E-Book Shop

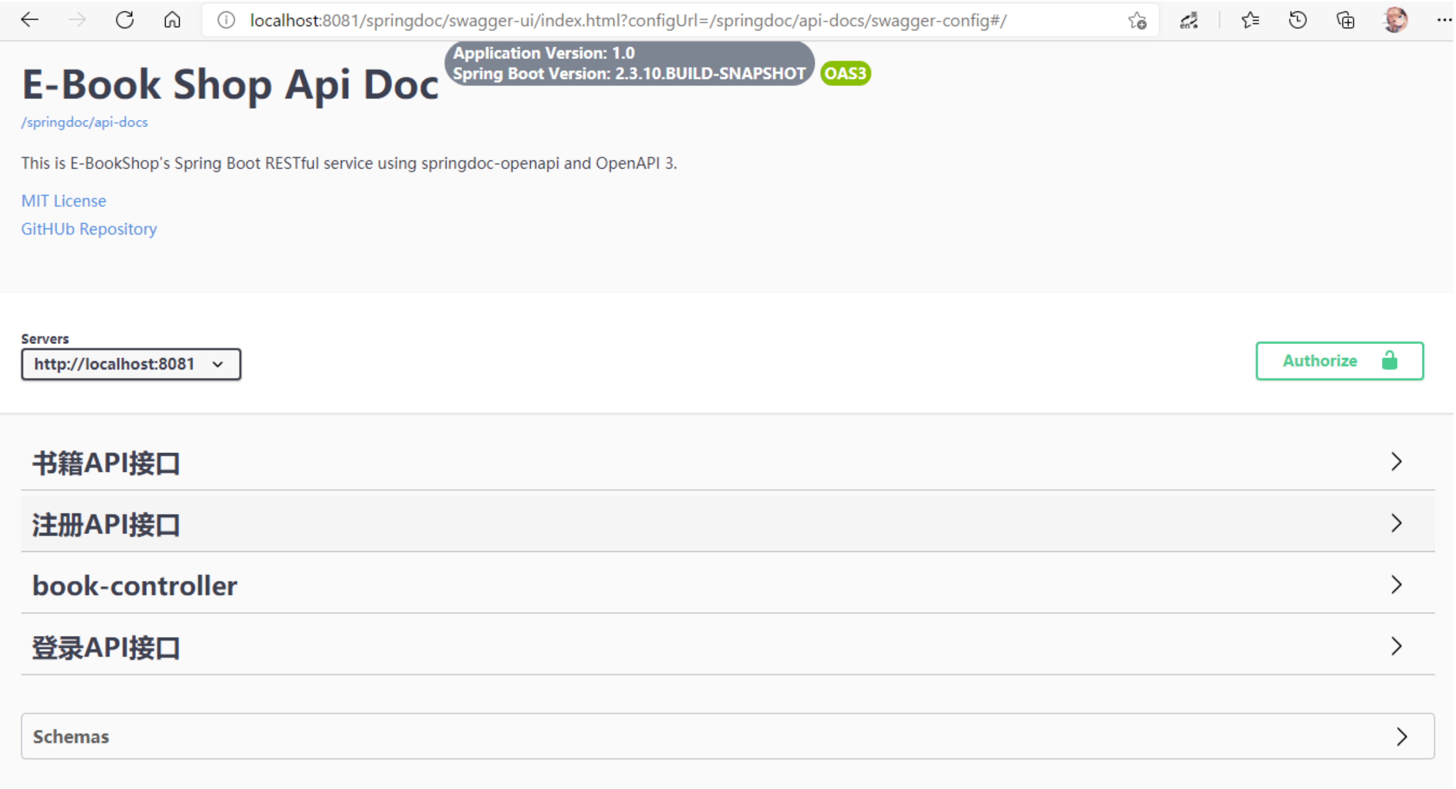
> E-BookShop Project for Spring Boot with Spring Security, Alibaba Druid, Thymeleaf, MySQL, Springdoc

