Wallet 钱包接口文档

v1.0

目录

概览

版本号

1.0

更新历史

更新时间	版本	更新说明
2018-4-16	1.0	新建。

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
nonce	string	是	随机数
timestamp	string	是	时间戳 如：1524146014086

示例

```
{
   "params" : {
      "sysId": "1111111",
      "txId": "12345678",
```

```
        "id": "12345678"
    },
    "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	result 数据或者 error 数据签名
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",
                "sysId": "10001"
        },
        "errCode": -1,
        "errMsg": "",
        "flag": true,
```

```
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

或者

```
{
        "errCode": 10001,
        "errMsg": "签名校验失败",
        "flag": false,
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
} }
```

错误码(code)

错误码	说明
10001	签名校验失败
10002	交易不存在
10003	余额不足，交易失败
10004	签名不正确，交易失败
…	

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 params 值末尾拼接随机数与时间戳，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 data 值末尾拼接 flag、随机数与时间戳，然后使用 sha1 进行摘要算法，得到字符串 stringA。

错误的场合对返回报文中的 errCode+errMsg+flag+随机数+时间戳，然后使用 sha1 进行摘要算法，得到字符串 stringA。

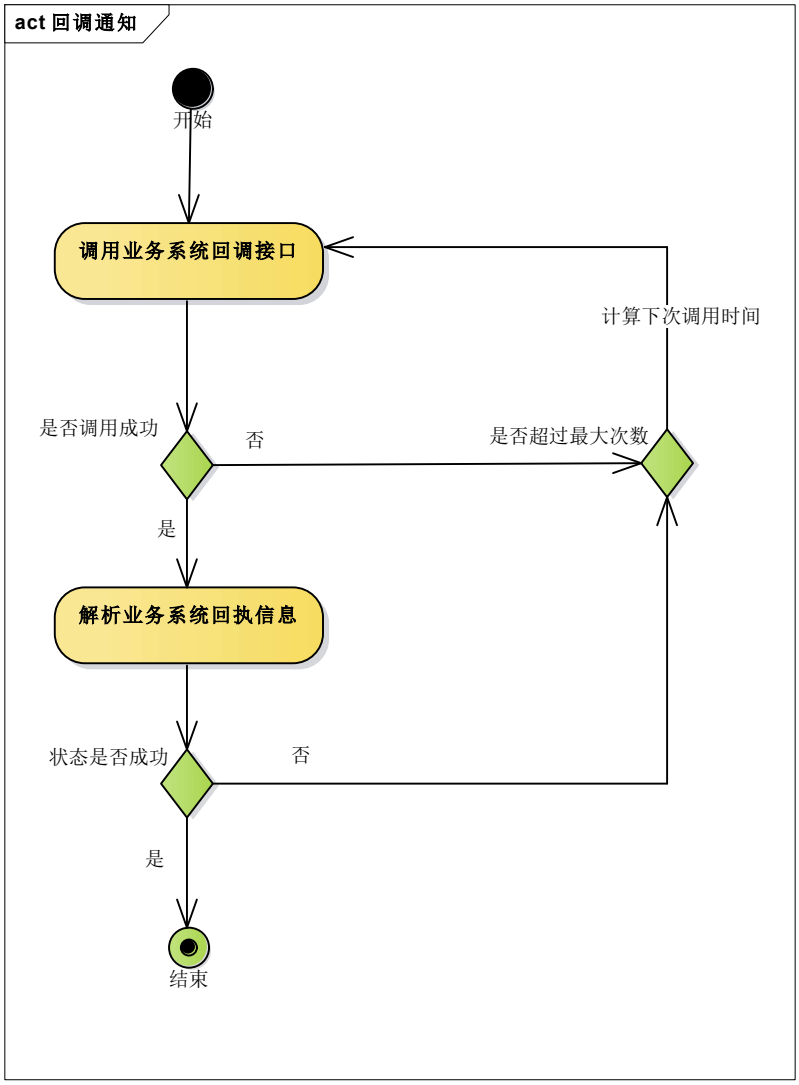
第二步，使用 RSA 私钥对 stringA 进行签名，得到字符串 stringB，具体参见附录签名及验签示例。

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：调用随机数函数生成，将得到的值转换为字符串。

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
sysId	string	是	平台分配的应用标识。

示例

```
{
        "params": {
                "sysId": "1111111"
        },
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

响应参数

属性	类型	是否必须	描述
sysId	string	是	平台分配的应用标识。
address	string	是	用户帐号绑定的地址

示例

```
{
        "data": {
                "address": "e1e9f04de623f676047c0650b36906b495ed7dc6",
                "sysId": "10001"
        },
```

```
        "errCode": -1,
        "errMsg": "",
        "flag": true,
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

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
sysId	string	是	平台分配的应用标识。
addresses	list	是	需要查询帐号的列表

示例

```
{
        "params": {
                "addresses": ["e1e9f04de623f676047c0650b36906b495ed7dc6",
                        "e2e9f04de623f676047c0650b36906b495ed7dc6",
                        "e2e9f04de623f676047c0650b369222495ed7dc6"
                ],
                "sysId": "10001"
        },
        "sign": "1234567890abcdef",
```

