**Wallet钱包接口文档**

**v1.5**

**目录**

[**Wallet钱包接口文档** 1](#_Toc514096606)

[**v1.0** 1](#_Toc514096607)

[概览 3](#_Toc514096608)

[版本号 3](#_Toc514096609)

[更新历史 3](#_Toc514096610)

[概览 3](#_Toc514096611)

[规则 4](#_Toc514096612)

[服务地址 4](#_Toc514096613)

[协议规则 4](#_Toc514096614)

[错误处理 4](#_Toc514096615)

[入参/出参 5](#_Toc514096616)

[安全 7](#_Toc514096617)

[数据加密 7](#_Toc514096618)

[签名算法 7](#_Toc514096619)

[平台异步通知 8](#_Toc514096620)

[接口列表 9](#_Toc514096621)

[newAccount创建用户链上账户 9](#_Toc514096622)

[getBalance账户余额查询 11](#_Toc514096623)

[sendTransaction账户转账 13](#_Toc514096624)

[getTransaction交易详情查询 16](#_Toc514096625)

[异步通知交易id（txid） 18](#_Toc514096626)

[异步通知交易状态 20](#_Toc514096627)

[附录 23](#_Toc514096628)

[错误码 23](#_Toc514096629)

[支持货币格式 24](#_Toc514096630)

[转账状态属性值 24](#_Toc514096631)

[PHP，C#和JAVA RSA签名及验签 25](#_Toc514096632)

[关于转账的二次签名 25](#_Toc514096633)

# 概览

# 版本号

1.5

## 更新历史

| **更新时间** | **版本** | **更新说明** |
| --- | --- | --- |
| 2018-4-16 | 1.0 | 新建。 |
| 2018-5-14 | 1.5 | 签名方案整理 |

## 概览

### 接口

| **接口** | **说明** |
| --- | --- |
| newAccount | 创建用户链上账户 |
| getBalance | 查询余额 |
| sendTransaction | 转账 |
| getTransaction | 查询交易详情 |

# 规则

## 服务地址

Wallet服务的接入地址为http(s)://ip:port/wallet/xxx。

## 协议规则

|  |  |
| --- | --- |
| 传输方式 | 支持通过HTTP通道进行请求通信 |
| 提交方式 | 采用POST方式提交。 |
| 数据格式 | 全部采用JSON格式。 |
| 字符编码 | 统一采用UTF-8。 |
| 签名算法 | SHA1withRSA。 |
| 签名要求 | 请求和接收数据均需要校验签名。 |
| 判断逻辑 | 先判断HTTP状态码，再处理业务逻辑。 |

## 错误处理

使用 HTTP 状态码 (status code) 来表明一个 API 请求的成功或失败状态。返回 HTTP 2XX 表明 API 请求成功。返回 HTTP 4XX 表明在请求 API 时提供了错误信息，例如参数缺失、参数错误、支付渠道错误等。返回 HTTP 5XX 表明 API 请求时，服务器发生了未知错误。

### HTTP 返回码

| **状态码** | **说明** |
| --- | --- |
| 200 – OK | 一切正常。 |
| 400 - Bad Request | 一般是由于参数缺失，参数不正确导致。 |
| 401 - Unauthorized | 一般是由于权限认证出错导致。 |
| 403 - Forbidden | 一般是由于流量或风险控制而拒绝导致。 |
| 5XX - Server Errors | 一般是服务器内部出现了错误。 |

## 入参/出参

### 请求参数

统一上传如下属性：

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| params | json | 是 | 封装的请求数据 |
| sign | string | 是 | params值签名，请参考签名算法 |
| nonce | string | 是 | 随机数，请参照随机数生成法 |
| timestamp | string | 是 | 时间戳，十三位毫秒级时间戳 如：1524146014086 |
| sysId | string | 是 | 业务系统唯一标识 |

#### 示例

{

"params" : {

"txId": "12345678",

"id": "12345678"

},

"sysId": "1111111",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

### 响应参数

统一返回如下属性：

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| data | json | 否 | 在成功时必须包含，当引起错误时必须不包含该成员 |
| errCode | int | 是 | 成功为-1，错误情况为错误代码 |
| errMsg | string | 是 | 错误详情 |
| flag | boolean | 是 | 成功标志 true:成功 flase :失败 |
| sign | string | 是 | 数据签名，请参数**签名算法** |
| nonce | string | 是 | 随机字符串，不长于32位，详见**随机数生成算法**。 |
| timestamp | string | 是 | 时间戳 如：1524146014086 |

