



根据现有业务，大致思考为上述的表模型，所有 ‘人’ 都标识为用户。

用户在注册员工时，应先为用户。所以在注册员工的时候应该以手机号为唯一标识，找用户

用户有注册来源，标识第一次的注册方式。

在业务系统处理时，无论是B端老师还是C端学生，都应该一律视为用户，在业务侧应该明确知道这个人的身份是什么

登录鉴权

用户鉴权基于JWT+Redis 共同实现

horse-user-auth, 引入鉴权模块, 默认所有的接口都会走鉴权, 如不需要鉴权使用 @NoAuth注解 跳过验证, 也可以使用配置设置白名单

```
1 user:
2   auth:
3     whiteList: /**/
```

引入horse-user-auth 后, 内部增加了一个拦截器, 会去拦截所有请求, 使用拿到Header内的Token后回去请求User服务的Dubbo接口会去校验Token 并拿到用户信息写入UserContext中

用户Token Redis 设计

整体Token 数据结构设计为Hash

key: UserId

hashKey: Token

value: UserLoginInfo

因为使用Hash数据结构, 无法使用Redis的过期淘汰对应的Token, 因此手动实现了Token的过期。

这样的数据结构, 将很好的实现, 对于系统和客户端各个维度实现登录限制和单点登录就非常简单, 目前只需要对UserLoginLimit中插入

对应客户端，系统，已经登录限制个数，设置对应的限制模式，拒绝登录和过期最老Token，在过期Token时会对Mq写入通知，各个系统可以根据需要去订阅

topic:

```
1 com.msb.user.api.mq.UserTokenLogoutMessage
```