

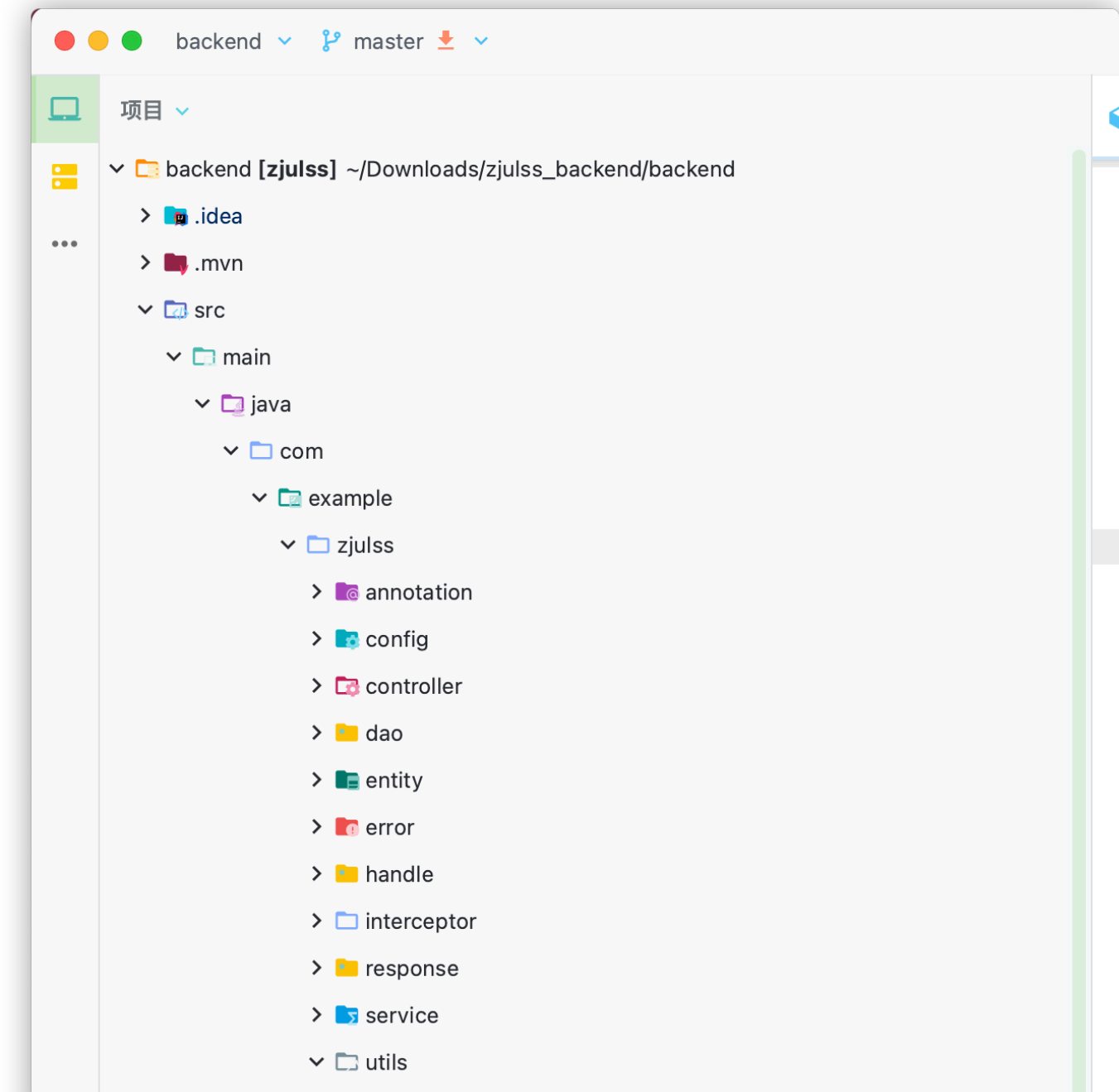
大规模信息系统开发课程 第一阶段个人报告

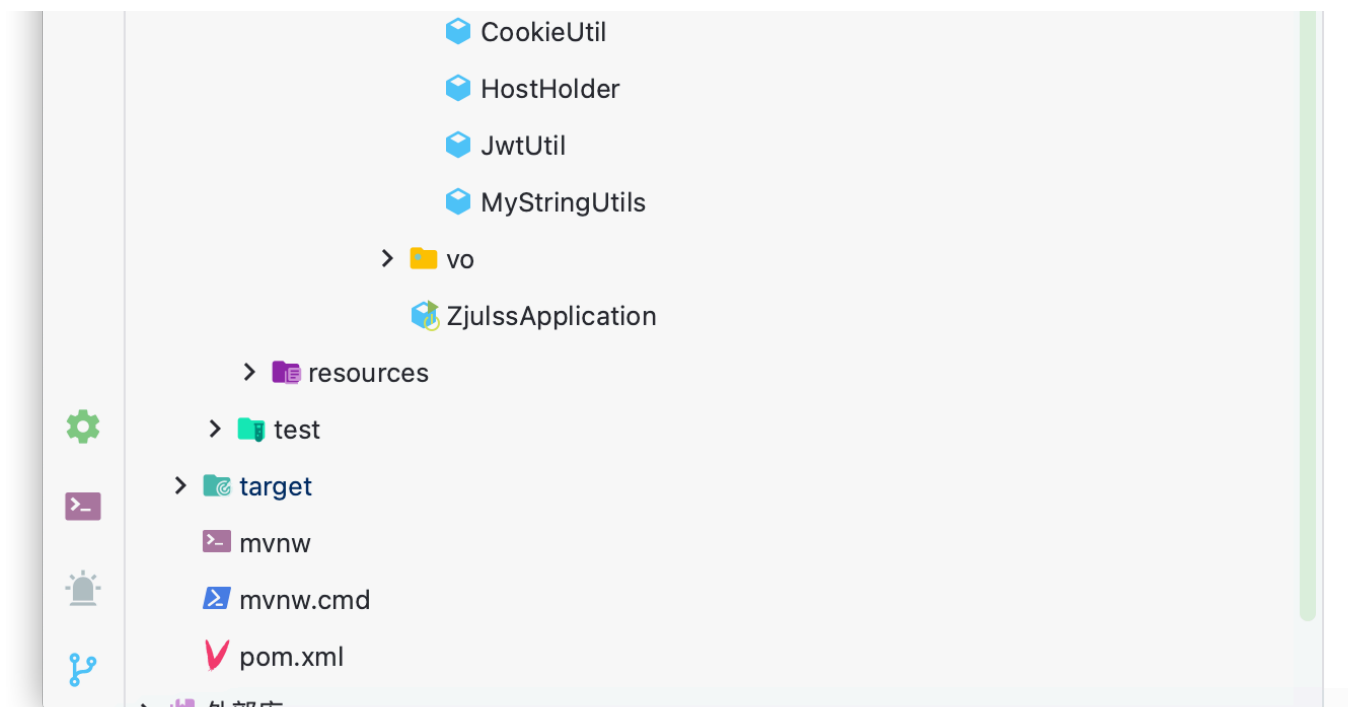
3200105872 庄毅非

1. 阶段概述

1.1 后端开发情况

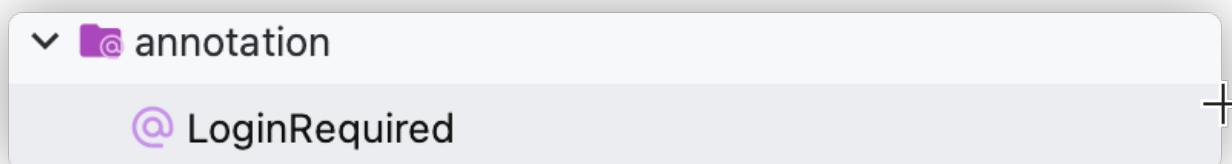
在这一阶段我主要参加了系统后端的总体设计，使用经典的 DDD 开发模型进行开发，将项目整体分为 DAO、Service、Controller 层，使用了 SpringBoot 框架进行开发，使用了 MyBatis 框架进行数据库操作，使用了 Lombok 框架进行代码的简化，使用了 Junit 框架进行单元测试，使用了 Maven 进行项目管理，使用了 Git 进行版本控制，使得后端项目整体开发比较顺利。以下是项目架构图。



