HPBWallet 协议文档

V1.0.1 更新：2019.4.26

简介

场景一：钱包 APP 内嵌 dapp 的 H5 页面，进行登录和支付。

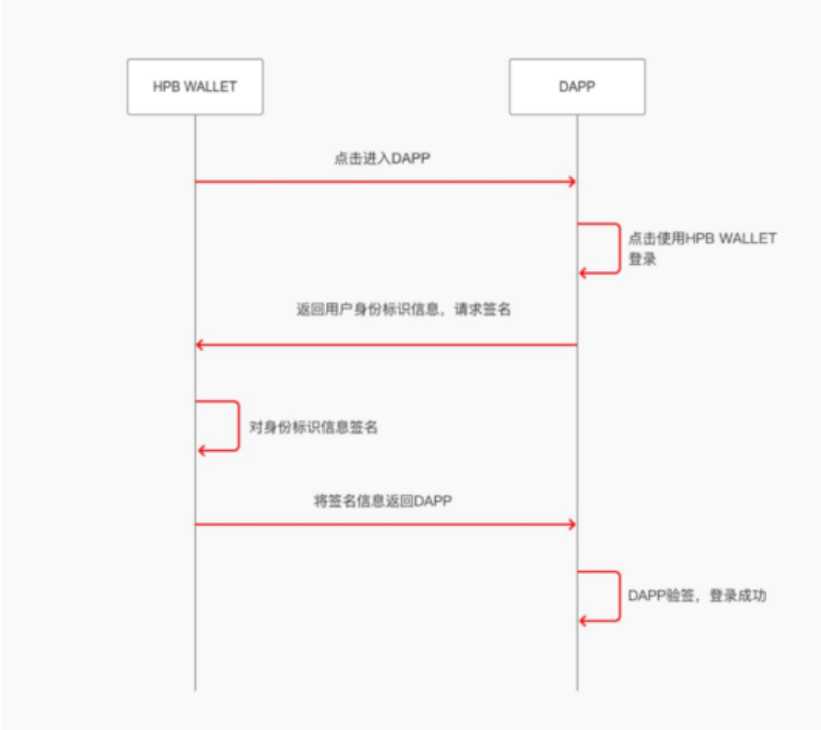
场景二：钱包 App 扫二维码进行登录和支付，适用于 Web 版 dapp。

场景一

登录

场景：在钱包内打开 H5,应用内进行用户登陆信息验证。

业务流程图：



- 我们会提供一个 js 中间件，提供 js 和 HPBWallet 的交互，便于 DApp 的 h5 调用。

- DApp 的 h5 调用 `signToLogin(params)` 方法，和 js 中间件交互，发起登录。
- dapp 需要传递的参数(json 格式)说明：

```
{
    protocol    string   // 协议名，钱包用来区分不同协议，本协议为
HPBWallet
    version     string   // 协议版本信息，如 1.0
    blockchain  string   // 公链标识（HPB）
    dappName    string   // dapp 名字
    dappIcon    string   // dapp 图标
    action      string   // 赋值为 login
    uuID        string   // dapp server 生成的，用于此次登录验证的唯
一标识
    expired   string           // 过期时间，unix 时间戳
    loginMemo     string      // 登录备注信息，钱包用来展示，可选
}
```

- 钱包对登录相关数据进行签名

```
 // 生成 sign 算法
let data = timestamp + account + uuID + ref     //ref 为钱包名，标示
来源
sign = ecc.sign(data, privateKey)
```

- HPBWallet 签名后，回传给 Dapp 的参数(json 格式)说明:

```
// 请求登录验证的数据格式
{
    protocol   string     // 协议名，钱包用来区分不同协议，本协议为
HPBWallet
    version    string     // 协议版本信息，如 1.0
    blockchain string    // 公链标识（HPB）
    timestamp  string     // 当前 UNIX 时间戳
    sign       string     // ECC 签名
    action      string   // 赋值为 login
    uuID       string     // dapp server 生成的，用于此次登录验证的唯
一标识
    account    string     // HPB 地址
    ref        string     // 来源,标识来自于 HPBWallet，可以设置成 HPB
钱包名称
}
```

