授权登录接口文档V0.1

1. 说明

本文档指在用javascript实现前端的授权登录，文中所有代码都是以javascript为开发语言；如果接入程序是基于nodejs的框架，也可以在后端完成。

1. 请求用户授权
2. 请求地址：<https://wallet.yoyow.org/#/authorize-service>
3. 请求参数表：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| platform | string | 是 | 平台id,用于识别请求发出方 |
| time | string | 是 | 时间戳，JavaScript的当前毫秒数 |
| which | string | 否 | 请求功能，默认为Login，目前只有Login |
| sign | string | 是 | 签名值，用于钱包校验平台请求的有效性 |
| state | string | 是 | 平台页面状态url |
| redirect | string | 否 | 平台接收钱包回调接口url，用于处理登录状态；如果不提供此参数，钱包会把授权信息带到state的后面；此url的域名必须与state的url域名一致。（目前没有其他语言的sdk所以暂时不支持其他语言在后端验证） |

1. 请求事例：

**var** oKey = PrivateKey.fromWif(**"平台所有者secondary私钥"**);  
**let** sendObj = {  
 **"platform"**: **"**247731382**"**,  
 **"time"**: **""** + (**new** Date()).getTime()  
}  
**var** signed = Signature.signBuffer(**new** Buffer(***JSON***.stringify(sendObj)), oKey);

**var** url = **`http://demo.yoyow.org:8000/#/authorize-service?platform=**${sendObj.**platform**}**&time=**${sendObj.**time**}**&sign=**${signed.toHex()}**&state=http://localhost/`**;

**window**.open(url);

其中“oKey”为平台所有者的零钱私钥，“sendObj”为签名输入对象

1. 接收用户授权
2. 参数表：

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| yoyow | String | YOYOW用户ID,指此用户同意授权 |
| time | String | 时间戳，JavaScript的当前毫秒数 |
| sign | String | 签名值，用于平台校验用户授权的有效性 |
| state | String | 如果请求时有 redirect和state参数时，则state原样传回 |

1. 校验签名事例

**var** sObj = {  
 **yoyow**: yoyow,  
 **time**: time  
};

**let** oPkey = PublicKey.fromPublicKeyString(secondary);  
**let** verify = Signature.fromHex(sign).verifyBuffer(**new** Buffer(***JSON***.stringify(sObj)), oPkey);  
**if** (verify) {  
 //用户有效授权

}else{

//用户无效授权

}

其中“secondary”为参数yoyow用户的零钱公钥