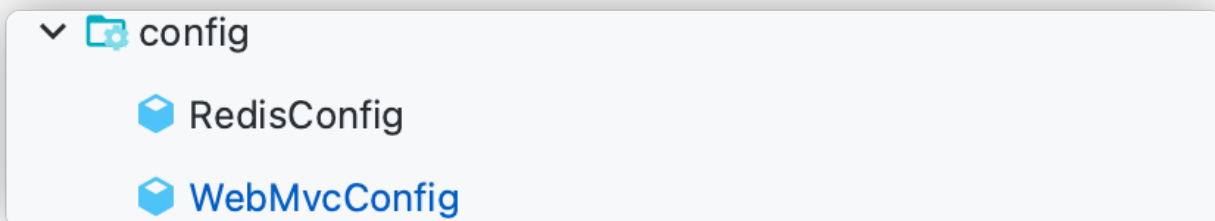


下面简要介绍以下各个模块的功能。

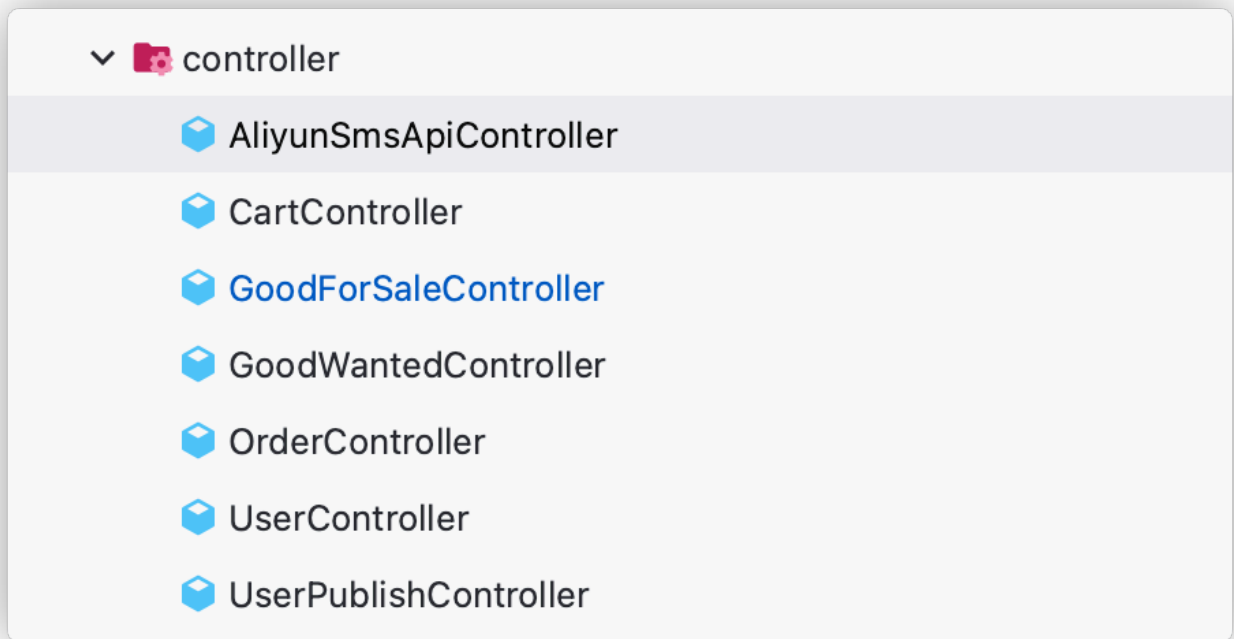
1. annotation 模块：这个模块定义了一个 `LoginRequired` 注解，方便后端对应请求的处理器判断接口是否需要用户登陆才可以访问。