- DApp 收到数据，实现 `function getCallback(params)` 接收原生回调传值 action 字段，判断此次行为，然后可以选择两种方式验签。 方式一：dapp 本地验签（Demo 提

供），方式二：dapp 发送到 dapp sever 验证 sign 签名数据，返回结果 code 给 dapp 判断成功状态。

附录：授权登录 JS Demo 代码

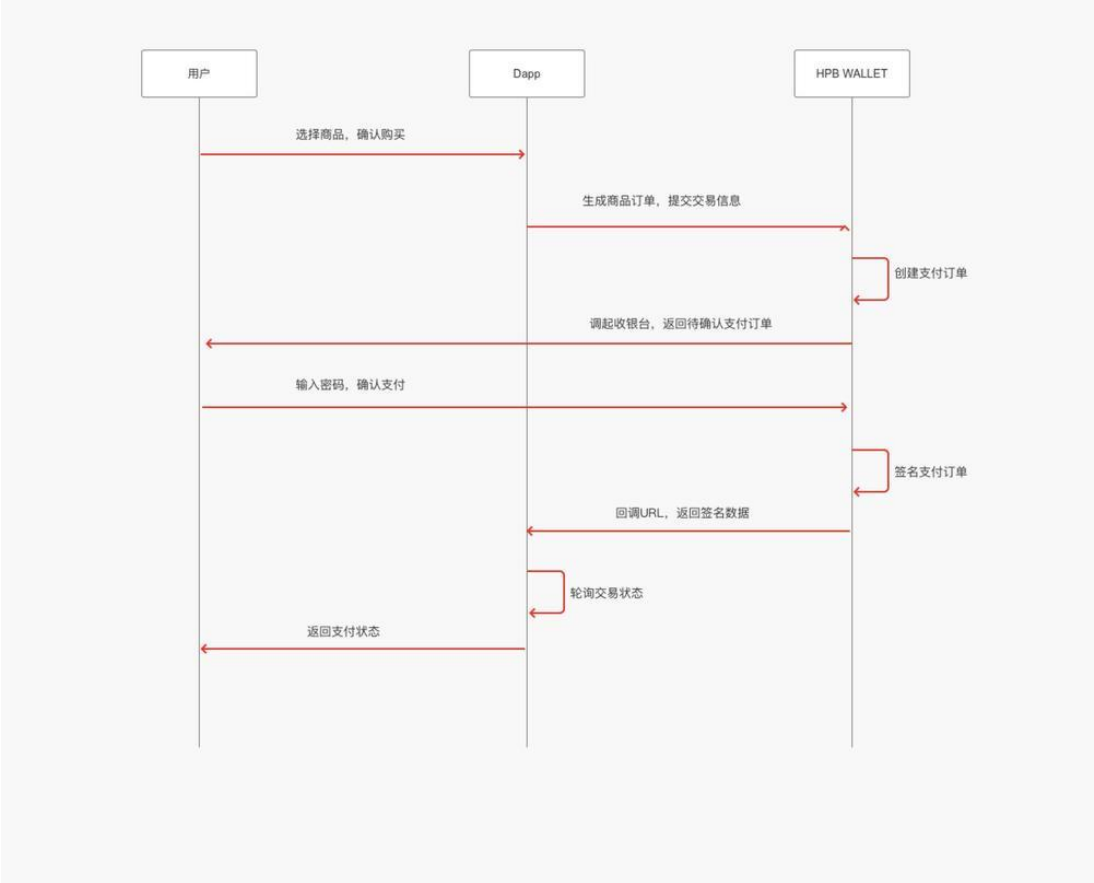
```
Step1
引入 js 依赖包

<script type="text/javascript" src="compiled_lib.js"> </script>
<script type="text/javascript" src="sign.js"></script>
<script src="./hpb_sdk.js"></script>

Step2
 //授权登陆调用 login 方法
    function login() {
            var loginParams = {
                protocol: 'HPBWallet',
                version: '1.0',
                blockchain: 'HPB',
                dappName: 'HPB dapp',
                dappIcon: 'HPB icon',
                action: 'login',
                uuID: '123',
                expired: '123456',
                loginMemo: 'test'
            }
            //./hpb_sdk.js 文件再 window 下添加了 hpbweb3 对象。调用对象的 login 函数，login 函数调用客户端 signToLogin 方法进行交互
            //  login 函数有两个参数。第一个参数是需要传给客户端的参数对象。第二个参数是客户端调用 getCallback 执行的函数。
            window.hpbweb3.login(loginParams, function (params) {
                var sign = params.sign;
                var msg = params.timestamp + params.account + params.uuID + params.ref;
                var signRet = signRetFromHex(sign);
                var pubKeyBuffer = recovery(signRet.r, signRet.s, signRet.v, msg);
                var address = generateAddress(pubKeyBuffer);
                if (address == params.account) {
                    var isValid = verify(msg, signRet.r, signRet.s, pubKeyBuffer);
                    if (isValid) {
                        //登陆成功
```

```
                        document.getElementById('result').innerHTML =
'登陆成功'
                    } else {
                        //登陆失败
                        document.getElementById('result').innerHTML =
'登陆失败'
                    }
                } else {
                    //验签失败
                    document.getElementById('result').innerHTML =
'address 不一样'
                }
            })
    }
```

