程序运行说明

班级：软硕2501班，软硕2503班

产品主题：前后端分离的图书管理系统

小组长：代武君-软硕2501班-M202577013

# 1 程序结构说明

程序主体分为以Vue框架为核心的前端，和以Spring Boot框架为核心的后端。其中前端代码位于项目根目录下的wj-vue文件夹，后端代码位于项目根目录下的wj文件夹。

后端核心程序位于wj\src\main\java\com\gm\wj文件夹中，其中各文件和目录的简要说明如下：

|  |  |
| --- | --- |
| **文件** | **功能说明** |
| WjApplication.java | Spring Boot应用启动类，包含main方法，用于启动后端 |
| config/ | 配置类目录，用于存放各种Spring配置类 |
| controller/ | 控制器层，接收前端HTTP请求，调用Service层处理业务逻辑，并将结果返回给前端 |
| service/ | 服务层，包含核心业务逻辑 |
| dao/ | 数据访问层，负责与数据库进行交互，执行CRUD操作 |
| entity/ | 实体类目录，其中的类与数据库表结构一一对应 |
| dto/ | 数据传输对象目录，用于在Controller和前端之间传输数据 |
| result/ | 统一响应结果封装目录，用于规范所有Controller返回给前端的JSON格式 |
| error/ | 存放错误页面(404.html)，在前端请求无法响应时返回错误页面 |
| util/ | 存放项目通用的工具类 |
| redis/ | 存放实现与Redis缓存交互的类 |
| realm/ | Shiro框架目录,负责实现用户权限管理 |

前端核心程序位于wj-vue\src文件夹中，其中各文件和目录的简要说明如下：

|  |  |
| --- | --- |
| **文件夹** | **功能说明** |
| assets/ | 静态资源目录，存放图片、字体、样式等静态文件 |
| components/ | 组件目录，存放Vue组件 |
| router/ | 路由配置目录，定义前端路由规则和页面导航逻辑 |
| styles/ | 全局样式目录，存放CSS/Less/Sass等样式文件 |
| utils/ | 工具函数目录，存放项目通用的工具函数 |
| App.vue | Vue应用根组件，作为所有组件的容器，定义全局页面布局 |
| main.js | Vue应用入口文件，初始化Vue实例并挂载到DOM，加载全局插件和配置 |

# 2 程序运行方法

### 2.1 项目信息

项目地址：[github](https://github.com/Antabot/White-Jotter?tab=readme-ov-file)

项目blog：[csdn](https://blog.csdn.net/Neuf_Soleil/article/details/88925013)，含全部的功能和开发过程介绍

### 2.2 运行环境

系统：windows10

依赖：

* [redis5.0.14.1](https://github.com/tporadowski/redis/releases)
* idea2025.2.1
* [jdk1.8](https://www.oracle.com/java/technologies/downloads/#java8:~:text=Java%2011-,Java%208,-Java%208%20Enterprise)（版本强制）
* [nodejs10.24.1](https://nodejs.org/dist/v10.24.1/node-v10.24.1-x64.msi)(版本强制)
* npm6.9.0(版本强制，nodejs10.24.1自带)
* mysql

### 2.3 部署步骤

**2.3.1 启动辅助模块**

1. 在mysql创建一个名为wj的数据库
2. 运行redis-server.exe启动redis服务
   * 1. **启动后端**
3. idea打开源代码的wj文件夹
4. 在file->project structure配置JKD为1.8
5. 进入wj/src/main/resources/application.properties，在spring.datasource. password填写mysql的登录密码
6. 运行wj/src/main/java/com/gm/wj/WjApplication.java
   * 1. **启动前端**
7. idea打开源代码的wj-vue文件夹
8. 打开idea左下角的terminal，依次执行如下命令（不会出现任何报错）

