

厚德 求真 砺学 笃行



西安电子科技大学  
XIDIAN UNIVERSITY

# 线上图书商城系统

2025年6月12日

TASK 07

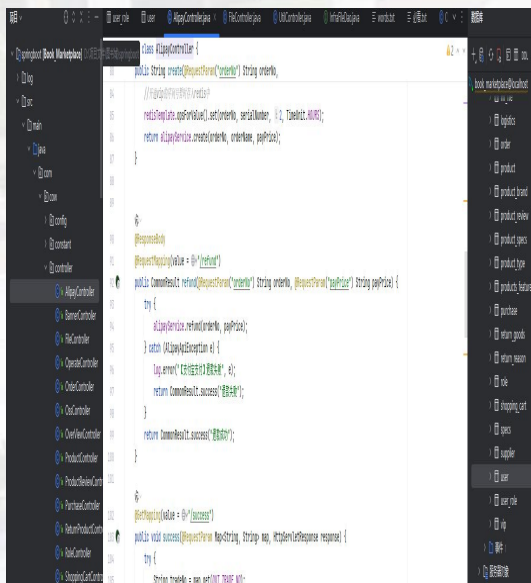
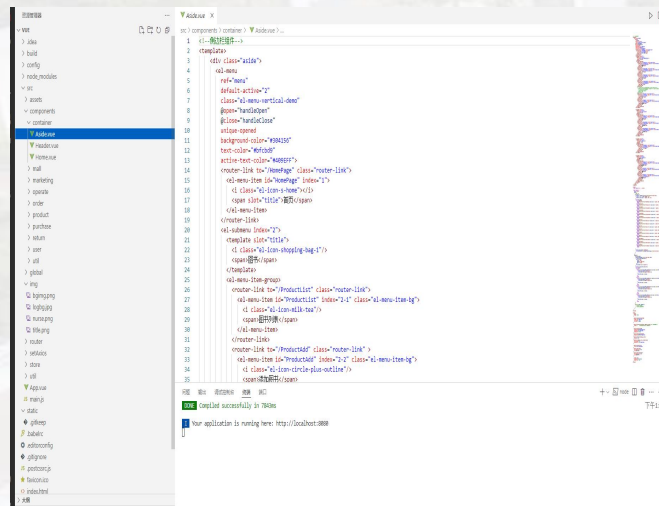
# Web应用构建



前端使用vue

后端使用springboot

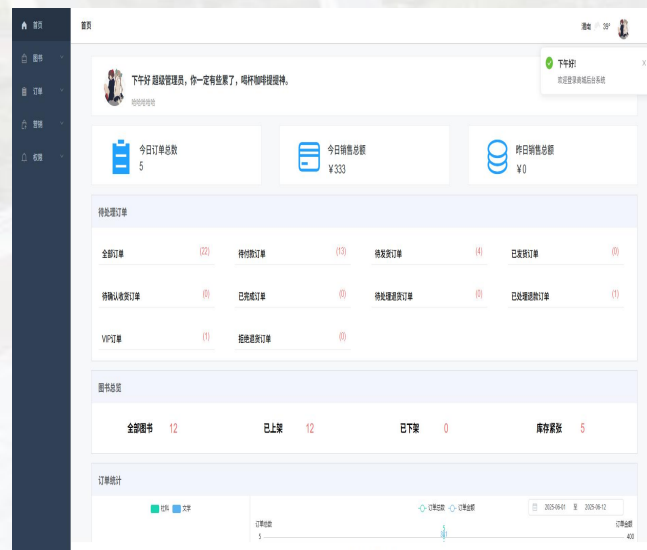
数据库采用mysql和redis



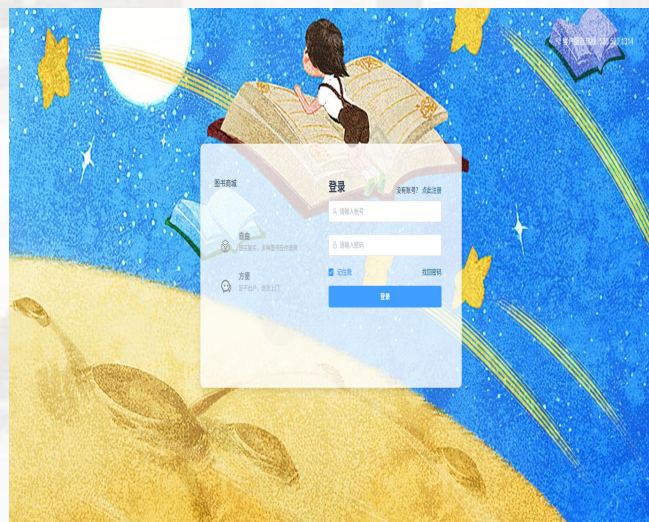




## 管理员页面



## 注册页面



## 用户页面



TASK 08

# Web应用测试



对类、方法进行测试

测试框架：**JUnit 5+Mockito+Spring Test**

JUnit 5 提供测试运行环境和基础注解（如@Test、@BeforeEach）  
通过@ExtendWith(MockitoExtension.class)与 Mockito 集成

