

Management Service Orderme detailed design specificatioN

Management Service v0.1

Hugo 2018-01-23

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Version** | **Revised by** | **Time** | **Remarks** |
| 1 | v0.1 | Hugo | 2018-01-23 | 初版 |
| 2 | v0.2 | Hugo | 2018-03-06 | 添加了json格式等 |
|  |  |  |  |  |

目录

[1. Introduction 5](#_Toc508117748)

[2. Login/Logout 5](#_Toc508117749)

[2.1 登录序列图 5](#_Toc508117750)

[2.2 登出序列图 6](#_Toc508117751)

[2.3 登录/登出设计要点 6](#_Toc508117752)

[2.4 Login Request 7](#_Toc508117753)

[2.5 Login Response 7](#_Toc508117754)

[2.6 Logout Request 8](#_Toc508117755)

[2.7 Logout Response 9](#_Toc508117756)

[3. Change Password(login password & trade password) 9](#_Toc508117757)

[3.1 密码修改设计要点 9](#_Toc508117758)

[3.2 Change Password Request 9](#_Toc508117759)

[3.3 Change Password Response 10](#_Toc508117760)

[4. ReSet Password(后续做) 11](#_Toc508117761)

[4.1 密码重置设计要点 11](#_Toc508117762)

[4.2 消息定义 11](#_Toc508117763)

[5. Management Assets 12](#_Toc508117764)

[5.1 设计要点 12](#_Toc508117765)

[5.2 消息定义 12](#_Toc508117766)

[6. KYC 13](#_Toc508117767)

[6.1 设计要点 13](#_Toc508117768)

[6.2 消息定义 13](#_Toc508117769)

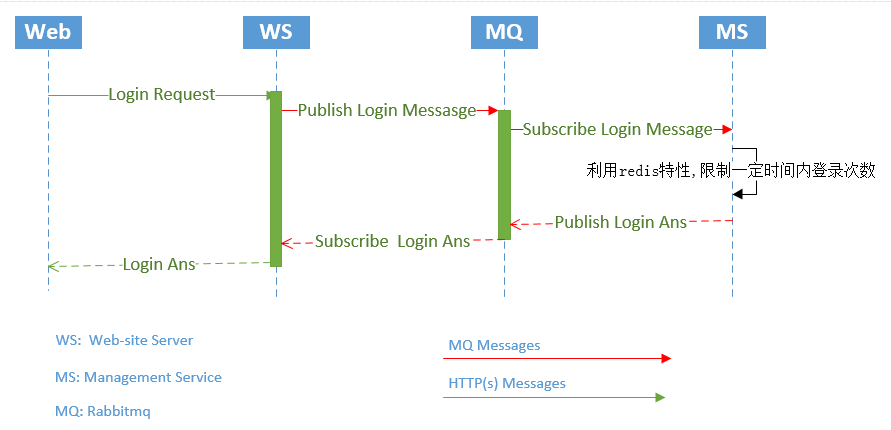
# Introduction

Management Service 是 Exchange 中的一个子模块，主要负责用户的

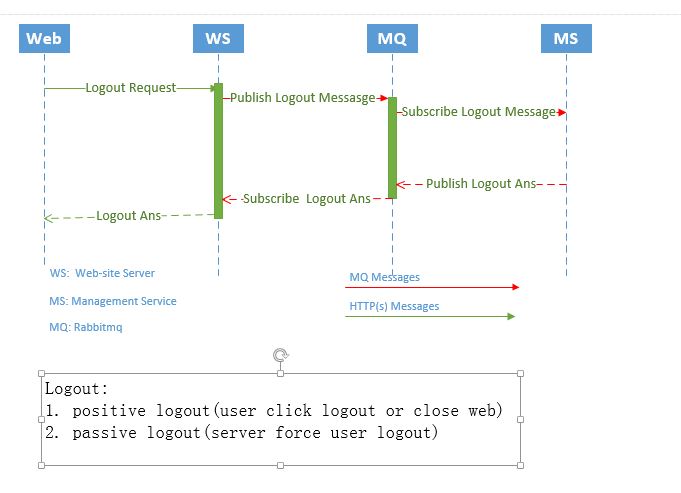
* 登录/登出
* 多次登录
* 密码修改
* 密码重置
* 交易码验证
* 资产管理如资产的查询、冻结、出入金等。
* KYC
* 开发初期密码为明文以方便调试，后期上线密码理应加密
* 每条消息不论是所发送的消息还是所接收的消息，都应实现异步定时定批插入数据库，以确保每条消息的持久化

# Login/Logout

## 登录序列图



## 登出序列图



## 登录/登出设计要点

* 每次登录服务端都会返回sessionid。
* 客户端收到这个sessionid 理应保存起来，后续的所有操作都应该带上该值。
* 当登录失败后，客户端应该主动关闭连接。
* 每个连接应该都有有效时间，当用户在有效时间内一直没有操作，那么应该断开此链接（该功能可以后续再做）。
* 消息体为json 格式。
* 一定时间内，登录次数有所限制，不能够频繁的发起登录请求(该功能后续再做)。
* 多次登录时，老的连接应该先关闭，然后重新建立新的连接。此处Management Service应该维护一个链表或者hash表，以User ID 为键，sessionid为value。
* 登录服务器Management Service应可负载均衡。
* 登出应该考虑ForceLogout，此逻辑由Management Service 管理的在线列表或者hash表实现(此处由redis的hash实现)