npm install

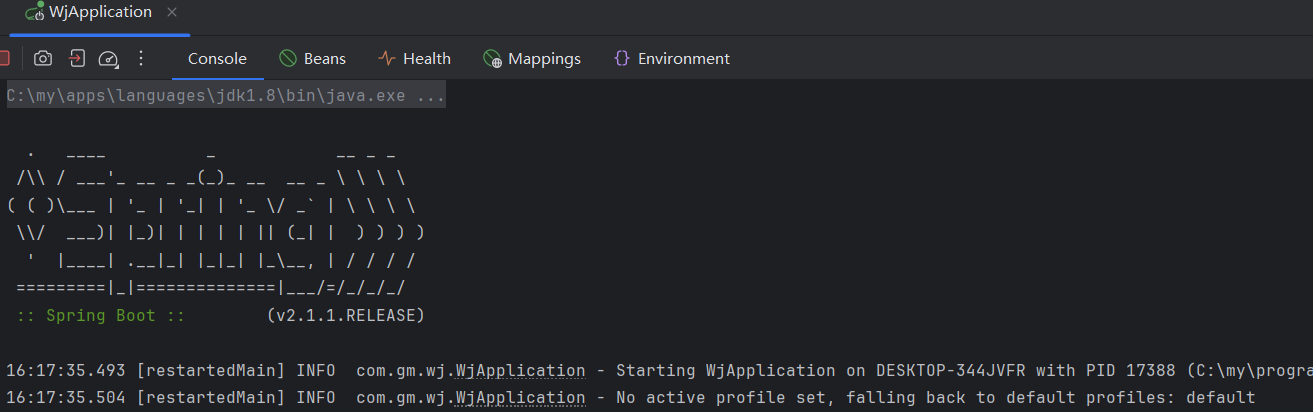
npm rebuild node-sass

npm run dev

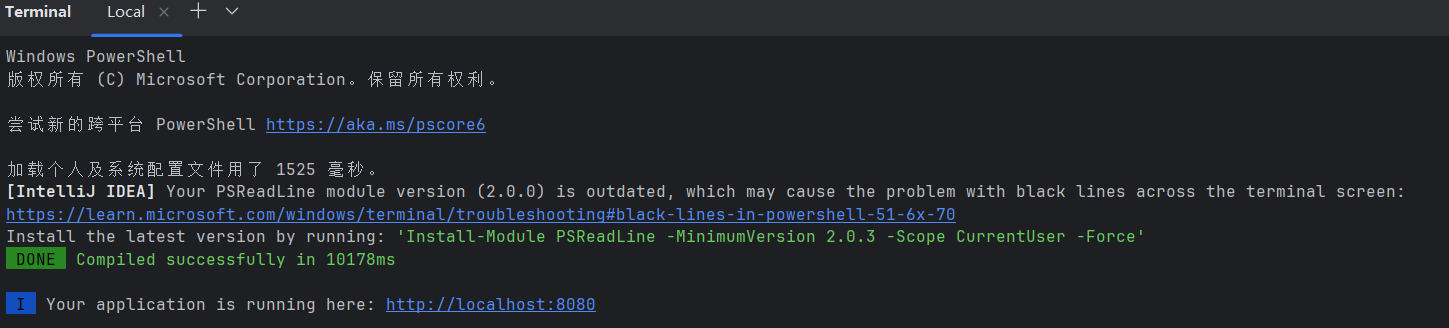
1. 浏览器访问http://localhost:8080/检查用户前端是否成功启动
2. 访问http://localhost:8080/admin/dashboard检查管理员前端和后端是否启动（账号admin，密码123）

# 3 程序运行效果

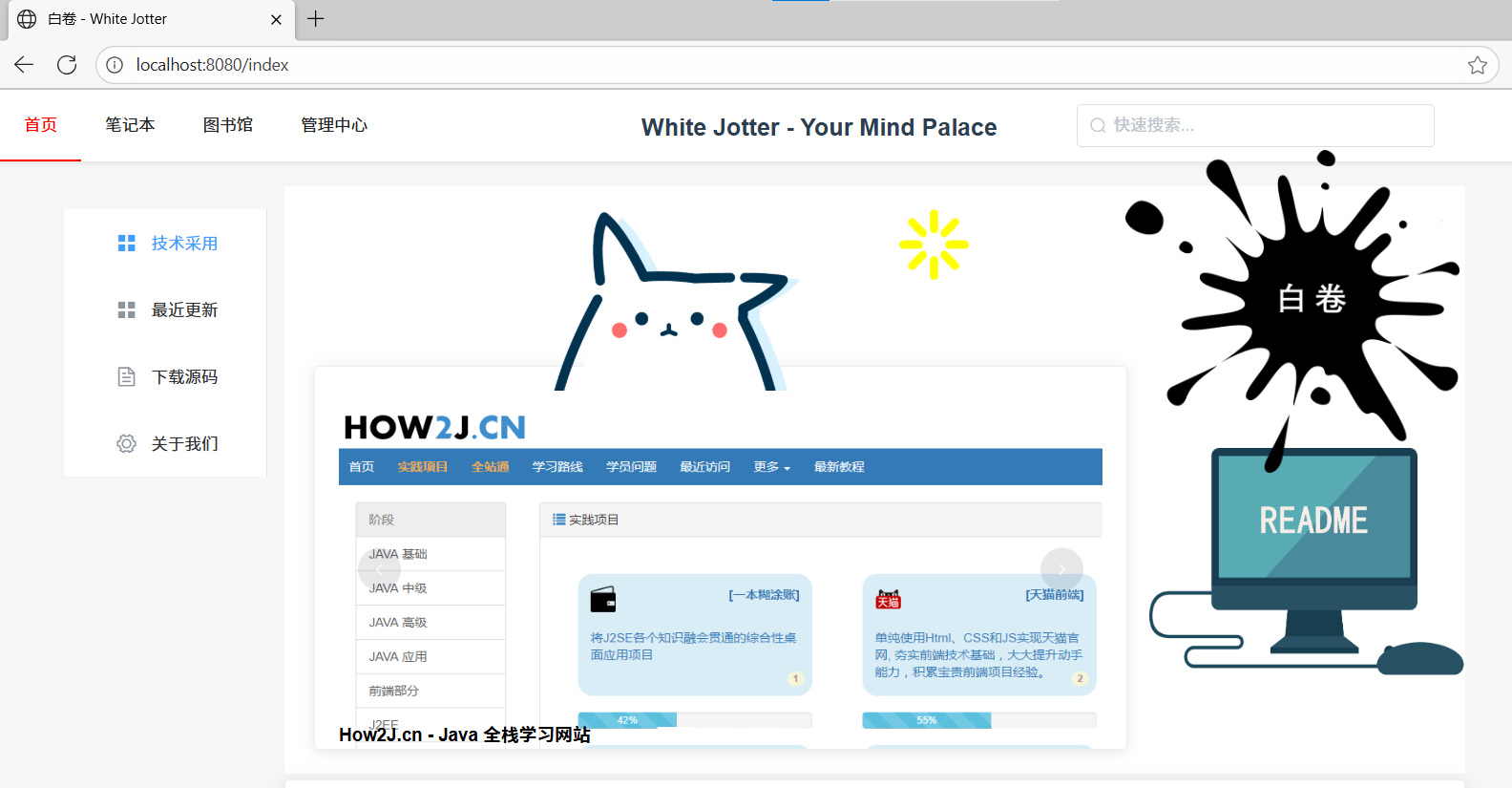
后端代码运行效果：



前端代码运行效果：



用户前端界面：



管理员前端界面：

