HSBC作业方案

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **作者** | **修改时间** | **修改内容** | **变更原因** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 整体说明 2](#_Toc107478070)

[1.1 Controller层 2](#_Toc107478071)

[1.2 dao和model 2](#_Toc107478072)

[2 auth验证 3](#_Toc107478073)

[2.1 token生成 3](#_Toc107478074)

[2.2 Token身份验证 3](#_Toc107478075)

[2.3 Token注销 3](#_Toc107478076)

[3 用户 3](#_Toc107478077)

[4 角色 4](#_Toc107478078)

[5 用户角色关系 4](#_Toc107478079)

# 整体说明

为了避免重复造轮子，整个web项目使用JAVA SpringBoot框架。（注：如果被提出禁止使用spring框架的不合理需求，提前进行讨论）

项目架构使用标准的controller和vo，service，dao和model。

## Controller层

接口API的controller层的逻辑：

1进行接口输入的vo对象的检查

2对service接口的调用

3将service返回结果包装并返回给用户

结果返回央视

{

"errorCode": 0, --0 成功 1失败

"errorMessage": "success", --消息，如果失败则是失败原因

"data": JSON对象 --具体的返回值，如果没有就是空

}

说明，对于输入对象的检查，第一个版本为了快速上线，首先只进行如下三个项目的检查，并且只对字符串进行检查。并且使用简单的自写方法。后续可以升级为使用JPA等组件进行检查。

检查可选

1 非空检查

2 长度检查

3 限定数字字母横杠下划线

## dao和model

由于本项目的特殊性，不使用文件或数据库的存储，而是使用内存存储。

第一个版本对查询要求较低，为了先上线全部使用hashmap进行键值对的存储。在后续版本优化为hashmap+list或者甚至B+数结构。

Key值都是用本来用于在数据库中作为唯一索引的关键列。

说明：如果后续使用mysql数据库，建议增加自增id列。

特别注意：

本次使用简单方案，在存储达到限额后就禁止后续的添加，要求用户进行删除。

未来的优化方案：

存储中增加上次使用时间，或者最近使用次数，并且在存储将满后根据这两个属性删除旧的记录。

Token存储

Key: token字符串 value：空

User存储：

Key: username(唯一限定值) value: User对象

Role存储：

Key： roleCode（唯一限定值） value：Role对象

UserRole存储：

UserRole代表user和role的关联关系

Key: username(唯一限定值) value：rolecode字符串数组

# auth验证

## token生成

详见接口文档的doAuth接口，通过提供用户名密码计算出一个token，用于用户的后续身份验证。

使用极简方案：

Token = 用户名 + 分隔符（###） + 当前毫秒时间戳，然后逆序

## Token身份验证

在对应需要身份验证的接口中，调用对应的checkToken方法，方法逻辑：

1 检查该token是否存在

2 逆序token，根据分隔符拆分

3 检查用户名是否存在

4 检查时间戳是否在两小时之内

## Token注销

从存储中将该token删除即可

# 用户

标准的增删查改，使用username作为唯一key。

增加和删除都需要校验唯一key。

删除时需要校验没有使用中的用户关系（后续实现）

通过接口收到用户的密码后，对密码进行解密（简单方案：将第一个字符放到最后一个字符），然后再进行加密存放到自己的内存中（极简方案：将字符串逆序）。在后续的所有使用中都先反逆序后再使用。（后续优化）

详略

# 角色

标准的增删查改，使用rolecode作为唯一key。

增加和删除都需要校验唯一key。

删除时需要校验没有使用中的用户关系（后续实现）

详略。

# 用户角色关系

用户角色关系没有专用的对象，直接将username和他对应的rolecode数组存储到hashmap中。

增删都对是否已存在进行检查，通过后更新hashmap。