Project Structure

```
├─.gitignore
├─.idea
├─.mvn
├─ebookshop.iml
├─LICENSE ------------------------------------------ // MIT License
├─mvnw
├─mvnw.cmd
├─pom.xml
├─README.md ---------------------------------------- // README文档
├─src
│ ├─main
│ │ ├─java
│ │ │ └─com
│ │ │   └─aruoxi
│ │ │     └─ebookshop
│ │ │       ├─common
│ │ │       │ └─CommonResult.java
│ │ │       ├─config ------------------------------- // 所有配置类
│ │ │       │ ├─ApplicationInitializer.java -------- // 程序初始化
│ │ │       │ ├─ExceptionHandle.java --------------- // 全局异常拦截器
│ │ │       │ ├─MyProps.java ----------------------- // 读取特定配置用于初始化
│ │ │       │ ├─SecurityConfiguration.java --------- // Spring Security配置类
│ │ │       │ └─springdoc -------------------------- // springdoc配置类
│ │ │       ├─controller --------------------------- // 路由控制器
│ │ │       │ ├─BookController.java
│ │ │       │ ├─dto -------------------------------- // 数据传输对象类集合
│ │ │       │ ├─LoginController.java
│ │ │       │ ├─MainController.java
│ │ │       │ ├─restController
│ │ │       │ └─UserRegistrationController.java
│ │ │       ├─domain ------------------------------- // 实体类
│ │ │       │ ├─Book.java
│ │ │       │ ├─QBook.java ------------------------- // Book属性类
│ │ │       │ ├─Role.java
│ │ │       │ └─User.java
│ │ │       ├─EbookshopApplication.java ------------ // 主程序启动类
│ │ │       ├─exception ---------------------------- // 自定义异常类
│ │ │       ├─repository --------------------------- // repository
│ │ │       │ ├─BookRepository.java
│ │ │       │ ├─RoleRepository.java
│ │ │       │ └─UserRepository.java
│ │ │       └─service ------------------------------ // 服务类
│ │ │         ├─BookService.java
│ │ │         ├─impl ------------------------------- // 服务的具体实现类
│ │ │         │ ├─BookServiceImpl.java
│ │ │         │ ├─RoleServiceImpl.java
│ │ │         │ └─UserServiceImpl.java
│ │ │         ├─RoleService.java
│ │ │         └─UserService.java
│ │ └─resources
│ │   ├─application.yml ---------------------------- // 配置文件
│ │   ├─banner.txt --------------------------------- // 自定义banner
│ │   ├─data.sql ----------------------------------- // 初始化数据sql
│ │   ├─META-INF
│ │   ├─schema.sql --------------------------------- // 初始化数据结构sql
│ │   ├─static ------------------------------------- // 静态文件
│ │   │ ├─assets
│ │   │ ├─css
│ │   │ ├─images
│ │   │ ├─js
│ │   │ └─uploadFile ------------------------------- // 上传文件
│ │   └─templates ---------------------------------- // 模板
│ │     ├─content.html
│ │     ├─home.html
│ │     ├─login.html
│ │     └─registration.html
│ └─test ------------------------------------------- // 测试类
│   └─java
│     └─com
│       └─aruoxi
│         └─ebookshop
│           ├─ApplicationTests.java
│           ├─dto
│           └─EbookshopApplicationTests.java
└─target
```

API Doc

http://localhost:8081/springdoc/index.html


localhost:8081/springdoc/swagger-ui/index.html?configUrl=/springdoc/api-docs/swagger-config#/
OAS3
Authorize
Schemas

E-Book Shop Api Doc

Application Version: 1.0
Spring Boot Version: 2.3.10.BUILD-SNAPSHOT

/springdoc/api-docs

This is E-BookShop's Spring Boot RESTful service using springdoc-openapi and OpenAPI 3.