## Login Request

* Router Key：WS.LOGIN.WSID（Login: WS publish）
* Bind Key: WS.LOGIN.\*(login: MS subscribe),
* Queue Name: WS.LOGIN.MS（Login Ans: WS subscribe）

Json body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"username":"username",

"loginpassword":"password",

"srcip":"192.168.0.1",

"wsid":1011,

"logintime":123457 // long

}

## Login Response

* Router Key：WS.ANS.LOGIN.WSID（Login ans: MS publish）
* Bind Key: WS.ANS.LOGIN.WSID(login ans: WS subscribe),
* Queue Name: WS.ANS.LOGIN.WSID（Login Ans: WS subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1011,

"status":true,

"errmsg":"ok",

}

## Logout Request

* Router Key：WS.LOGOUT.WSID（Logout: WS publish）
* Bind Key: WS.LOGOUT.\*(logout: MS subscribe),
* Queue Name: WS.LOGOUT.MS（Logout Ans: WS subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1011,

"logouttime":123457 // long

}

## Logout Response

* Router Key：WS.ANS.LOGOUT.WSID（Logout ans: WS publish）
* Bind Key: WS.ANS.LOGOUT.WSID(Logout ans: WS subscribe),
* Queue Name: WS.ANS.LOGOUT.WSID（Logout Ans: WS subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1011,

"logouttime":123457 // long

"status":true,

"errmsg":"ok",

}

# Change Password(login password & trade password)

## 密码修改设计要点

* 密码修改时，需要把新老密码一并传过来
* 初期密码不加密，为明文，以方便调试
* Type：1-登录密码修改，2-交易密码修改

## Change Password Request

* Router Key：WS.CHANGEPWD.WSID（Change Password: WS publish）
* Bind Key: WS.CHANGEPWD.\*(Change Password: MS subscribe),
* Queue Name: WS.CHANGEPWD.MS（Change Password: WS subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1011,

"oldpassword":"sdgsdh",

"password":"sdgsdh234",

"type":1, //1-登录密码修改，2-交易密码修改

}

## Change Password Response

* Router Key：WS.ANS.CHANGEPWD.WSID（Change Password ans: WS publish）
* Bind Key: WS.ANS.CHANGEPWD.WSID(Change Password ans: MS subscribe),
* Queue Name: WS.ANS.CHANGEPWD.MS（Change Password Ans: WS subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1011,

"changetime":12346, // long

"status":true,

"errmsg":"ok"

}

# ReSet Password(后续做)

## 密码重置设计要点

* 用户名和注册邮箱重置密码
* 用户名和注册手机重置密码
* 为了安全考虑，还可以对注册时设置的问题进行校验（此处暂缺）
* 重置密码时校验邮箱或者手机号（甚至注册时的问题）后，向注册邮箱或者手机号下发新的密码，用户通过该密码重新登录并可再次修改密码
* meth-手机号/邮箱
* quest-注册时的密码校验，应为数组，此为可选项（此处暂缺）

## 消息定义

消息定义与change password 雷同，主要是type 字段不同

* Router Key：WS.CHANGEPWS.WSID（change password request-WS publish）, WS.CHANGEPWSANS.WSID（change password answer-MS publish）
* Bind Key: WS.CHANGEPWS.\*( change password request-MS subscribe), WS.LOGINANS.WSID（change password answer-WS subscribe）
* Body: reqid，type，method，userid，change\_time，sec\_websocket\_accept，wsid，[quest,…]（change password request）
* Body: reqid，type，userid，change\_time，sec\_websocket\_accept，wsid, status，errcode，errmsg（change password answer）

# Management Assets

## 设计要点

* 冻结(交易过程中理应冻结，由订单触发，此时ORDERSRV 向 MS 下发订单消息)

## Request

* Router Key：Order2Manage（ordersrv publish）
* Bind Key: Order2Manage (MS subscribe),
* Queue Name: Order2Manage\_Q（WS subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"tag":0x3003,

"userid":1234, // long

"orderid":"123456",

"symbol":,"LTC/BTC",

"ordertype":0,

"transtype":0,

"ordernumber":1.3223,

"price":12.2315,

"ordersrvid":1

}

## Response

* Router Key：Manage2Order.ordersrvid（MS publish）
* Bind Key: Manage2Order.ordersrvid. (ordersrv subscribe),
* Queue Name: Manage2Order\_Q（ordersrv subscribe）

Json Body:

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"tag":0x3004,

"userid":1234, // long

"orderid":"123456",

"errmsg":,"ok",

"status":1

}

# KYC

## 设计要点

* 有国内国外之分
* 国内只能够用passport
* 国外还支持其他证件类型，比如驾照
* Type：1-国内，2-国外
* id\_type： 1-passport，2-driver license
* image: 通过base64从二进制转为json object array。

## Request

* Router Key：WS.KYC.WSID（WS publish）
* Bind Key: WS.KYC.\* (MS subscribe),
* Queue Name: WS.KYC(MS subscribe）

Json

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1,

"country":,"china",

"firstname":,"name1",

"secondname":,"name2",

"idtype":1,

"idno":,"123456789x",

"image":,"oksldgsjlh",

}

## Response

* Router Key：WS.ANS.KYC.WSID（MS publish）
* Bind Key: WS.ANS.KYC.WSID (WS subscribe),
* Queue Name: WS.ANS.KYC.WSID(MS subscribe）

Json

{

"sessionid":"sdgsdg",

"reqid":"sdgsdh",

"userid":1234, // long

"wsid":1,

"status":true,

"errmsg":"ok",

}