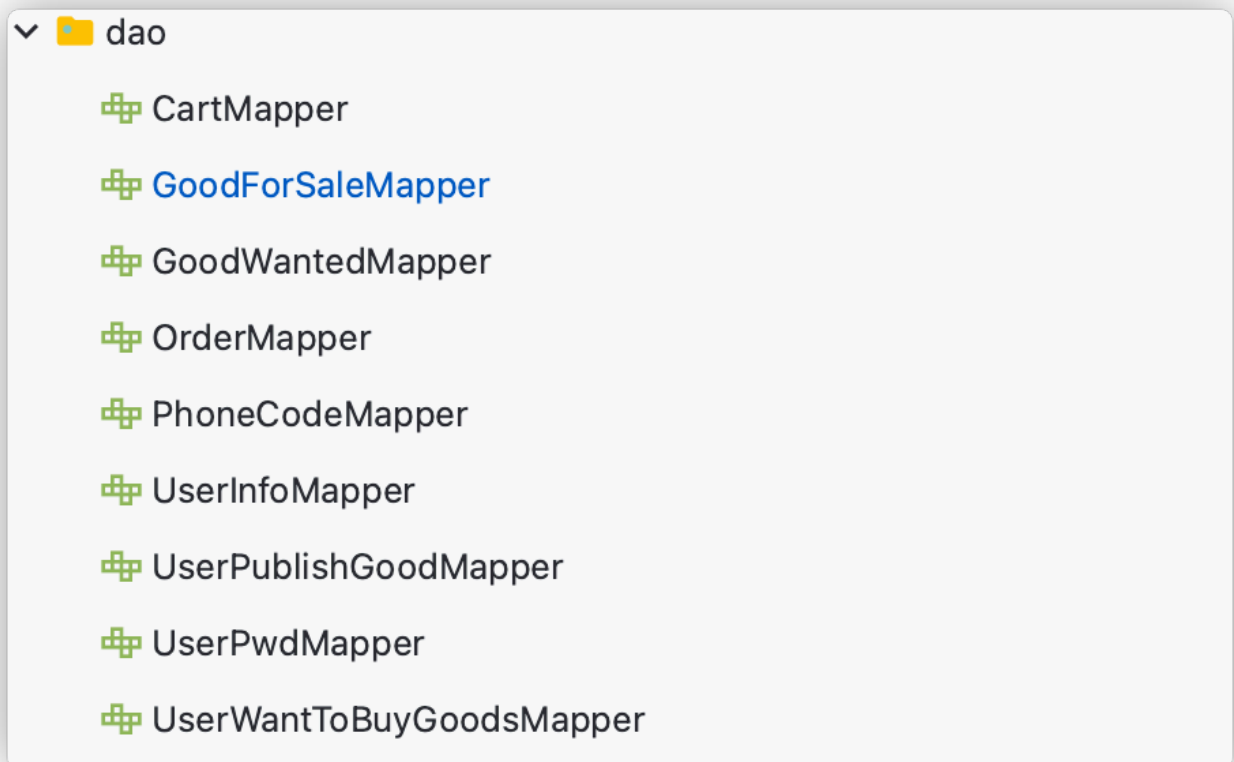
2. config 模块：这里主要是引入了 `Redis`，以及进行了上述注解处理器的注册。