支付

场景：在钱包内打开 H5,应用内进行支付。

业务流程图：



- DApp 的 h5 调用 `startToPay(params)` 方法，和 js 中间件交互，发起支付。
- dapp 需要传递的参数(json 格式)说明：

```
{
    protocol    string   // 协议名，钱包用来区分不同协议，本协议为
HPBWallet
    version     string   // 协议版本信息，如 1.0
    blockchain  string   // 公链标识（HPB）
    isSend      bool     // 是否需要 HPBWallet 发送签名到主网(默认
YES)，YES 为需要，NO 为不需要，可选
    dappName    string   // dapp 名字，用于在钱包 APP 中展示，可选
    dappIcon    string   // dapp 图标 Url，用于在钱包 APP 中展示，可选
    action      string   // 支付时，赋值为 pay，必须
    to          string   // 收款人的 hpb 账号，必须
    amount      string   // 转账数量(1 HPB 就是 1，0.05 HPB 就为
0.05)，必须
    desc        string    // 交易的说明信息，钱包在付款 UI 展示给用
户，最长不要超过 128 个字节，可选
    expired     string    // 交易过期时间，unix 时间戳
    notifyUrl    string    //dapp sever 通知地址 可选
    orderId      string    // dapp 订单唯一编号  必须
}
```

- 钱包接收到数据后（此步骤会判断支付白名单），生成一笔 HPB 的转账，提交到 HPB 主网。

- 钱包回传给 DAPP 的参数说明（方式二）：

```
    result      string   //0 为用户取消，1 为成功，  2 为失败
    protocol    string   // 协议名，钱包用来区分不同协议，本协议为
HPBWallet，必须
    version     string   // 协议版本信息，如 1.0，必须
    orderId      string    // dapp 订单唯一编号  必须
    blockchain  string   // 公链标识（HPB），必须
    action      string   // 支付时，赋值为 pay，必须
    txID        string   //交易哈希，可选（isSend 为 YES 时必须）
    signature   string   //交易签名，可选（isSend 为 NO 时必须）
    amount      string   // 转账数量(1 HPB 就是 1，0.05 HPB 就为
0.05)，必须
```

- 方式一：如果 notifyUrl 不为空，则钱包 post 数据到 notifyUrl 上，有 dapp 服务器后续处理，获取支付状态。

- 方式二：notifyUrl 为空，DApp 实现 `function getCallback(params)` 通过 action 字段，判断为支付行为后，可以拿到签名数据 signature 或者交易哈希 txID,通过交易哈希轮询查询交易，获取支付状态。

