## 说明

目前楼兰平台有四条链，零售正式链、零售测试链、影视正式链、影视测试链。每条链的url不同，以影视测试链为例，url为：

https://www.lianlianchains.com/**mogaotest**/XXX

四条链的url仅红色字体部分不一样，使用相应的联盟链时，需要替换为：

零售正式链：retail

零售测试链：retailtest

影视正式链：mogao

影视测试链：mogaotest

下面的接口文档以mogaotest为例进行说明。

示例中提供了三个接口：

1、用户注册开户接口(见接口文档章节)。

该接口在平台联盟链注册一个用户，并为该用户开一个账户，账户中主要记录权限控制信息（如用户平台身份，用户公钥hash值）， 后续事物执行及查询会根据权限控制信息做身份校验。

2、事务执行接口(见接口文档章节)。

该接口执行某些增删改区块链账本的操作，并将这些修改持久化。

3、查询接口(见接口文档章节)。

该接口查询链上的数据。注意，如果在查询接口中执行了修改区块链账本的操作，这些操作不会生效。

签名机制说明：

楼兰平台使用签名机制来验证用户的身份。即在客户端生成用户的一对公私钥，调用开户接口时，将公钥hash作为参数保存在链中；后续调用执行或查询接口时，对请求参数使用私钥做签名，然后在区块链中对签名做校验，验证通过才允许执行后续操作。

平台目前采用的签名算法是secp256k1，公钥hash算法为hash160（sha256和ripemd160两次hash：ripemd160（sha256（pubkey）））。

示例（nodejs伪代码）：

1、生成公私钥

const secp256k1 = require('secp256k1/elliptic');

const crypto = require('crypto')

let privKey

do {

privKey = crypto.randomBytes(32)

} while (!secp256k1.privateKeyVerify(privKey))

const pubKey = secp256k1.publicKeyCreate(privKey)

调用开户接口时，将pubKey做hash160并转为base64编码作为一个参数传入。

crypto.createHash('ripemd160').update(crypto.createHash('sha256').update(pubKey).digest()).digest('base64')

2、签名

三个接口都有这两个参数：fcn和args

先生成参数的摘要：

var allArgs = fcn + ',' + args.join(',')

var msg = crypto.createHash('sha256').update(allArgs).digest()

然后对摘要做签名：

var sign = secp256k1.sign(msg, privKey)

再将该签名对象转换为区块链可识别的格式，并转为base64编码，调用接口时传入

Buffer.concat([sign.signature, new Buffer([sign.recovery])]) .toString('base64')

结果验证：

私钥hex："1fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff"

公钥hex："02a6c36f0904a863815b366120df8536f0c1477d9213fcc2421a0efa4eae64abf8"

参数摘要hex："1111111111111111111111111111111111111111111111111111111111111111"

公钥hash160的base64结果："QxD6f7YhR/FjKAk70/BhP+ljR58="

签名的base64结果：

" J55xiSqbx8zIKVbrBGmkD9xkTb44VvqNhWjLT8/hesgHJRcruZNsU7YjP1xUtvnHZaR9bV47QnSmvd4G4JJS0QE="

如果使用上例中的私钥、公钥、参数摘要，得出的公钥hash160的base64结果以及签名的base64结果和上例中一致，说明就可以通过平台的验证。

## 接口文档

### 用户注册并开户示例

**参数说明**

fcn：函数名，account表示新开账户。

usr：待注册的用户名。

ccname：chainCode名称。

args：参数列表, 如果是GET方式，下例可以这样请求（多个arg用','分隔）https://www.lianlianchains.com/mogaotest/register&fcn=account&args=arg1,arg2,arg3；下面的示例使用POST请求，数据使用JSON格式，后续示例和此例相同，不再说明。

第一个参数为用户名，第二个参数为账户名（可以和用户名保持一致），第三个参数为公钥的hash160结果并转为base64格式（方法参考第一章《说明》）。

signature: 对参数做的签名（方法参考第一章《说明》）

**调用示例**

请求URL：

|  |
| --- |
| https://www.lianlianchains.com/mogaotest/register |

请求数据：

|  |
| --- |
| {  "fcn": "account",  "usr": "userName",  "ccname": "chainCodeName",  "args": ["userName", "accountName", "pubkeyHashBase64"],  "signature": "signatureBase64"  } |

响应正文：

|  |
| --- |
| {  "code": 0,  "msg": "OK",  "result":{  "payload":""  "txid":" 80b550099289d5a9e362e3266e088309...."  }  } |

响应说明：

code：返回码，0表示成功，非0表示失败。

msg：返回信息，错误时会有错误信息返回。

result: payload为链码返回的数据，txid为本次执行的交易id。

### 执行示例

**参数说明**

fcn：函数名。

usr：待注册的用户名。

ccname：chainCode名称。

args：参数列表。

第一个参数为用户名，第二个参数为账户名（可以和用户名保持一致），第三个参数为写入的值，随便输入。

signature: 对参数做的签名（方法参考第一章《说明》）

**调用示例**

请求URL：

|  |
| --- |
| https://www.lianlianchains.com/mogaotest/invoke |

请求数据：

|  |
| --- |
| {  "fcn": "writeXXX",  "usr": "userName",  "ccname": "chainCodeName",  "args": ["userName", "accountName", "arg3"],  "signature": "signatureBase64"  } |

说明：如果是在楼兰网站的开放平台中在某个合约上点击执行和查询（如下图），则不需要输入usr、ccname和signature这三个参数，平台会自动添加这几个参数，并且不会验证签名。



响应正文：

|  |
| --- |
| {  "code": 0,  "msg": "OK",  "result":{  "payload":""  "txid":" 80b550099289d5a9e362e3266e088309...."  }  } |

响应说明：

code：返回码，0表示成功，非0表示失败。

msg：返回信息，错误时会有错误信息返回。

result: payload为链码返回的执行结果，txid为本次执行的交易id。

### 查询示例

**参数说明**

fcn：函数名。

usr：待注册的用户名。

ccname：chainCode名称。

args：参数列表。

第一个参数为用户名，第二个参数为账户名（可以和用户名保持一致）。

signature: 对参数做的签名（方法参考第一章《说明》）

**调用示例**

请求URL：

|  |
| --- |
| https://www.lianlianchains.com/mogaotest/query |

请求数据：

|  |
| --- |
| {  "fcn": "queryXXX",  "usr": "userName",  "ccname": "chainCodeName",  "args": ["userName", "accountName"],  "signature": "signatureBase64"  } |

说明：如果是在楼兰网站的开放平台中在某个合约上点击执行和查询，则不需要输入usr、ccname和signature这三个参数，平台会自动添加这几个参数，并且不会验证签名。

响应正文：

|  |
| --- |
| {  "code": 0,  "msg": "OK",  "result": ""  } |

响应说明：

code：返回码，0表示成功，非0表示失败。

msg：返回信息，错误时会有错误信息返回。

result:链码返回的查询数据。