MIT License
GitHUb Repository

Servers
http://localhost:8081

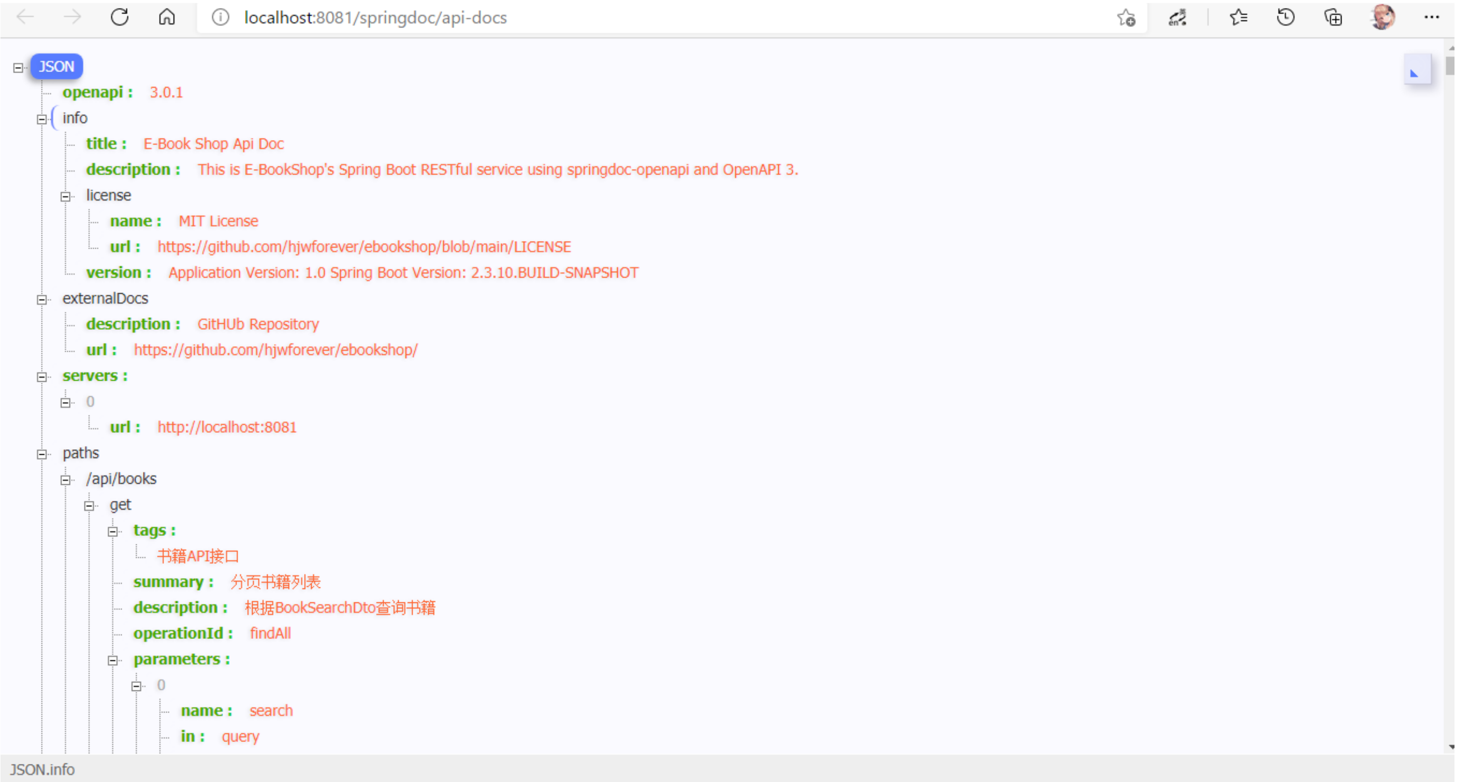
书籍API接口

注册API接口

book-controller

登录API接口

http://localhost:8081/springdoc/api-docs



localhost:8081/springdoc/api-docs

```
JSON
  openapi :   3.0.1
  info
    title :   E-Book Shop Api Doc
    description :   This is E-BookShop's Spring Boot RESTful service using springdoc-openapi and OpenAPI 3.
    license
      name :   MIT License
      url :   https://github.com/hjwforever/ebookshop/blob/main/LICENSE
    version :   Application Version: 1.0 Spring Boot Version: 2.3.10.BUILD-SNAPSHOT
  externalDocs
    description :   GitHUb Repository
    url :   https://github.com/hjwforever/ebookshop/
  servers :
    0
      url :   http://localhost:8081
  paths
    /api/books
      get
        tags :
          书籍API接口
        summary :   分页书籍列表
        description :   根据BookSearchDto查询书籍
        operationId :   findAll
        parameters :
          0
            name :   search
            in :   query
```