附录：支付 JS Demo 代码

```
Step1
引入 js 依赖包

<script type="text/javascript" src="./web3_hpb.min.js"></script>
<script src="./hpb_sdk.js"></script>

Step2

var Web3ForHpb = require('web3_hpb');
    var web3Hpb = new Web3ForHpb();
    web3Hpb.setProvider(new
web3Hpb.providers.HttpProvider("http://192.168.0.121:8080/nodeurl"));

    function pay() {
            var payParams = {
                protocol: 'HPBWallet',
                version: '1.0',
                blockchain: 'HPB',
                dappName: 'HPB dapp',
                dappIcon: 'HPB icon',
                action: 'pay',
                to: '0xed37f755e56b1d49642dce8ff2b788ae33263c94',
                amount: '0.1',
                precision: '122.22',
                desc: 'desc',
                expired: '123131expired2321'
            }
            //获取交易状态函数
            function getStatus(id) {
                var txReceipt =
web3Hpb.hpb.getTransactionReceipt('0xf236ab148ed00884611ece976b2ebb43
77cb33ddfefde58bf6a6ac4ab8aee067'); //参数 id
                if (txReceipt.status === '0x1') {
                    document.getElementById('result').innerHTML =
'pay success'
                    document.getElementById('loading').style.display
= 'none'
```

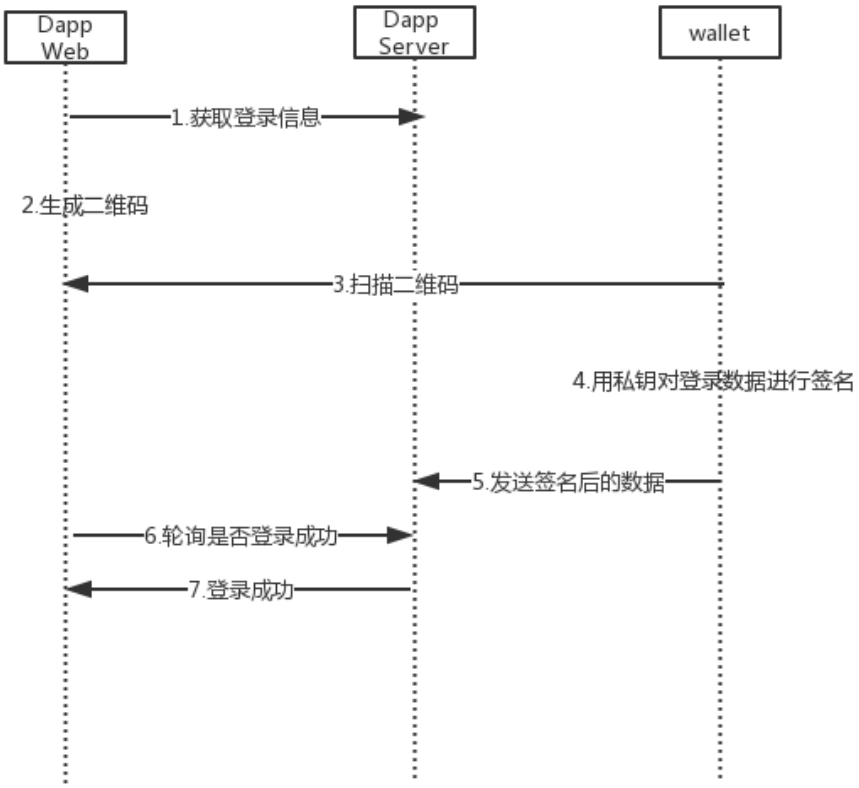
```
                    clearInterval(loop)
                }
            }
            //./hpb_sdk.js 文件再 window 下添加了 hpbweb3 对象。调用对象的 pay 函数，pay 函数调用客户端 startToPay 方法进行交互
            //  pay 函数有两个参数。第一个参数是需要传给客户端的参数对象。第二个参数是客户端调用 getCallback 执行的函数(⚠如果选择 notifyUrl 回调方式，可以忽略客户端 getCallback 回调)。
            window.hpbweb3.pay(payParams, function (params) {
                //获取 txID
                if (params && params.txID) {
                    var txID = params.txID;
                    //循环查询交易状态
                    var loop = setInterval(function ()
{ getStatus(txID) }, 1000);
                } else {
                    document.getElementById('result').innerHTML =
'pay error'
                }
            })
        }
```