```
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

响应参数

属性	类型	是否必须	描述
sysId	string	是	平台分配的应用标识。
Map	map	是	各帐号余额情况 key:aaddress value :List< balanceInfo>

详细说明：

balanceInfo

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

示例

```
{
        "data": {
                "balanceDetails": {
                        "e1e9f04de623f676047c0650b36906b495ed7dc6": [{
                                "amount": 20000.02000000,
                                "currency": "HNB"
                        },
                        {
                                "amount": 2011.02000000,
                                "currency": "ETH"
                        }],
                        "ffasdffdsdfasdae2312tead89089pjgiaodjfaj": [{
                                "amount": 20000.02000000,
                                "currency": "HNB"
                        },
                        {
                                "amount": 2011.02000000,
```

```
                                    "currency": "ETH"
                            }]
                    },
                    "sysId": "10001"
            },
            "errCode": -1,
            "errMsg": "",
            "flag": true
}
```

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
sysId	string	是	平台分配的应用标识。
transactions	list	是	转账信息集合

详细说明：

transactionDetailDto

amount	decimal	是	交易金额
currency	string	是	币种：eth、hnb 等
from	string	是	转出地址

to	string	是	转入地址
id	string	是	业务系统交易唯一标识

示例

```
{
        "params": {
                "sysId": "10001",
                "transactions": [{
                        "amount": 20.10000000,
                        "currency": "hnb",
                        "from": "e1e9f04de223f676047c0650b36906b495ed7dc6",
                        "id": "5969ecfef4664a73aa8cabcfce331c7b",
                        "to": "e1e9f04de223f676047c0650b36906b495ed7dc6"
                }, {
                        "amount": 10.10000000,
                        "currency": "eth",
                        "from": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                        "id": "5969ecfef4664a73448cabcfce331c7b",
                        "to": "e1e9f04de223f676047c0220b36906b495ed7dc6"
                }]
        },
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

响应参数

属性	类型	是否必须	描述
sysId	string	是	平台分配的应用标识。
receipts	list	是	交易回执

详细说明：

receipt

orderNo	string	是	交易流水号
id	string	是	业务系统交易唯一标识

示例

```
{
        "data": {
                "receipts": [{
                        "id": "adjf123ioh",
                        "orderNo": "0xD1c82c71cC567d63Fd53D5B91dcAC6156E5B96B3"
                },
                {
                        "id": "adjf123ioh",
                        "orderNo": "0xD1c82c71cC567d63Fd53D5B91dcAC6156E5B96B3"
                }],
                "sysId ": "NZG2T7ME"
        },
        "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
sysId	string	是	平台分配的应用标识。
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": {
                "sysId": "1111111",
                "txid": "123581254",
                "orderNo": "0x123581254"
        },
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

响应参数

属性	类型	是否必须	描述
sysId	string	是	平台分配的应用标识。
amount	decimal	是	交易金额
currency	string	是	币种：eth、hnb 等
from	string	否	转出地址（为空的场合，表示提现）

属性	类型	是否必须	描述
to	string	是	转入地址
id	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": {
                 "sysId": "1111111",
                 "crateTime": "2018-04-19 23:43:43 643",
                 "from": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                 "id": "5969ecfef4664a73448cabcfce331c7b",
                 "to": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                 "amount": 20.10000000,
                 "tradeBlokNo": 54721231,
                 "tradeFee": 0.20000000,
                 "orderNo": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                 "tradeStatus": "02",
                 "currency": "hnb",
                 "txId": "e1e9f0444423f676047c0650b36906b495ed7dc6"
        },
        "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
id	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": {
                "sysId": "1111111",
                "id": "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
sysId	string	是	平台分配的应用标识。
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": {
                "sysId": "1111111",
                "status": true
        },
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

或者

```
{
        "params": {
                "sysId": "1111111",
                "status": false,
                "msg": "订单不存在"
        },
        "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
sysId	string	是	平台分配的应用标识。
to	string	是	转入地址
id	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

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
from	string	否	转出地址（为空的场合，表示提现）

示例


```
{
        "data": {
                "sysId": "1111111",
                "crateTime": "2018-04-19 23:43:43 643",
                "from": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                "address": "e1e9f0444423f676047c0650b36906b495ed7dc6"
                "id": "5969ecfef4664a73448cabcfce331c7b",
                "to": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                "tradeBlokNo": 54721231,
                "tradeFee": 0.20000000,
                "orderNo": "e1e9f0444423f676047c0650b36906b495ed7dc6",
                "tradeStatus": "02",
                "currency": "hnb",
                "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
sysId	string	是	平台分配的应用标识。
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": {
                "sysId": "1111111",
                "status": true
        },
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

或者

```
{
        "params": {
                "sysId": "1111111",
                "status": false,
                "msg": "订单不存在"
        },
        "sign": "1234567890abcdef",
        "nonce": "c49d982e211e423eb4038853227cb81b",
        "timestamp": "1524146014086"
}
```

附录

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