单点登录：（一张user表 用户名、密码（加@JsonIgnore注解，序列化时忽略它）、电话、邮箱）

业务：

当模块太多时，很多系统都需要登录验证，避免此现象就有了单点登录，由一个系统来完成登录操作，并通过将用户的token和用户信息写入Redis中，返回信息时写入到用户cookie中，之后所有的系统如要登录验证则通过sso的通过token查询Redis 的用户信息的API。

实现：

数据可用性：（参数、数据类型（会根据类型switch 匹配前端传递的参数））

匹配用户传入的电话、邮箱、用户名是否已经存在

用户注册：

用实体类注解的形式（jsr303,不知道是不是），校验用户注册的信息，接收参数时还需在参数前加**@Valid注解**,它和**BindingResult 是一起用的**，用来获取实体类里参数不满足的错误信息，如果用户注册的信息不满足则给前端返回相应的错误信息（其实这一段校验只是双重备份，在前端拦截就够了其实），如果满足则将用户信息补充创建时间和修改时间，**并且将用户的密码以MD5加密的形式（DigestUtils.md5Hex（）这个工具类就可以实现）持久化到数据库。**

用户登录：（user）

首先拿用户名去数据库里查询是否存在，如果不存在直接返回，如果存在则拿用户名和经过MD5加密后的密码去数据库匹配用户，如果用户不存在则直接返回，如果存在则生成**token**（以时间戳和用户名并且MD5加密的形式生成），然后以token为key、用户信息为value（因为已经在user类中给password加@JsonIgnore用于忽略序列化json串）、并且加生存时间（30分钟），存储于redis数据库，并且将token存储于cookie 中

还有一个用于其他系统用来校验用户登录的接口：（token）

直接拿token 去Redis 中查用户，如果用户不存在则直接返回，如果存在则刷新当前用户的token 在Redis中的生存时间（30分钟），最后返回用户信息。