场景二

登录

场景：使用钱包扫描 web 网站上二维码登录

业务流程图：



- web 的 dapp 生成二维码，HPBWallet 扫描登录二维码，包含以下数据：
- dapp 生成二维码所需参数(json 格式)说明：

```
{
    protocol    string   // 协议名，钱包用来区分不同协议，本协议为
HPBWallet
    version     string   // 协议版本信息，如 1.0
    blockchain  string   // 公链标识（HPB）
    dappName    string   // dapp 名字
    dappIcon    string   // dapp 图标
    action      string   // 赋值为 login
    uuID        string   // dapp server 生成的，用于此次登录验证的唯
一标识
    expired     string      // 二维码过期时间，unix 时间戳
    loginUrl    string   // dapp server 上用于接受登录验证信息的 url
    loginMemo   string      // 登录备注信息，钱包用来展示，可选
}
```

- 网站生成二维码包含的数据格式为：

```
https://www.hpb.io/client?login={
    "protocol": "HPBWallet",
```

```
    "version": "1.0.0",
    "blockchain": "HPB",
    "dappName": "name",
    "dappIcon":
"http://wx4.sinaimg.cn/large/a1b61d0aly1fn2h3xwat6j20dw0dwtbp.jpg",
    "action": "login",
    "uuID": "dwdwwvvcscacxa222",
    "expired": "1547793880",
    "loginUrl": "https://……",
    "loginMemo": "使用你的账号信息（钱包昵称，钱使用你的账号信息（钱
包昵称，钱包地址）登录该应用。"
}
```

- HPBWallet 签名后，post 数据到 Dapp 提供的 `loginUrl`.参数(json 格式)说明如下:

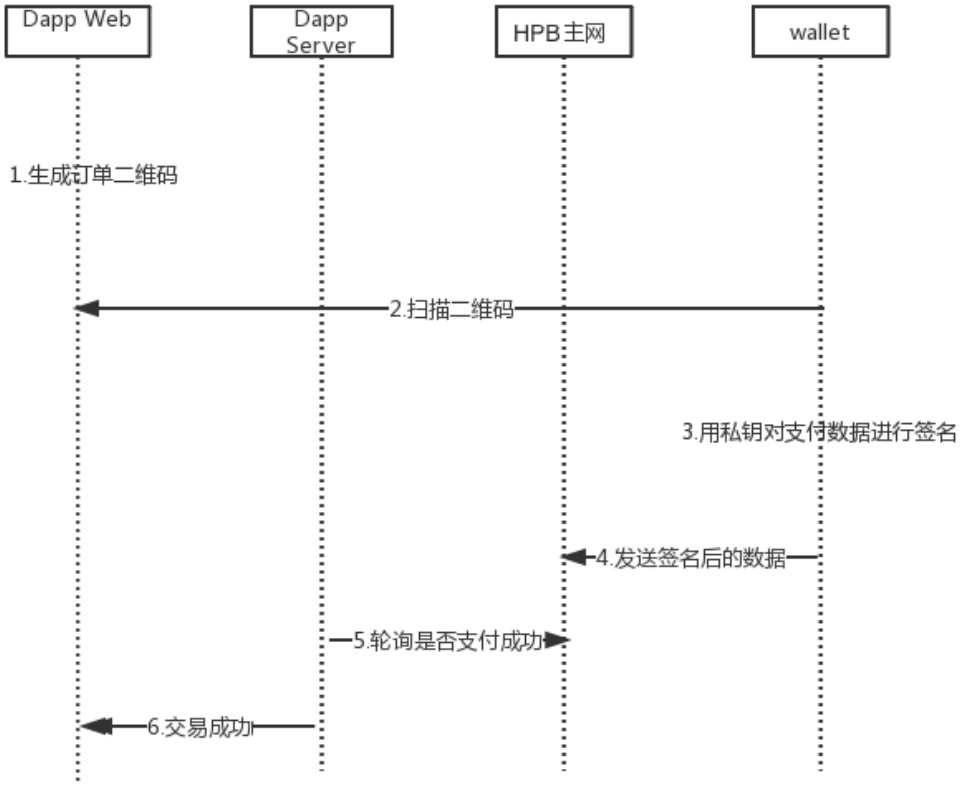
```
// 请求登录验证的数据格式
{
    protocol   string     // 协议名，钱包用来区分不同协议，本协议为
HPBWallet
    version    string     // 协议版本信息，如 1.0
    blockchain string    // 公链标识（HPB）
    timestamp  string     // 当前 UNIX 时间戳
    sign       string     // ECC 签名
    action      string   // 赋值为 login
    uuID       string     // dapp server 生成的，用于此次登录验证的唯
一标识
    account    string     // HPB 地址
    ref        string     // 来源,标识来自于 HPBWallet，可以设置成 HPB
钱包名称
}
```