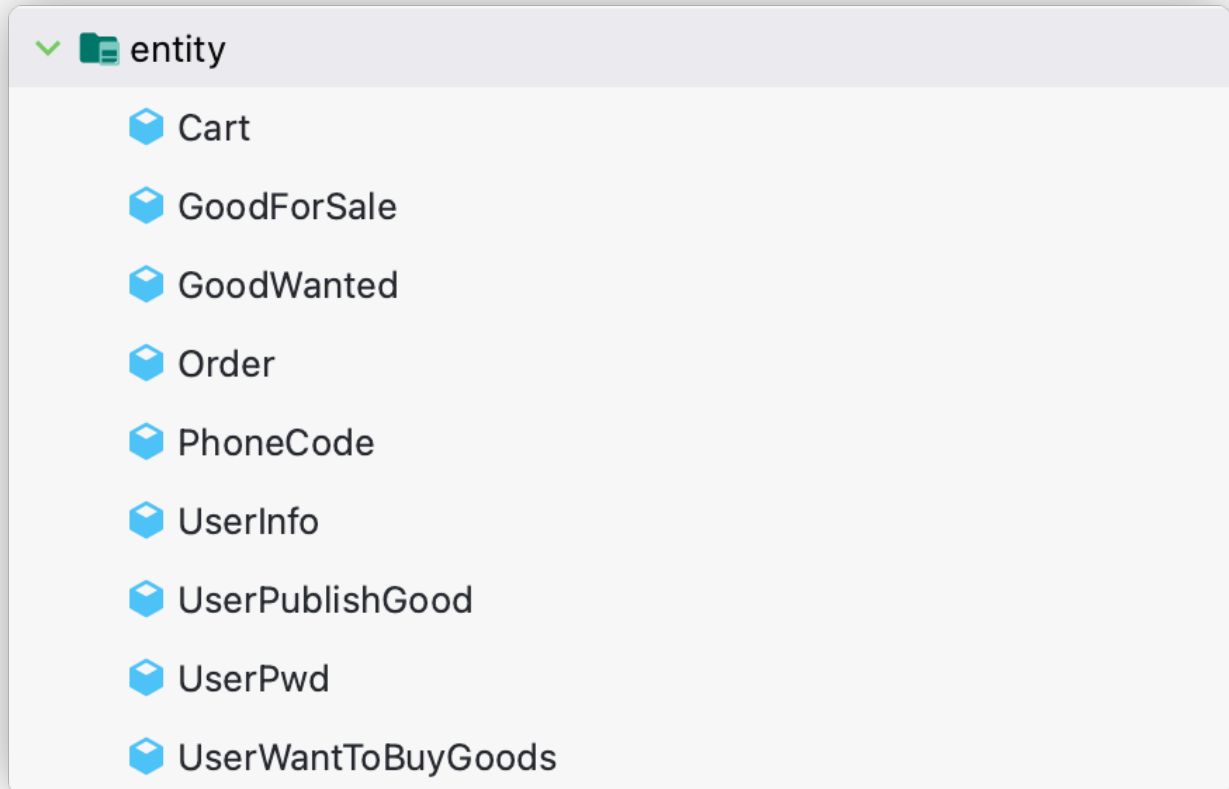
3. controller 模块：这里主要是定义了一系列后端接口的处理器，包括阿里云短信接口，购物车、商品出售、商品求购、订单、用户请求的处理。



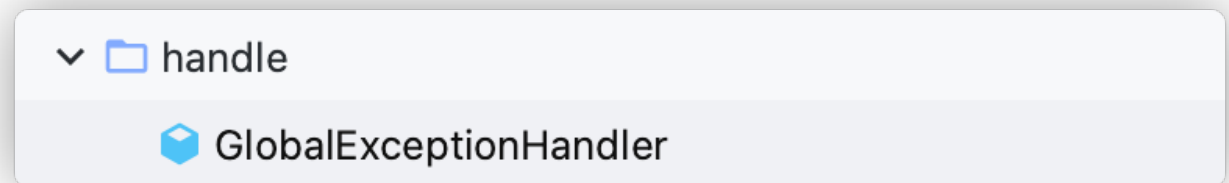
4. dao 模块：这里主要是定义了一系列的数据库操作接口。



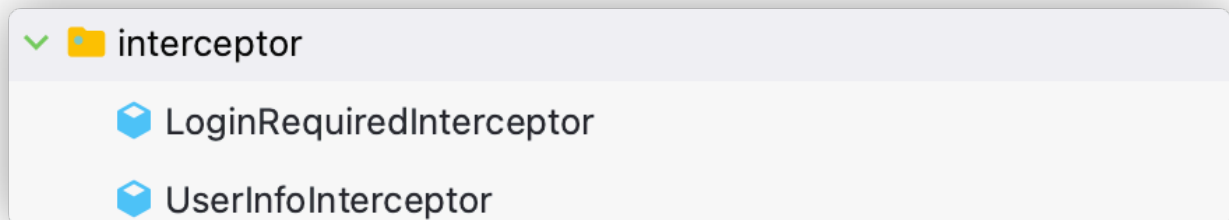
5. entity 模块：这里主要是定义了一系列的实体类，包括购物车、商品出售、商品求购、订单、用户等。



