

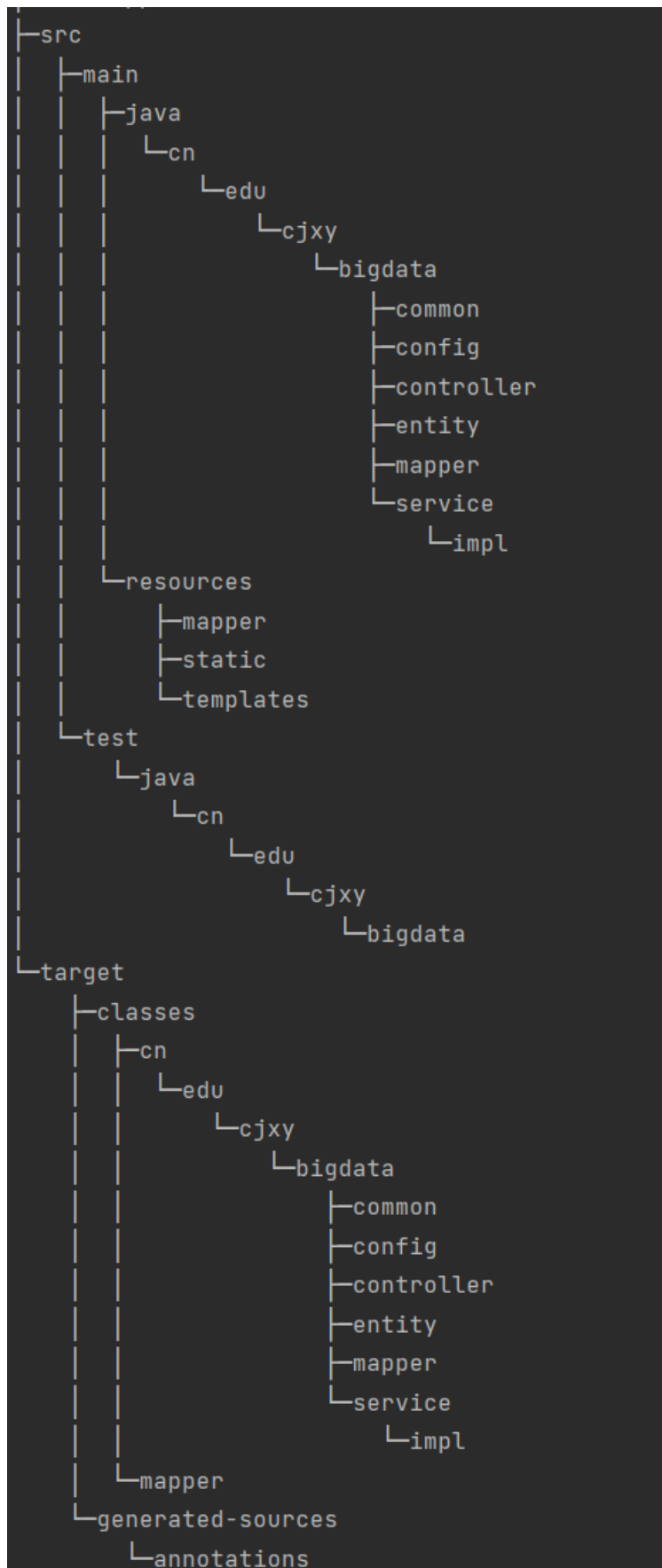
豆瓣图书网页前后端

项目概述

- 前端:[艾丽思]
- 后端: `Springboot` `JDBC` `java`
- 数据库: `mysql`

后端部分

项目结构



项目版本

1. Spring Initializr 项目

| Key | Value | Key | Value |
|-------|---------|-----|-------------|
| 名称/工件 | bigdata | 组 | cn.edu.cjxy |
| 语言 | Java | JDK | 1.8 |
| 类型 | Maven | | |

2. 依赖项

| 包名 | 作用 |
|----------------------|---------|
| mybatis-plus | 数据库操作工具 |
| knife4j | 文档管理 |
| mysql-connector-java | 数据库依赖 |
| spring-boot-web | 后端主程序 |
| lombok | 开发工具 |