Mockito 模拟依赖对象（如UserDao），使用@Mock和@InjectMocks注解  
验证方法调用（如verify(userDao).selectById(1)）  
配置模拟对象行为  
（如when (userDao.selectById(1)).thenReturn(testUser)）

Spring Test 通过@ExtendWith(MockitoExtension.class)提供上下文支持  
未直接使用@SpringBootTest，通过 Mockito 模拟 Spring 组件





## 功能测试

编写不同的测试用例对编写的controller包下的接口进行功能测试

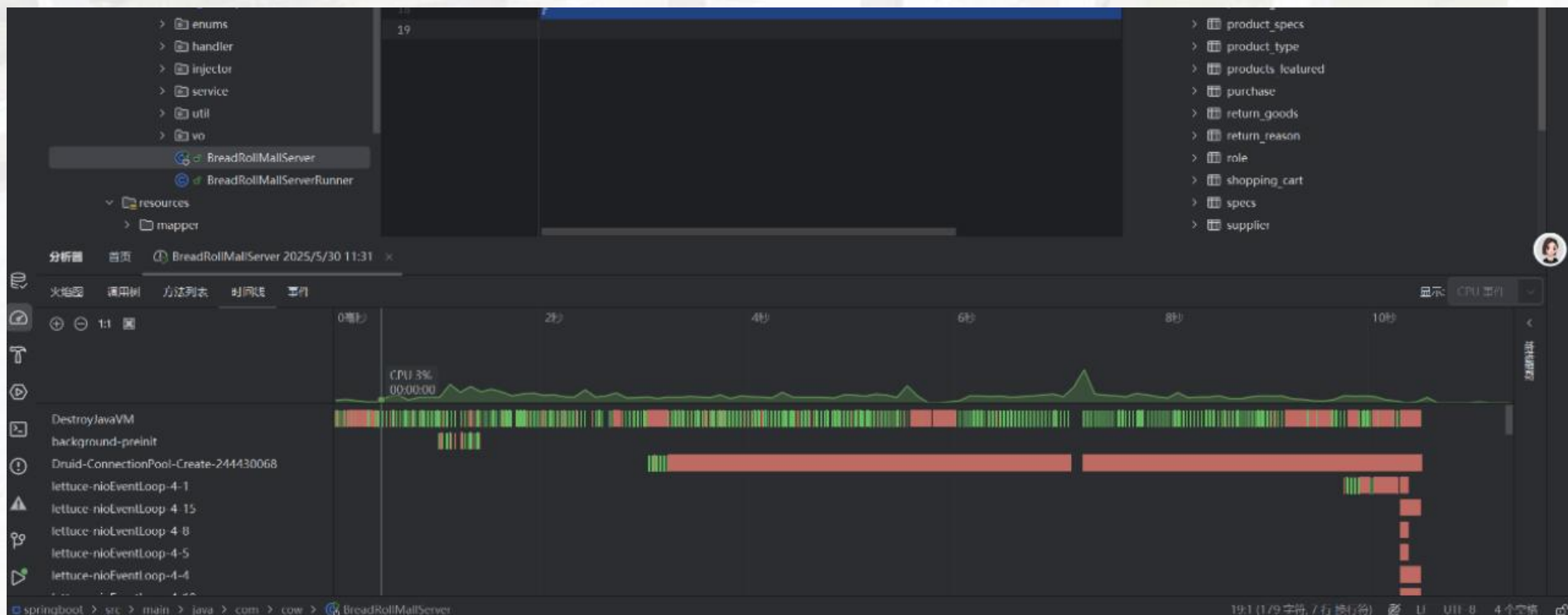
The screenshot shows an IDE with a project structure on the left and a REST client request in the main editor. The project structure includes packages like SpecsController, UserController, UserRoleController, UtilController, dao, and entity. The entity package contains various entities like Banner, Cache, InfFileDO, Logistics, Order, Product, ProductBrand, ProductReview, ProductSpecs, ProductType, Purchase, ReturnGoods, ReturnReason, and Role. The REST client request is a GET request to the endpoint `2025-05-30T114733.500.json`. The response is a JSON object with the following fields:

```
Content-Type: application/json
{
  "productId": 1001,
  "productNo": "P20250530001",
  "productName": "测试商品",
  "productType": "图书",
  "productDescribe": "这是一个测试商品描述",
  "productBrand": "未知",
  "inPrice": 20.5,
  "outPrice": 39.9,
  "productStock": 100,
  "lowestStock": 10,
  "isStockOut": false,
  "isNew": true,
  "isSale": true,
  "saleTime": "2025-05-30T12:00:00",
  "productUrl": "https://picsum.photos/200/300"
}
```



