

E-bookshop 项目开发文档



项目成员：

王昱

胡锦涛

姚聪

目录

-1
- 1. 开发规划.....3
 - 1.1 人员分工3
 - 1.2 开发环境及工具.....3
- 2.总体设计3
 - 2.1 项目介绍:3
 - 2.2 功能结构图.....4
 - 2.3 用例图4
 - 2.4 数据库设计:.....5
- 3.详细设计6
 - 3.1 Controller 设计.....6
 - 3.1.1 BookController.....6
 - 3.2 entity 设计6
 - 3.2.1 Book6
 - 3.2.2 UserInfo.....6
 - 3.2.3 Role7
 - 3.3 Repository.....7
 - 3.3.1 BooksRepository7
 - 3.3.2 UserInfoRepository.....7
 - 3.3.3 RoleRepository7
 - 3.4 service7
 - 3.4.1 bookservice.....7
 - 3.4.2 UserInfoService.....7
 - 3.4.3 RoleService7
 - 3.5 Config.....8
 - 3.5.1 JWTConfig.....8
 - 3.5.2 WebConfig.....8
 - 3.5.3 JWTFilter8
 - 3.6 Testing8
 - 3.6.1 单元测试.....8
 - 3.6.2 集成测试.....9
 - 3.7 前端.....10
 - 3.7.1 实现步骤.....10
 - 3.7.2 界面10

1. 开发规划

1.1 人员分工

角色	分工	人员
项目经理	项目主体设计、前端界面编写、文档撰写	王昱
开发人员	测试模块、数据库连接、文档撰写	胡锦涛
开发人员	JWT 登录验证、角色权限识别、文档撰写	姚聪

1.2 开发环境及工具

开发语言：java

开发软件：idea, webstorm

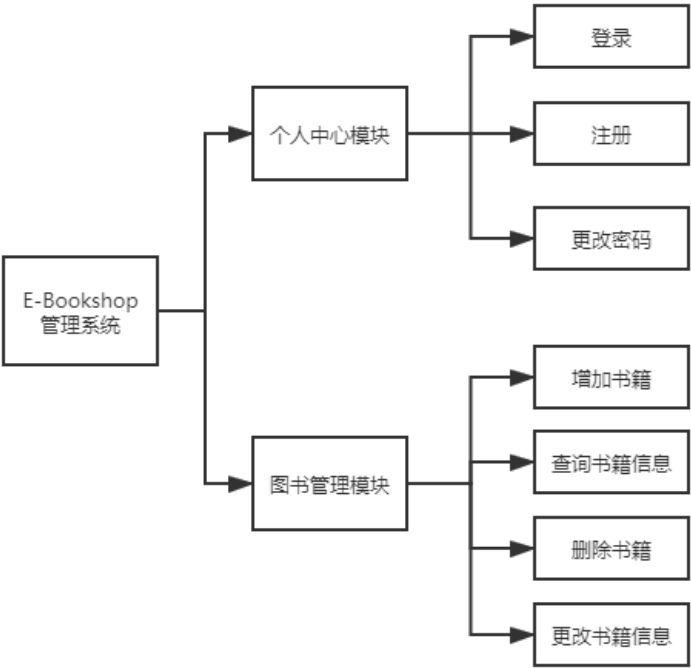
开发框架：Spring Boot, Vue

2.总体设计

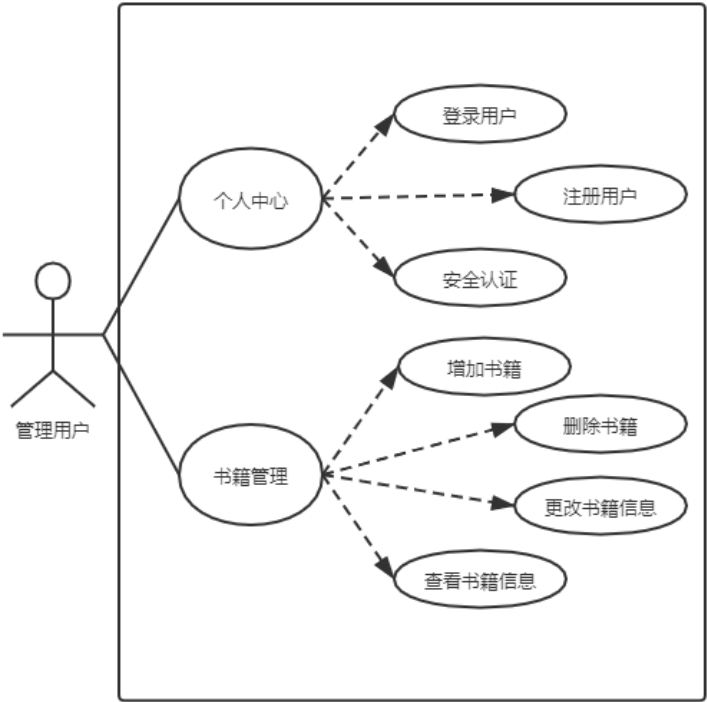
2.1 项目介绍:

E-bookshop 网站的开发，通过 Spring Boot + Vue 的方式前后端分离实现功能。主要功能有：登录注册，图书信息的增删改查等功能，并使用 Spring Security + JWT 实现角色权限管理与登录验证，controller 使用 Restful 风格。

2.2 功能结构图



2.3 用例图



2.4 数据库设计:

Book 表

对象 books @mytest (本地root) - 表				
开始事务 文本 筛选 排序 导入 导出				
ID	bookname	author	intro	price
1	红楼梦	曹雪芹	小说以贾	50.56
2	三少爷的剑	古龙	《三少爷	30.70
7	西游记	吴承恩	猴子取经	40.50
10	test	testAuthor	用于更新	12.50
11	book1	Author1	用于插入	50.00

Users 表

开始事务 文本 筛选 排序			
id	password	username	
1	123456	admin	
2	123456	test	

Role 表

开始事务 文本	
id	rolename
1	ADMIN
2	NORMAL

User_role 表

开始事务 文本	
r_id	u_id
1	1
2	2

3.详细设计

3.1 Controller 设计

3.1.1 BookController

使用 RestFul 风格重写上次作业的 controller，实现书籍管理部分接口设计，包括查看书籍列表、对书籍进行增删改等。

1. 返回书籍列表

```
@GetMapping(value = "/books")
public List<Book> getBooks()
```

2. 返回指定书籍

```
@GetMapping("/books/{id}")
public Book getBook(@PathVariable("id") Integer id)
```

3. 增加书籍

```
@PostMapping(value = "/books")
public void addBook(@RequestBody Book book)
```

4. 修改指定书籍

```
@PutMapping("/books")
public void modifyBook(@RequestBody Book book)
```

5. 删除指定书籍

```
@DeleteMapping("/books/{id}")
public void deleteBook(@PathVariable("id") Integer id)
```

6. 根据书名模糊查询

```
@GetMapping("/books/searchResult/{searchKey}")
public List<Book> searchBooks(@PathVariable("searchKey") String searchKey)
```

3.2 entity 设计

3.2.1 Book

Book 类属性定义，包括 (ID,bookname, author, address, price)，同时实现定义类的构造器和 get、set 方法

3.2.2 UserInfo

UserInfo 类属性定义，包括 (id,username,password)，同时实现定义类的 get, set 放方法。同时 Userinfo 和 Role 通过 JPA 实现多对多联系，可通过 user 对象获得对应 role 角色列表。

3.2.3 Role

Role 类属性定义，包括 (ID,rolename)，同时实现定义类的构造器和 get、set 方法

3.3 Repository

3.3.1 BooksRepository

继承了 JpaRepository，自动实现了有关 book 表的相关 Dao 层操作。

3.3.2 UserInfoRepository

继承了 JpaRepository，自动实现了有关 userInfo 表的相关 Dao 层操作。

3.3.3 RoleRepository

继承了 JpaRepository，自动实现了有关 role 表的相关 Dao 层操作。

3.4 service

3.4.1 bookservice

设计 IBookService 接口，定义相应的数据库操作。BookService 类实现 IBookService 接口，通过 JPA 实现对应的数据库操作。

3.4.2 UserInfoService

设计 IUserInfo 接口，实现了 UserDetailsService 的接口，定义实现了响应数据库操作，并自定义了一个 findByUsername 的方法，来支持查找相应用户名对应的 UserInfo 类。