#### 示例

{

"data": {

"address": "e1e9f04de623f676047c0650b36906b495ed7dc6"

},

"errCode": -1,

"errMsg": "",

"flag": true,

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

或者

{

"errCode": 1001,

"errMsg": "签名校验失败",

"flag": false,

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

} }

### 错误码(code)

请参考附录错误码

# 安全

## 数据加密

无（内网通信）

## 签名算法

请求签名生成的步骤如下：

第一步，对请求报文中的参数进行排序，然后拼接参数，对每个非空参数之间用&符号连接，然后使用sha1进行摘要算法，得到字符串stringA。

第二步，使用RSA私钥对stringA进行签名，得到字符串stringB，具体参见附录签名及验签示例。

返回报文签名生成的步骤如下：

第一步

正常的场合对返回报文中的进行排序，然后拼接参数，对每个非空参数之间用&符号连接，然后使用sha1进行摘要算法，得到字符串stringA。

第二步，使用RSA私钥对stringA进行签名，得到字符串stringB，具体参见附录签名及验签示例。

### 随机数生成算法

协议中包含字段nonce，主要保证签名不可预测。我们推荐生成随机数算法如下：调用java的uuid生成去除中间的-符号。随机数会用来做重放攻击判断，同一个随机数在一定时间内不可重复使用。

### 时间有效期

为保证请求的有效性，加入了时间戳的判断，时间戳为毫秒级，例如1514739661000，时间戳的有效期为10分钟。

# 平台异步通知

正常情况，平台会及时将订单状态通知到商户系统，通知策略如下：



**说明**：

平台会在 25 小时内向商户后台不断重发通知，最多 8 次。重试通知时间间隔分别为 5s、10s、2min、5min、10min、30min、1h、2h、6h、15h，一旦收到商户系统的正确响应则不再通知。通知协议详见**结果通知**接口。

# 接口列表

## newAccount创建用户链上账户

### 应用场景

给用户绑定一个链上账户

### 接口地址

http(s):*//ip:port/wallet/newAccount*

### 请求参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |

#### 示例

{

"params": {},

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

"sysId": "zx0001"

}

### 响应参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| address | string | 是 | 用户帐号绑定的地址 |

#### 示例

{

"sign": "H7YNFphJzyE2L2+o80z5miRrH8cYmW+SuilsjjFRlSvuPMvpBdY/vDgkM7t7dMtM6vLZWN9rlsvK9riwMoGKHRXhxoVMOfUJvo7NTr6BgcKY5QymYoYTtxzuhOz8V6SRikk7sFkweUwAW71DO1pVsr30EdWJy5f/azhhPKD+3Lo=",

"flag": true,

"data": {

"sysId": "zx0001",

"address": "0x02c1b1dfce9fcd7614324f91cae2aae1fc4a4dae"

},

"errCode": -1,

"errMsg": ""

}

## getBalance账户余额查询

### 应用场景

根据传来的地址列表，查询每个帐户的余额，地址为空时查询所有地址余额，一次最多查询20个地址的数据。

### 接口地址

http(s):*//ip:port/wallet/getBalance*

### 请求参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| addressList | list | 是 | 需要查询帐号的列表 |

#### 示例

{

"params": {

"addressList": ["0x972d5Ad8a8e681C59249BAe8F2fB21D467CfabB5",

"0xAE95C217a3a97969Ea8c0826c7a11f12440159FD",

"0x3efac9d63b1a0079bed0e5ba95fe409cd456734d",

"0xdff877f31d820964a2d04392c23ae1a98331b6e0"

]

},

"sysId": "zx0001"

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

### 响应参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| Map | map | 是 | 各帐号余额情况  key:address value :List< balanceDetails> |

详细说明：

BalanceDetailInfo

|  |  |  |  |
| --- | --- | --- | --- |
| currency | string | 是 | 币种：eth、hnb等 |
| amount | BigDecimal | 是 | 余额数量 |

#### 示例

{

"nonce": "HVY3BY4WRQNDTJM7",

"timestamp": "1525083428026",

"sign": "YU9kzBEfeiFxA6LCacG7YC7WipgDtIxhO4oOGZgfYCs+5B+KiUPuflu7usvBH7fFcjBS10pcvD2TL6tP2TV9j7zvWjsEXl0Ge+bv9R36Ms8F4mYGrkE6CmjnJXdD04gavhMSJI57g9Hrv4AX7VYiO1klbQiibJ1xs/8LSr9Ypcw=",