## 性能测试

使用springboot自带性能测试工具

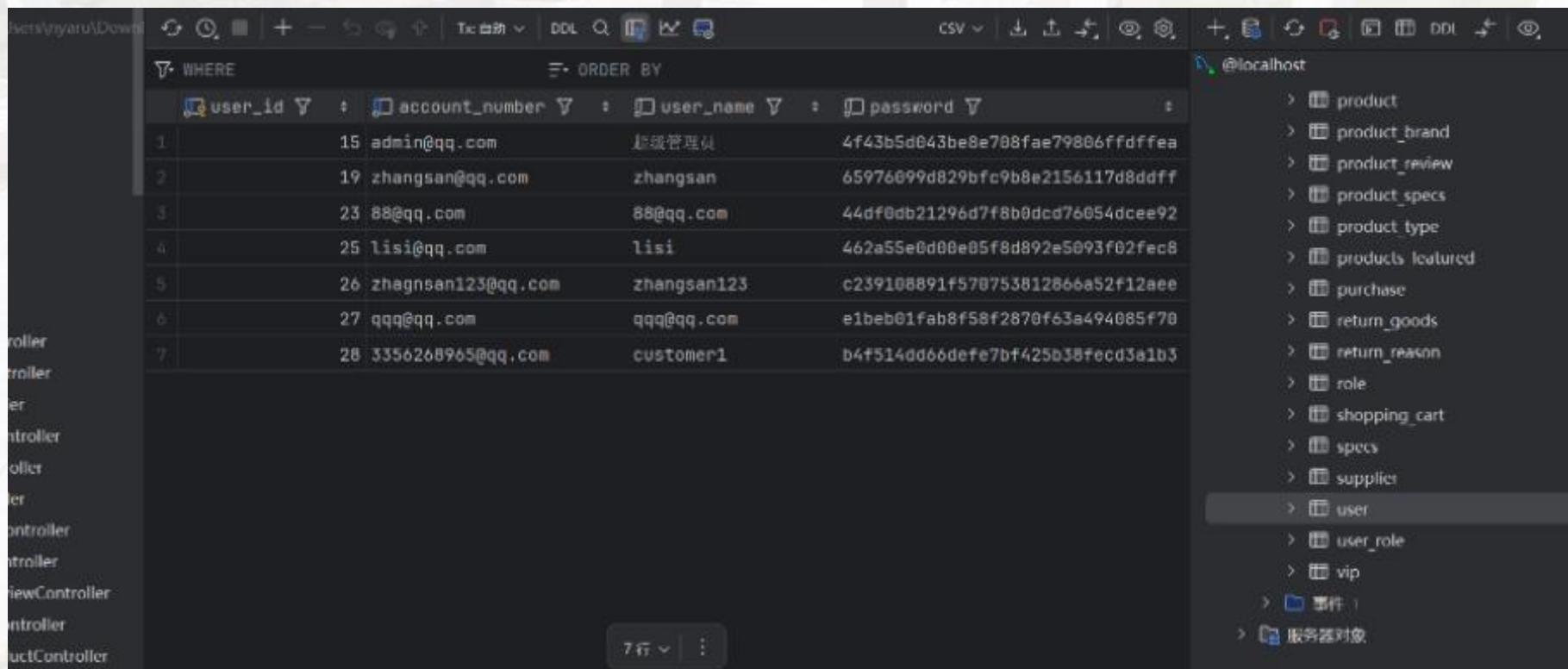






## 安全性测试

用户密码Md5加密，用户输入密码后进行加密再与数据库进行比较。



	user_id	account_number	user_name	password
1	15	admin@qq.com	超级管理员	4f43b5d043be8e708fae79806ffdfdea
2	19	zhangsan@qq.com	zhangsan	65976099d829bfc9b8e2156117d8ddff
3	23	88@qq.com	88@qq.com	44df0db21296d7f8b0dcd76054dcee92
4	25	lisi@qq.com	lisi	462a55e0d00e05f8d892e5093f02fec8
5	26	zhagnsan123@qq.com	zhangsan123	c239108891f570753812866a52f12aee
6	27	qqq@qq.com	qqq@qq.com	e1beb01fab8f58f2870f63a494085f70
7	28	3356268965@qq.com	customer1	b4f514dd66defe7bf425b38fec3a1b3