3.4.3 RoleService

设计 IRolerInfo 接口，实现了 RoleService 的接口，定义实现了响应数据库操作。

3.5 Config

3.5.1 JWTConfig

JWTUtils 类：配置jwt的相关函数，包括：Jwt的生成，校验以及从Jwt中获取用户信息。

3.5.2 WebConfig

SecurityConfig 类：继承了 WebSecurityConfigurerAdapter 类，配置 http 的相关需求，以及跨域的相关配置。

3.5.3 JWTFilter

JWTAuthenticationEntryPoint 类：实现了 AuthenticationEntryPoint 接口，处理了在没有登陆时的相关信息。

JWTAuthenticationFilter 类：继承了 UsernamePasswordAuthenticationFilter 类，从输入流中获取登录信息并进行验证，成功时给返回的 response 添加 JWT 令牌，失败时给 response 添加失败的相关信息。

JWTAuthorizationFilter 类：继承了 BasicAuthenticationFilter 类，设置拦截器，对前端吃的请求信息进行拦截，判断是否有 Authorization 信息，无则放行，有则获取用户信息。

3.6 Testing

本项目基于 Spring 框架提供的测试模块 spring-test，针对 Service 层进行单元测试，针对 Controller 层进行集成测试

3.6.1 单元测试

本项目针对 Service 层进行单元测试，实现了 BookServiceTest 类，分别初始化实例并实现了对数据库增删改查功能的测试，共进行了 5 个测试，通过 5 个，未通过 0 个，测试通过