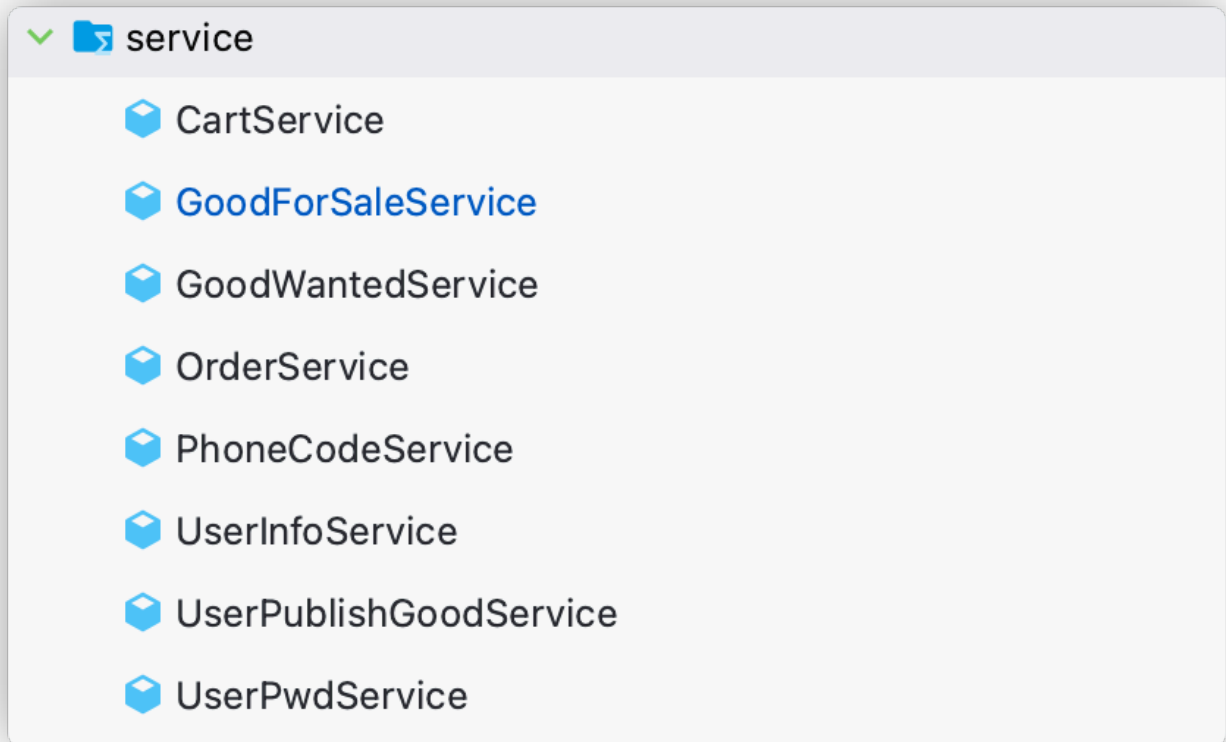
6. handle 模块：这里注册了全局异常处理器。



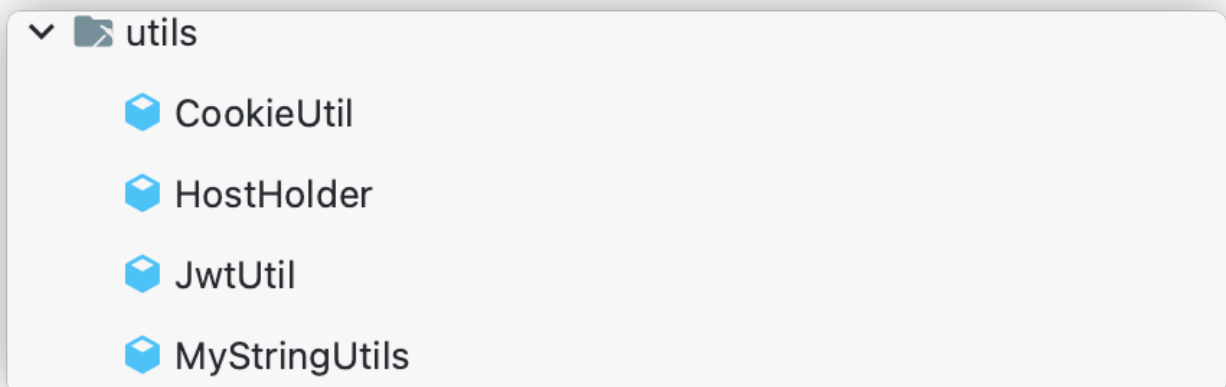
7. interceptor 模块：这里注册了全局拦截器，用于判断用户是否登陆。