项目配置

1. 依赖代码

pom.xml

```
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus-boot-starter</artifactId>
  <version>3.4.1</version>
</dependency>
<dependency>
  <groupId>com.baomidou</groupId>
  <artifactId>mybatis-plus-generator</artifactId>
  <version>3.4.1</version>
</dependency>
<dependency>
  <groupId>com.github.xiaoymin</groupId>
  <artifactId>knife4j-spring-boot-starter</artifactId>
  <version>2.0.9</version>
</dependency>
```

2. 配置文件

application.yaml

```
spring:
  datasource:
    url: jdbc:mysql://127.0.0.1:3306/bigdata?
useUnicode=true&characterEncoding=UTF-8&userSSL=false&serverTimezone=GMT%2B8
    driver-class-name: com.mysql.cj.jdbc.Driver
    username: root
    password: 123456
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
```

3. 跨域处理

corsFilter.java

```
package cn.edu.cjxy.bigdata.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
import org.springframework.web.filter.CorsFilter;

@Configuration
public class corsFilter {
    @Bean
    public CorsFilter CorsFilter() {
        CorsConfiguration corsConfiguration = new CorsConfiguration();

        corsConfiguration.addAllowedOriginPattern("*");
        corsConfiguration.addAllowedHeader("*");
        corsConfiguration.addAllowedMethod("*");
        corsConfiguration.setAllowCredentials(true);

        UrlBasedCorsConfigurationSource urlBasedCorsConfigurationSource =
            new UrlBasedCorsConfigurationSource();

        urlBasedCorsConfigurationSource.registerCorsConfiguration("/**", corsConfiguration);

        return new CorsFilter(urlBasedCorsConfigurationSource);
    }
}
```

项目功能

1. 数据接口

EchartsController.java

```
package cn.edu.cjxy.bigdata.controller;

import cn.edu.cjxy.bigdata.entity.BookCategory;
import cn.edu.cjxy.bigdata.entity.BookNum;
import cn.edu.cjxy.bigdata.service.BookCategoryService;
import cn.edu.cjxy.bigdata.service.BookNumService;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@Api(tags = "图")
@RestController
```

```

@RequestMapping("/echarts")
public class EchartsController {
    @Autowired
    private BookNumService bookNumService;

    @ApiOperation("图书数据总数表")
    @GetMapping("/bookNum")
    public List<BookNum> bookNumList(){
        return bookNumService.list();
    }

    @Autowired
    private BookCategoryService bookCategoryService;

    @ApiOperation("图书大类统计表")
    @GetMapping("/bookCategory")
    public List<BookCategory> bookCategoryList(){
        return bookCategoryService.list();
    }
}

```

2. 分页与查询

TablesController.java

```

package cn.edu.cjxy.bigdata.controller;

import cn.edu.cjxy.bigdata.common.QueryPageParam;
import cn.edu.cjxy.bigdata.entity.BookList;
import cn.edu.cjxy.bigdata.service.BookListService;
import com.baomidou.mybatisplus.core.conditions.query.LambdaQueryWrapper;
import com.baomidou.mybatisplus.core.metadata.IPage;
import com.baomidou.mybatisplus.extension.plugins.pagination.Page;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.HashMap;
import java.util.List;

@Api(tags = "表")
@RestController
@RequestMapping("/tables")
public class TablesController {

    @Autowired
    private BookListService bookListService;

    @ApiOperation("最终推荐书单统计表")
    @GetMapping("/bookList")
    public List<BookList> bookListALL(){
        return bookListService.list();
    }

    @ApiOperation("最终推荐书单统计表分页")
    @PostMapping("/bookListPage")
    public List<BookList> bookListPage(@RequestBody QueryPageParam query){

```

```
HashMap book = query.getBook();
String bookName = (String)book.get("bookName");
System.out.println((String)book.get("bookName"));

Page<BookList> page = new Page();
page.setCurrent(query.getPageNum());
page.setSize(query.getPageSize());

LambdaQueryWrapper<BookList> lambdaQueryWrapper = new
LambdaQueryWrapper();
lambdaQueryWrapper.like(BookList::getBookName, bookName);

IPage result = bookListService.page(page, lambdaQueryWrapper);

System.out.println(result.getTotal());

return result.getRecords();
}
}
```

3. 登录与注册

UserController.java

项目部署

1. 在idea中将代码打成jar包
2. 在cmd中运行jar

```
java -jar jar包名
```