TASK 09

# Web应用运维



### 1.1 商品信息维护

- **定期更新**商品信息（价格调整、库存更新、商品描述优化、图片更新）
- **商品分类**维护（分类结构调整、分类描述优化、分类图片更新）

### 1.2 用户内容维护

- **用户评价管理**（评价审核、违规评价处理、优质评价置顶）
- **用户反馈处理**（问题分类、及时响应、解决方案跟踪）

### 1.3 系统内容维护

- 新闻公告更新、帮助文档维护、系统通知管理、活动信息发布





### 2.1 内容完整性检查

- 商品信息完整性（标题、描述、价格、图片、规格参数）
- 分类信息完整性
- 用户评价完整性

### 2.2 内容准确性验证

- 价格准确性、库存准确性、商品描述准确性、活动信息准确性

### 2.3 内容时效性管理

- 过期内容清理、活动信息更新、促销信息管理、新闻公告更新



### 3.1 数据库备份

- 每日全量备份、实时增量备份、定期备份验证、备份文件管理

### 3.2 文件备份

- 商品图片备份、系统配置文件备份、日志文件备份、用户上传文件备份

### 3.3 备份恢复机制

- 备份文件存储、恢复流程制定、恢复演练、应急预案

## URL优化和网站结构优化

1. URL结构优化
2. 生成网站地图
3. 实现面包屑导航

## 采用原因：

1. 提高搜索引擎对网站结构的理解
2. 提升URL的可读性和用户体验
3. 便于搜索引擎爬虫抓取内容
4. 有助于提高关键词排名
5. 增加网站内部链接权重





## 技术优化和性能提升

1. 页面加载优化
2. js处理
3. 移动端适配
4. 响应式设计

## 3.2 采用原因

1. 提高网站加载速度
2. 改善移动端用户体验
3. 提升搜索引擎排名
4. 增加用户停留时间
5. 提高网站整体性能



TASK 11

# Web应用性能 和可用性分析 与调优





## 1. 后端性能优化 (SpringBoot)

- 使用**异步处理任务**：如邮件通知、支付结果回调等，使用@Async提高响应速度。
- **分页加载数据**：对图书列表、订单列表等内容，采用分页查询避免一次性加载大量数据，使用PageHelper或JPA的分页支持。
- **缓存机制**：---使用Redis缓存热门图书数据、首页轮播图信息、分类目录等，减少数据库访问压力。---SpringCache注解方式配合Redis使用，自动处理缓存。
- **数据库连接池调优**：---使用HikariCP (SpringBoot默认)；---调整maximumPoolSize等参数以支持高并发。





## 1. 后端性能优化 (SpringBoot)

- **使用异步处理任务**：如邮件通知、支付结果回调等，使用@Async提高响应速度。
- **分页加载数据**：对图书列表等内容，采用分页查询避免一次性加载大量数据。
- **缓存机制**：使用Redis缓存热门图书数据、首页轮播图信息、分类目录等，减少数据库访问压力。SpringCache注解方式配合Redis使用，自动处理缓存。
- **数据库连接池调优**：使用HikariCP；调整maximumPoolSize等参数以支持高并发。
- **减少数据库请求**：合理使用@Transactional 控制事务；避免N+1 查询，优化JPA或MyBatis 的查询语句；批量处理插入和更新。
- **接口限流**：使用Bucket4j 或Sentinel 对特定接口进行限流处理，防止恶意请求或突发流量导致系统崩溃。

## 2. 前端性能优化 (Vue)

- **懒加载 (Lazy Load)**：图片懒加载，减少页面初始加载资源；Vue 路由懒加载（使用动态import()）减少首页加载体积。
- **组件缓存**：使用<keep-alive> 对路由组件进行缓存，提升返回页面速度。
- **打包优化**：使用Webpack 的代码拆分 (Code Splitting)；压缩JavaScript 和CSS；Tree Shaking 移除未使用的代码。
- **减少HTTP 请求数**：合并资源文件 (CSS、JS)；使用HTTP/2 提升请求并发效率。



### 3. 网络与部署优化

- 前后端分离部署，通过Nginx 做静态资源分发和反向代理；
- GZIP 压缩静态资源；
- CDN 加速静态资源（如图书封面图片）；
- 部署集群+ 负载均衡（如Nginx+Spring Boot 多实例）以实现高可用性；
- 服务监控与日志分析：如使用Spring Boot Admin、Prometheus+Grafana、ELK 进行服务性能与异常监控。