"flag": true,

"data": {

"balanceDetails": {

"0xdff877f31d820964a2d04392c23ae1a98331b6e0": [

{

"currency": "ETH",

"amount": 50.199076

},

{

"currency": "HNB",

"amount": 200

}

],

"0x972d5Ad8a8e681C59249BAe8F2fB21D467CfabB5": [

{

"currency": "ETH",

"amount": 116.297525092

},

{

"currency": "HNB",

"amount": 989580.000000000009997298

}

],

"0xAE95C217a3a97969Ea8c0826c7a11f12440159FD": [

{

"currency": "ETH",

"amount": 58.3

},

{

"currency": "HNB",

"amount": 1000515.0000000000000218

}

],

"0x3efac9d63b1a0079bed0e5ba95fe409cd456734d": [

{

"currency": "ETH",

"amount": 279.78314316

},

{

"currency": "HNB",

"amount": 9800

}

]

}

},

"errCode": -1,

"errMsg": ""

}

## sendTransaction账户转账

### 应用场景

根据转账信息进行批量转账服务

### 接口地址

http(s):*//ip:port/wallet/sendTransaction*

### 请求参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| verSignText | string | 是 | 签名内容List< TransactionDetailDto> |
| verSign | string | 是 | 审核员签名 |

详细说明：

transactionDetailDto

|  |  |  |  |
| --- | --- | --- | --- |
| amount | decimal | 是 | 交易金额 |
| currency | string | 是 | 币种：eth、hnb等 |
| fromAddress | string | 是 | 转出地址，业务平台提币操作时传空 |
| toAddress | string | 是 | 转入地址 |
| sourceId | string | 是 | 业务系统交易唯一标识 |
| userId | string | 是 | 审核员唯一id |
| verifyCode | string | 是 | 手机验证码 |

#### 示例

{

"params": {

"transactions": [{

"amount": 20.10000000,

"currency": " ETH",

"fromAddress": "0x3efac9d63b1a0079bed0e5ba95fe409cd456734d",

"toAddress": "0xdff877f31d820964a2d04392c23ae1a98331b6e0",

"sourceId": "1234567",

"userId": "sdfashjdhfkjashdfaskjhfaslkjdhfa",

"verifyCode": "1234567"

},

{

"amount": 100.00,

"currency": " HNB",

"fromAddress": "0x3efac9d63b1a0079bed0e5ba95fe409cd456734d",

"toAddress": "0xdff877f31d820964a2d04392c23ae1a98331b6e0",

"sourceId": "1234568 ",

"userId": "sdfashjdhfkjashdfaskjhfaslkjdhfa",

"verifyCode": "1234567"

}]

},

"verSign": "1111111"

"sysId": " zx0001",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

### 响应参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| receipts | list | 是 | 交易回执 |

详细说明：

receipt

|  |  |  |  |
| --- | --- | --- | --- |
| orderNo | string | 是 | 交易流水号 |
| sourceId | string | 是 | 业务系统交易唯一标识 |

#### 示例

{

"data": {

"receipts": [{

"orderNo": "20180430182116514",

"sourceId": "1234567"

},

{

"orderNo": "20180430182116218",

"sourceId": "1234568"

}]

},

"errCode": -1,

"errMsg": "",

"flag": true,

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

## getTransaction交易详情查询

### 应用场景

根据txid查询转账情况

### 接口地址

http(s):*//ip:port/wallet/getTransaction*

### 请求参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| txid | string | 否 | 交易id，用此id可在geth上直接查询交易详情 |
| orderNo | string | 否 | 钱包服务的交易订单号 |

#### 示例

{

"params": {

"txid":

"0x035b0c5bf777fc24054e101cae8e7ec753a6451900d734165d361446ab180213",

"orderNo": "20180428125852652"

},

"sysId": "zx0001",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

### 响应参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| amount | decimal | 是 | 交易金额 |
| currency | string | 是 | 币种：eth、hnb等 |
| fromAddress | string | 否 | 转出地址（为空的场合，表示提现） |
| toAddress | string | 是 | 转入地址 |
| sourceId | string | 否 | 业务系统交易唯一标识 |
| createTime | string | 是 | 交易时间 |
| tradeBlockNo | BigInteger | 否 | 交易发生区块号 |
| tradeFee | decimal | 否 | 交易手续费 |
| tradeStatus | string | 是 | 交易状态 |
| txid | string | 是 | 交易id |
| orderNo | string | 否 | 钱包服务的交易订单号 |