8. service 模块：这里对 `controller` 模块提供了对应的访问接口，用来完成各项服务。



9. utils 模块：这里主要是提供了对应的工具类，完成一些基础的功能，比如 jwt 生成和解码。



1.2 服务器部署

同时我也参与了后端的部署工作，我在第一阶段的需求完成之后，我将后端部署到了服务器上，使得前段同学可以直接调试。

```
not active
JDBC Connection [HikariProxyConnection@950213060 wrapping com.mysql.cj.jdbc.ConnectionImpl@76ce8118] will not be managed by Spring
==> Preparing: select * from good where uid = ?
==> Parameters: 10(Integer)
<== Total: 0
Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@144e50bd]
Bearer eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJlYTRlNTZiZi1jN2U3LTRkMGUtYTkyZS0xZjJlZTYwODM4N2UiLCJzZWUiOiIxMCIzImIzcyI6InpqcWxzczIiLCJpYXQiOiJE20DIwNjk4MTcsImV4cCI6MTY4MjI0MjYxN30.zJ79AMgyZFfbPs-LATeRwW03VzKtK-XvjlkH9XXd-Y
<== Total: 0
Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@4fb4298a]
Creating a new SqlSession
SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@27cd91df] was not registered for synchronization because synchronization is not active
JDBC Connection [HikariProxyConnection@1448047078 wrapping com.mysql.cj.jdbc.ConnectionImpl@26a87c59] will not be managed by Spring
==> Preparing: select id,user_name,modify,phone,real_name,clazz,sno,dormitory,gender,create_time,avatar from user_info where id = ?
==> Parameters: 10(Integer)
<== Columns: id, user_name, modify, phone, real_name, clazz, sno, dormitory, gender, create_time, avatar
<== Row: 10, 123456, null, 18985013706, , 软工2003, 3200105319, 32, 男, 2023-04-14 07:34:43,
<== Total: 1
Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@27cd91df]
UserInfo(id=10, modify=null, userName=123456, phone=18985013706, realName=, clazz=软工2003, sno=3200105319, dormitory=32, gender=男, createTime=2023-04-14T07:34:43, avatar=)
Creating a new SqlSession
SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@52512cf2] was not registered for synchronization because synchronization is not active
JDBC Connection [HikariProxyConnection@433654798 wrapping com.mysql.cj.jdbc.ConnectionImpl@26a87c59] will not be managed by Spring
==> Preparing: select * from cart where uid = ? and display = 1
==> Parameters: 10(Integer)
<== Total: 0
Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@52512cf2]
[backend-00:java* 1:bash- "zjulss2-s" 04:06 30-Apr-23]
```

1.3 前后端联调

同时我也参加了和前端的对接工作，在和前端第一阶段的开发同学对接的时候，他指出了调试的时候出现的一些问题（比如 jwt token 过期的问题），我也及时进行修改，将修改之后的版本部署到服务器上。

最后，我在第一阶段也做了一些优化操作，比如用户信息这一热点数据存储在 Redis 中，这样来避免程序运行过程中多次冗余查询用户数据库，提高应用的并发能力。

2. 后续工作

在后续的工作中，我计划继续和前端对接，开发前端所需的接口，同时也会继续优化后端的代码，使得后端的代码更加简洁，同时也会继续优化后端的性能，使得后端的性能更加优秀。

在提高并发能力方面，我计划使用消息队列进行购买信息的推送，来对后端进行削峰操作，同时也保证消息不会丢失。

目前需求文档里的商品搜索功能展示还没有实现，我计划在后期使用 Elasticsearch 框架进行搜索功能的支持，让系统在商品量非常大的时候也可以在很短的时间内完成精确搜索和模糊搜索，保证用户的体验。