double.Parse(this.textBox33.Text);

order.Price = double.Parse(this.textBox34.Text);

order.Pair = (this.radioButton6.Checked ? "SWT/CNY" : "CNY/SWT") + ":jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or";

## 取消挂单

当井通账号提交挂单后，在挂单未完全成交前，账号可以取消未完成的挂单。取消的依据是未完成的单子的序号， 在提交挂单时由系统返回的一个大于零的整数值。每个账号做的每笔交易都有个序号，只有挂单交易的序号用于操作。

Example

string address = this.textBox35.Text;

string secret = this.textBox36.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address, secret);

int order = int.Parse(this.textBox37.Text);

## 查询挂单信息

当井通账号对所属资产挂单成功后，可以通过挂单返回的Hash值查询单个挂单信息。 井通账号如果有多个挂单挂出，在挂单未成交时，也可以获得这些挂单的信息。 如果挂单成交，则通过查询交易可以得知挂单成交的信息。

此外，井通区块链系统中也可以查询某一对资产的挂单信息，比如使用用户通A置换用户通B。

Example

string address = this.textBox23.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

string hash = this.textBox24.Text;

Jingtum.API.Order order = wallet.GetOrder(hash);

string address = this.textBox25.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

int pageSize = int.Parse(this.textBox26.Text);

int page = int.Parse(this.textBox27.Text);

List<Jingtum.API.Order> orders = wallet.GetOrderList(pageSize, page);

## 查询市场深度

市场深度是市场里面对于某个货币对的所有挂单，通过市场深度功能，可以获得卖单和买单列表的结果

Example

string address = this.textBox28.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

int pageSize = int.Parse(this.textBox29.Text);

int page = int.Parse(this.textBox30.Text);

string baseCurrency = "SWT";

string baseCounterParty = "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or";

string counterCurrency = "CNY";

string counterCounterParty = "jGa9J9TkqtBcUoHe2zqhVFFbgUVED6o9or";

Jingtum.API.OrderBookResponse response = null;

if(this.radioButton1.Checked)

{

response = wallet.GetOrderBooks(baseCurrency, baseCounterParty, counterCurrency, counterCounterParty, pageSize, page);

}

else if (this.radioButton2.Checked)

{

response = wallet.GetOrderBookBids(baseCurrency, baseCounterParty, counterCurrency, counterCounterParty, pageSize, page);

}

else if (this.radioButton3.Checked)

{

response = wallet.GetOrderBookAsks(baseCurrency, baseCounterParty, counterCurrency, counterCounterParty, pageSize, page);

}

# 记录类信息

## 查询交易信息

在交易完成后，用户可查询单个交易记录信息。输入参数为交易完成后返回的hash值。 井通交易记录类型分为如下几种：

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

Example

string address = this.textBox16.Text;

string hash = this.textBox17.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

Jingtum.API.Transaction transaction = wallet.GetTransaction(hash);

## 查询交易历史

井通提供查询账号的交易历史，包括交易时间、交易Hash、交易费用等内容，同时提供多种维度的查询方法， 包括根据支付方地址，对方地址，是否移除失败的支付历史，支付方向，返回的每页数据量、返回第几页的数据。 返回的结果信息里面的每个对象的字段解释信息请见数据格式的说明部分。请注意如果不设置选项的话， 默认返回的交易历史是第一页的10条数据。

Example

string address = this.textBox18.Text;

Jingtum.API.Wallet wallet = new Jingtum.API.Wallet(address);

int pageSize = int.Parse(this.textBox19.Text);

int page = int.Parse(this.textBox20.Text);

int ledger = this.textBox21.Text == string.Empty ? 0 : int.Parse(this.textBox21.Text);

int seq = this.textBox22.Text == string.Empty ? 0 : int.Parse(this.textBox22.Text);

List<Jingtum.API.Transaction> transactions = wallet.GetTransactions(pageSize, page, ledger, seq);