#### 示例

{

"data": {

"crateTime": "2018-04-19 23:43:43 643",

"fromAddress": "0x3efac9d63b1a0079bed0e5ba95fe409cd456734d",

"sourceId": "5969ecfef4664a73448cabcfce331c7b",

"toAddress": "0x514954e162454666aba220d8c2a672c8ef865e51",

"amount": 0,

"tradeBlokNo": 56026,

"tradeFee": 0.0044,

"orderNo": "e1e9f0444423f676047c0650b36906b495ed7dc6",

"tradeStatus": "02",

"currency": "HNB",

"txid":

"0x035b0c5bf777fc24054e101cae8e7ec753a6451900d734165d361446ab180213"

},

"errCode": -1,

"errMsg": "",

"flag": true,

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

## 异步通知交易id（txid）

### 应用场景

由于钱包服务组装转账报文时，需要操作的内容较多，占用时间较长，因此组装发送交易的方式采用异步的方式进行，交易产生的交易id又回调的方式通知业务系统。

### 接口地址

业务系统提供

### 请求参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| sourceId | string | 是 | 业务系统交易唯一标识 |
| orderNo | string | 是 | 钱包服务交易订单号 |
| txid | string | 是 | 交易id，可用此id在geth上直接查询 |

#### 示例

{

"data": {

"sourceId": "5969ecfef4664a73448cabcfce331c7b",

"orderNo": "e1e9f0444423f676047c0650b36906b495ed7dc6",

"txId": "e1e9f0444423f676047c0650b36906b495ed7dc6"

},

"errCode": -1,

"errMsg": "",

"flag": true,

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

### 响应参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| status | boolean | 是 | 回执状态 ture:更新成功 flase:更新失败 |
| msg | String | 否 | 更新失败的场合，必须项目 |

#### 示例

{

"params": {

"status": true

},

"sysId": "1111111",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

或者

{

"params": {

"status": false,

"msg": "订单不存在"

},

"sysId": "1111111",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

## 异步通知交易状态

### 应用场景

交易的确认不能够在转账发起的时候立即获取结果，因此当Wallet钱包服务监听到本系统用户的交易已经确认完毕，会向业务系统发送一个交易成功通知请求，推送机制请参考**平台异步通知**

### 接口地址

业务系统提供

### 请求参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| toAddress | string | 是 | 转入地址 |
| sourceId | string | 否 | 业务系统交易唯一标识 |
| createTime | string | 是 | 交易时间 |
| tradeBlockNo | BigInteger | 否 | 交易发生区块号 |
| tradeFee | decimal | 否 | 交易手续费 |
| tradeStatus | string | 是 | 交易状态 |
| txid | string | 是 | 交易id |
| amount | decimal | 是 | 交易金额 |
| currency | string | 是 | 币种：eth、hnb等 |
| fromAddress | string | 否 | 转出地址（为空的场合，表示提现） |

#### 示例

{

"data": {

"crateTime": "2018-04-19 23:43:43 643",

"fromAddress": "e1e9f0444423f676047c0650b36906b495ed7dc6",

"sourceId": "5969ecfef4664a73448cabcfce331c7b",

"toAddress": "e1e9f0444423f676047c0650b36906b495ed7dc6",

"tradeBlokNo": 54721231,

"tradeFee": 0.20000000,

"orderNo": "e1e9f0444423f676047c0650b36906b495ed7dc6",

"tradeStatus": "02",

"currency": "hnb",

"txId": "e1e9f0444423f676047c0650b36906b495ed7dc6"

},

"sysId": "1111111",

"errCode": -1,

"errMsg": "",

"flag": true,

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

### 响应参数

| **属性** | **类型** | **是否必须** | **描述** |
| --- | --- | --- | --- |
| status | boolean | 是 | 回执状态 ture:更新成功 flase:更新失败 |
| msg | String | 否 | 更新失败的场合，必须项目 |

#### 示例

{

"params": {

"status": true

},

"sysId": "1111111",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

或者

{

"params": {

"status": false,

"msg": "订单不存在"

},

"sysId": "1111111",

"sign": "1234567890abcdef",

"nonce": "c49d982e211e423eb4038853227cb81b",

"timestamp": "1524146014086"

}

# 附录

## 错误码

| **错误码** | **说明** |
| --- | --- |
| 1000 | 参数不能为空 |
| 1001 | 签名错误 |
| 1002 | 重放攻击异常 |
| 1003 | 请求已经过期 |
| 1004 | 转账异常，GAS不够 |
| 1005 | 此系统ID未授权 |
| 1006 | 此交易异常没有成功发出 |
| 1007 | 钱包流水号异常 |
| 1008 | 手机验证码错误 |
| 1009 | 审核员签名验证失败 |
| 400 | Bad Request! |
| 401 | NotAuthorization |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 500 | Internal Server Error |
| 2000 | [服务器]运行时异常 |
| 2001 | [服务器]空值异常 |
| 2002 | [服务器]数据类型转换异常 |
| 2003 | [服务器]IO异常 |
| 2004 | [服务器]未知方法异常 |
| 2005 | [服务器]数组越界异常 |
| 2006 | [服务器]网络异常 |