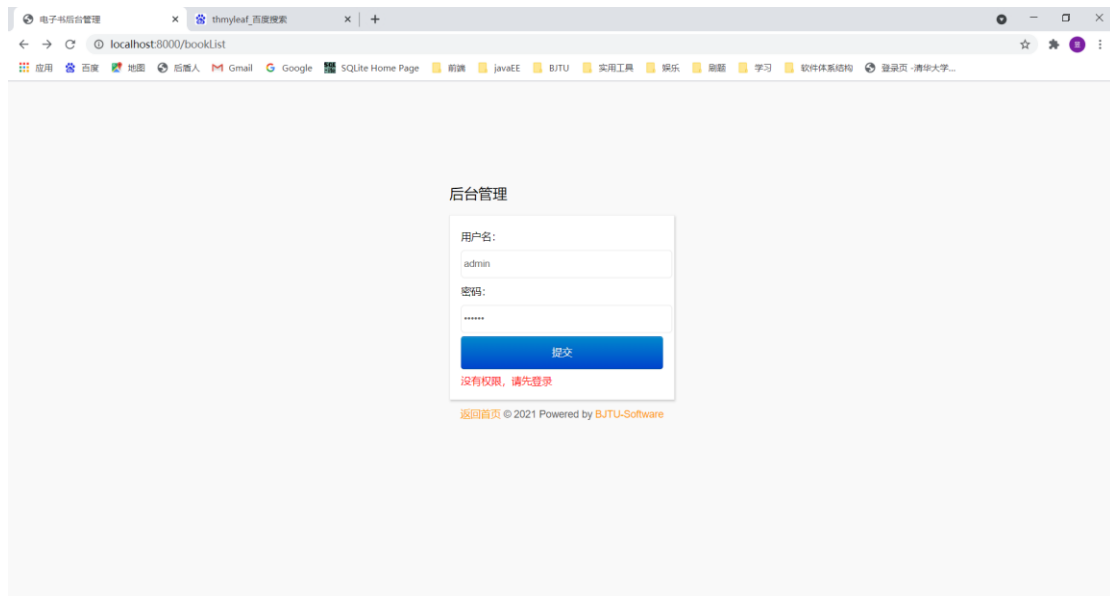
3.7 前端

3.7.1 实现步骤

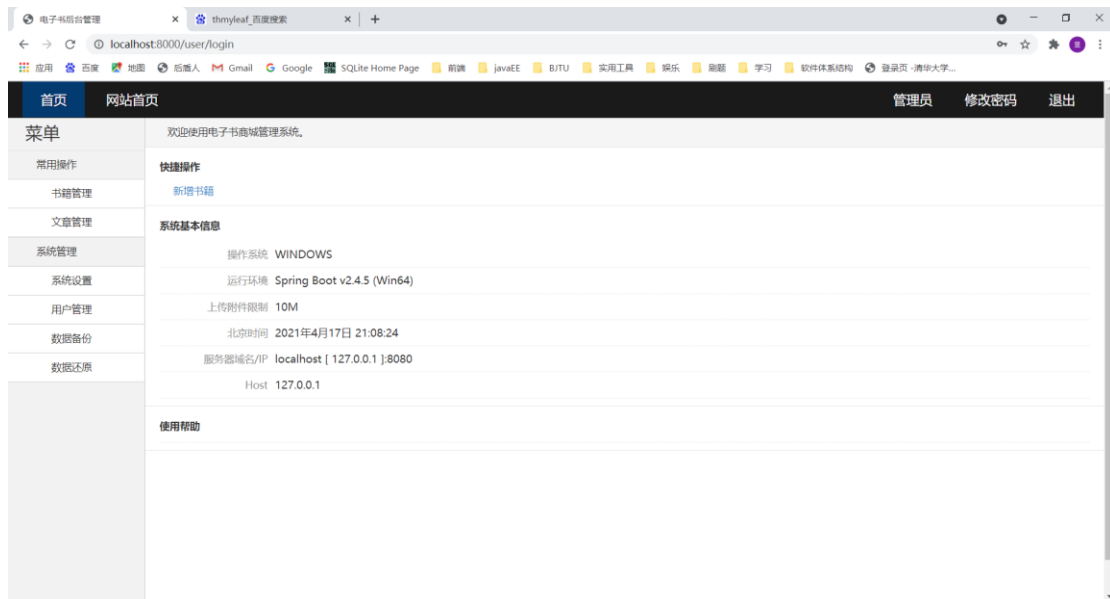
- 1.使用上次的 HTML 模板
- 2.通过 vue-cli 创建前端工程项目
- 3.改造模板为 vue 形式，同时提出共同部分和特殊部分形成组件
- 4.通过 vue-router 实现前端导航
- 5.通过 axios 实现异步发送 http 请求

3.7.2 界面

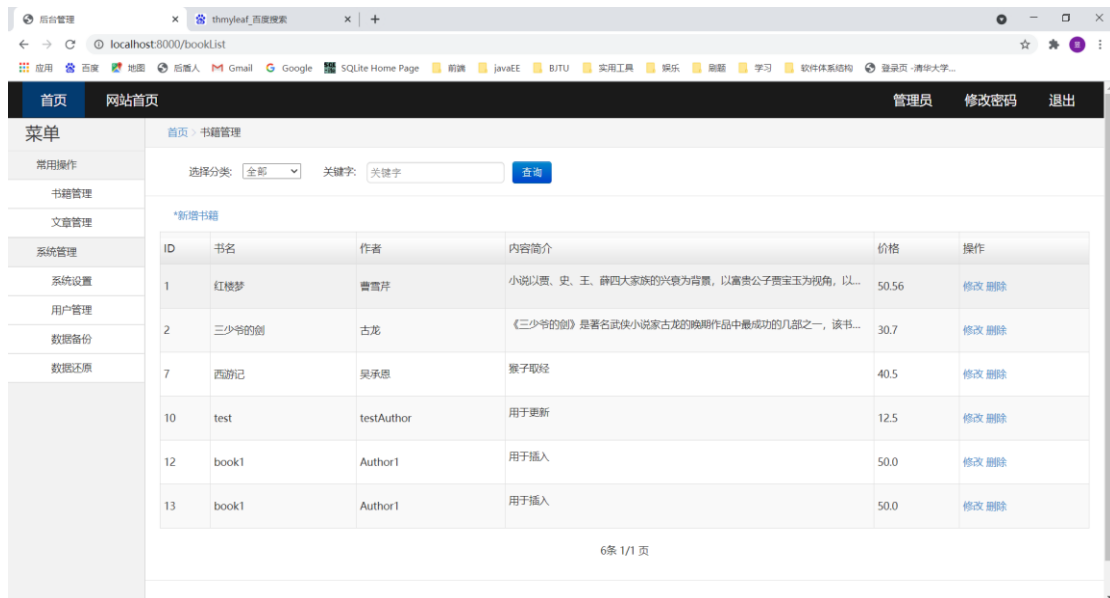
1.登录界面 (login.vue)



1. 主界面(index.vue)



2. 书籍列表界面(bookList.vue)



3. 增加、修改书籍界面(insert.vue,modify.vue)

