## 用户

**登录实现思路**

1. 校验验证码,帐号,密码等
2. 校验成功生成唯一token作为已登录标志
3. 保存登录日志
4. 保存到session(google cache)
5. 保存到redis
6. 校验错误,则保存到session计数,超过5次生成验证码
7. 校验成功移除session错误计数,取消验证码

**获取当前用户思路**

1. 拦截器获取header传递过来的token
2. 从session cache获取
3. Session cache 没有则查询redis
4. 查询redis没有则token无效,redis有则说明应用已经重启或者session已经过期,重新设置session,以便下次查询

**注:** goddess项目 token传递是通过dubbo的隐式参数传递: 登录拦截器(LoginIntercept) handlerUserToken方法进行设置

association 项目则通过监听器(RequestListener)对当前线程的request进行监控,在获取token是每次从新的request获取设置

**用户注册思路**

1. 根据session id 生成对应的验证码
2. 注册必须提交验证码(现在验证码已经注释,暂时不用,如果有提示需要验证码,请填写随意数字即可)
3. 检测密码的强度
4. 校验帐号是否已被注册
5. 注册成功,移除session验证码

**注**:社群联盟的注册虽然跟新issp的思路是一样的,但是社群联盟注册会多一些初始化的逻辑代码,比如注册成功会自动登录,详见代码,都写有详细注释.而新issp的设计为多系统设计,如区分多公司使用,用户区分用SystemNO来区分是哪个公司或者哪个系统的,目前只有用户模块区分.

**涉及表**

**association:**

user(用户),user\_info(用户信息),user\_sign(签到),user\_login\_log(登录日志)

**goddess:**

user(用户) user\_department(部门),user\_detail(详情信息),user\_login\_log(登录日志),user\_position(职位)

其他（权限已不使用）：

rbac\_group（组），rbac\_group\_role（组角色），rbac\_group\_user（组用户），rbac\_permission（资源表）

rbac\_role（角色），rbac\_role\_permission（角色资源），rbac\_user\_role（用户角色）