## 支持货币格式

| **支付渠道** | **描述** |
| --- | --- |
| eth | 以太坊 |
| hnb |  |
| hgs |  |

## 转账状态属性值

| **订单状态** | **描述** |
| --- | --- |
| 01 | 等待交易打包 |
| 02 | 等待确认交易（<=12个确认） |
| 03 | 交易完成（>12个确认） |
| 04 | 交易失败 |

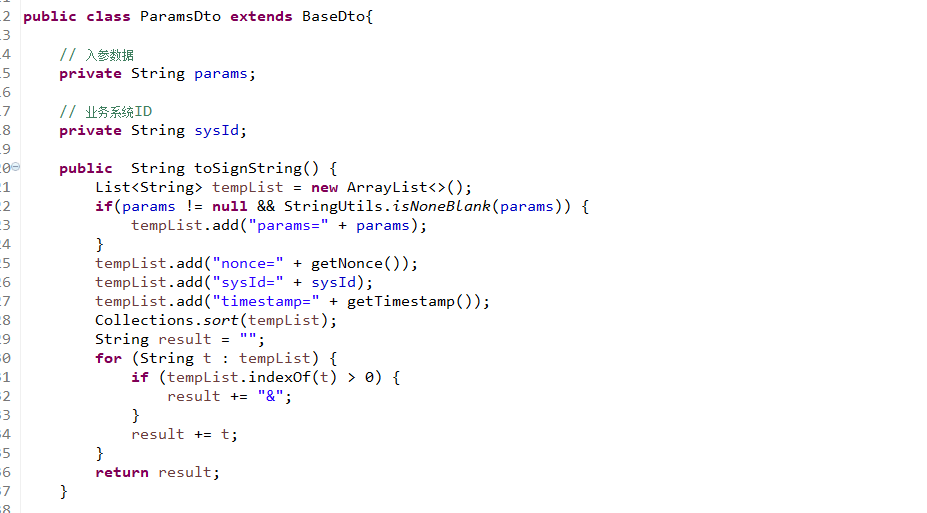
## PHP，C#和JAVA RSA签名及验签

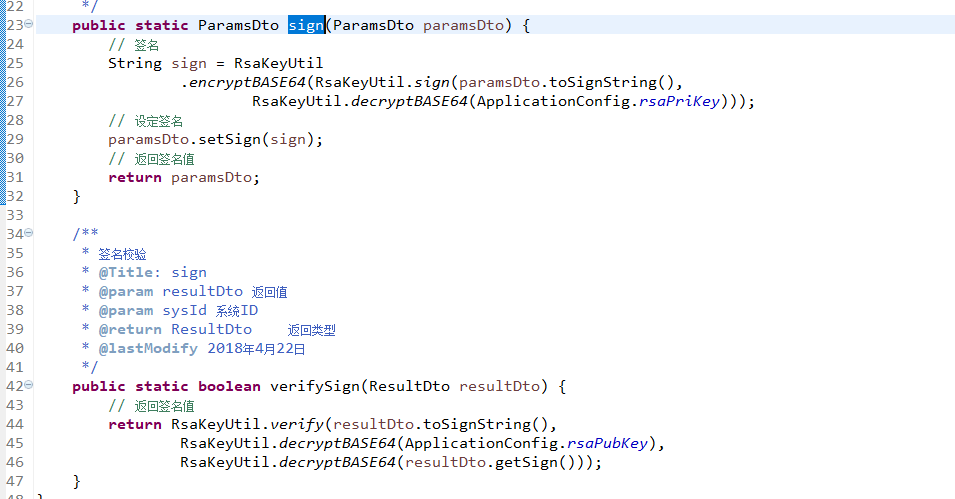
<http://zhuoyaopingzi.iteye.com/blog/1992205>

## 关于转账的二次签名

审核员进行转账操作时，需要审核员使用U盾中保存的私钥在页面上对转账内容进行签名，签名对象为转账对象序列化后的数据。然后把签名前数据与前面作为一个对象传递到后台，后台把这个对象作为请求参数params，封装过后用业务平台私钥进行二次签名，把请求参数发送到钱包服务平台。如此即完成了二次签名。

签名代码可参照如下：





发送请求代码如下：