- dapp server 收到数据，验证 sign 签名数据，并返回结果 code；若验证成功，则在 dapp 的业务逻辑中，将该用户设为已登录状态。

支付

场景：钱包扫描网站二维码进行支付

业务流程图：



- web 的 dapp 生成支付二维码，HPBWallet 扫描二维码支付(此处 HPBWallet 要判断支付地址白名单)

- dapp 生成支付二维码需要传递的参数(json 格式)说明：

```
{
    protocol    string   // 协议名，钱包用来区分不同协议，本协议为 HPBWallet
    version     string   // 协议版本信息，如 1.0
    blockchain  string   // 公链标识（HPB）
    isSend      bool     // 是否需要 HPBWallet 发送签名到主网(默认 YES)，YES 为需要，NO 为不需要，可选
    dappName    string   // dapp 名字，用于在钱包 APP 中展示，可选
    dappIcon    string   // dapp 图标 Url，用于在钱包 APP 中展示，可选
    action      string   // 支付时，赋值为 pay，必须
    to          string   // 收款人的 hpb 账号必须
    amount      string   // 转账数量(1 HPB 就是 1, 0.05 HPB 就为 0.05)，必须
    desc        string    // 交易的说明信息，钱包在付款 UI 展示给用户，最长不要超过 128 个字节，可选
    expired     string    // 交易过期时间，unix 时间戳
```

```
  notifyUrl      string   //dapp Sever 地址，为了接收 HPBWallet 交易签名数据或交易哈希
    orderId      string    // dapp 订单唯一编号  必须
}
```

- 网站生成二维码包含的数据格式为：

```
https://www.hpb.io/client?pay={
    "protocol": "HPBWallet",
    "version": "1.0.0",
      ...........
    }
```

- HPBWallet 签名后，post 数据到 Dapp 提供的 `notifyUrl`.参数(json 格式)说明如下:

```
    result      string   //0 为用户取消，1 为成功，  2 为失败
    protocol   string   // 协议名，钱包用来区分不同协议，本协议为 HPBWallet，必须
    version     string   // 协议版本信息，如 1.0，必须
    blockchain  string   // 公链标识（HPB），必须
    action      string   // 支付时，赋值为 pay，必须
    txID        string   //交易哈希，可选（isSend 为 YES 时必须）
    signature   string   //交易签名，可选（isSend 为 NO 时必须）
    amount      string   //转账数量(1 HPB 就是 1，0.05 HPB 就为 0.05)，必须
    orderId      string    // dapp 订单唯一编号  必须
```

- dapp sever 轮询查询交易状态，并返回 dapp 判断支付状态。