## 1. 用户界面与交互设计（前端）

- **响应式设计**：支持PC 端本地浏览；
- **友好的错误提示与加载状态**：页面加载中显示Spinner；操作失败/成功给出明确提示（Toast、Snackbar）；对404、500 页面进行美化，提供返回主页按钮。
- **导航清晰**：分类导航清晰可见；面包屑导航增强用户定位；表单验证与提示：对购物车、支付、注册等表单进行前端验证； 提供实时错误反馈（如“邮箱格式错误”）。

## 2. 功能可用性（后端）

- **支付模拟稳定性：**支付宝沙箱环境异常情况的兜底处理；对支付状态做幂等性处理，避免重复下单；
- **搜索功能优化：**支持拼音模糊匹配。
- **访问控制与权限管理：**卖家/买家分权限显示功能；防止未授权用户访问管理页面。



TASK 12

# Web应用安全性分析与防护





## 1. SQL 注入

- 危险点：使用字符串拼接构造 SQL 语句。
- 建议：使用 PreparedStatement 代替 Statement，避免直接拼接用户输入。

## 2. XSS（跨站脚本攻击）

- 危险点：将用户输入直接输出到网页中而不做 HTML 编码。
- 建议：使用 StringEscapeUtils.escapeHtml4() 或前端框架中的 XSS 防护机制。



### 3. CSRF（跨站请求伪造）

- 危险点：用户登录后自动发送敏感请求（如删除账户）。
- 建议：为表单添加 CSRF Token 并在服务端验证。

### 4. 敏感信息明文存储

- 危险点：将数据库账号密码写死在代码或配置文件中，或以明文存储用户密码。
- 建议：使用 `application.properties`（或 `.env`）进行配置管理。用户密码使用哈希算法（如 `bcrypt`）加密存储。

### 5. 身份认证与会话管理不当

- 危险点：未检查用户身份即可访问受限资源。
- 建议：使用 HttpSession 管理用户登录状态；对关键控制器加认证注解（如 @PreAuthorize、@Secured）。

### 6. 文件上传漏洞

- 危险点：允许上传任意文件，可能被上传恶意 .jsp 或 .exe。
- 建议：验证文件类型和扩展名；存储路径应避免与 Web 路径一致；重命名上传文件，避免执行。





### 1. 密文存储密码（使用 BCrypt 加密）

明文存储密码是严重安全漏洞。使用 BCrypt 等哈希算法可以抵御暴力破解和泄露攻击。

### 2. 用户认证与授权（Spring Security）

配置用户认证（内存或数据库），从数据库中加载用户和角色：



## SQL 注入测试用例

用例编号	测试目标	请求位置	测试输入
SQL-01	登录接口注入	/login 参数 username	' OR '1'='1
SQL-02	注册接口注入	/register 参数 email	test@test.com'); DROP TABLE users; --
SQL-03	查询商品	/search?keyword=	' OR 1=1 --

## XSS 测试用例

用例编号	测试目标	输入位置	测试输入
XSS-01	用户评论区	comment 参数	<script>alert('xss')</script>
XSS-02	用户名字段	username	<img src=x onerror=alert(1)>
XSS-03	搜索框回显	q	<svg/onload=alert(1)>



## CSRF 测试用例

## 认证与授权绕过测试

用例编号	测试目标	请求位置	测试操作
CSRF-01	删除商品	模拟已登录用户访问 	操作失败，需 CSRF Token
CSRF-02	修改密码	模拟表单提交，无 Token	修改失败，提示无效请求

用例编号	测试目标	路径	测试操作
AUTH-01	未登录访问受限页面	/user/dashboard	应重定向到登录页面
AUTH-02	普通用户访问管理员接口	/admin/addProduct	返回 403 Forbidden
AUTH-03	修改用户 ID 尝试越权	/user/edit?id=2	拒绝操作，不允许更改他人信息



## 密码安全测试用例

用例编号	测试目标	测试内容
PASS-01	注册时密码弱	使用 123456 或 password
PASS-02	数据库存储密码	查看数据库中的密码字段
PASS-03	登录密码错误	输入错误密码登录

## 敏感信息泄露测试用例

用例编号	测试目标	请求
INFO-01	访问 .git/ 路径	/webapp/.git/config
INFO-02	报错信息暴露	故意请求无效字段或 SQL 错误

厚德 求真 砺学 笃行

# THANKS

Q U E S T I O N S & A N S W E R S

2025年6月12日



西安电子科技大学