JSON.info

E-Book Shop 功能文档

> 详情可看**附件设计文档**或**项目在线接口文档**

基本功能

- 登录
- 注册
- 看书
- 上传
- 音乐播放器
- 书籍的增删改查(API)

进阶功能

1. 分页+ajax局部刷新

- 书籍列表分页 及 局部刷新
- 阅读书籍分页 及 局部刷新

2. 拦截器及过滤器 (intercepters/filters)

- 路由拦截器
- 全局异常拦截器
- 登录跳转至首页

3. 集成Spring Security (Authentication and authorization)

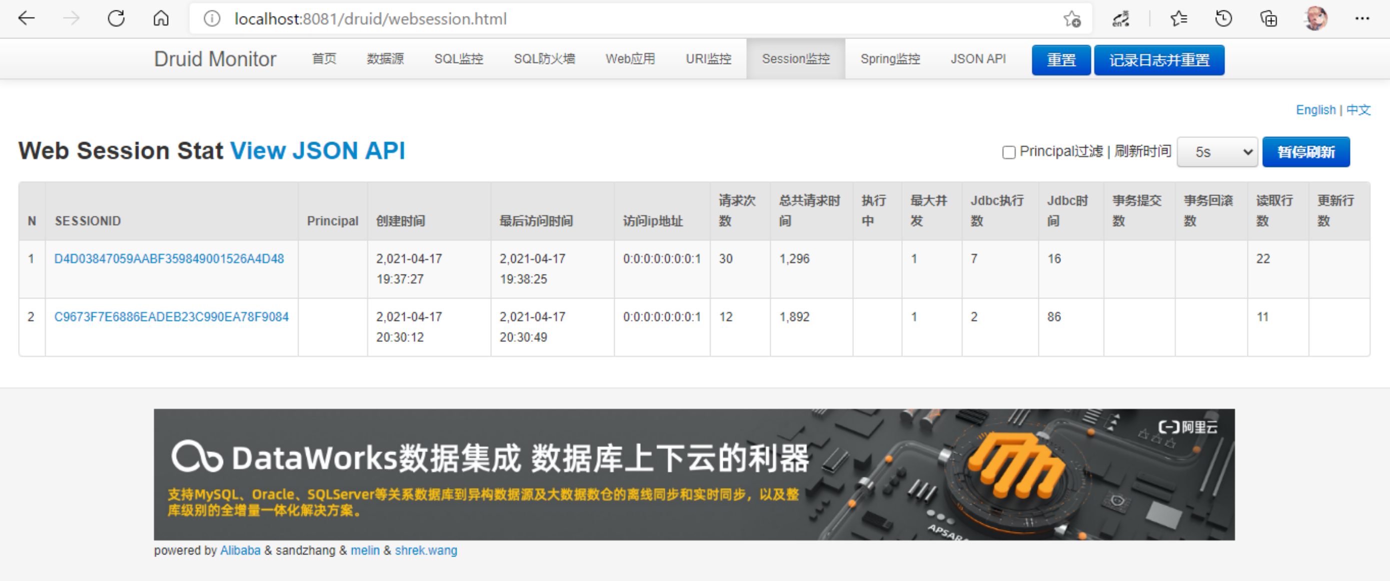
- 登录注册
- 权限分级
- 密码加密
- 路由拦截

4. 权限分级

- 游客(未登录用户)只能浏览书籍列表，不可阅读及下载
- 登录用户(User) 可以阅读及下载书籍, 但不可上传书籍
- 管理员(Admin) 拥有以上所有权限， 并可上传及修改书籍 （增删改查）

集成功能

1. SpringSecurity
2. Druid数据库监控(访问路径 http:/localhost:8081/druid)



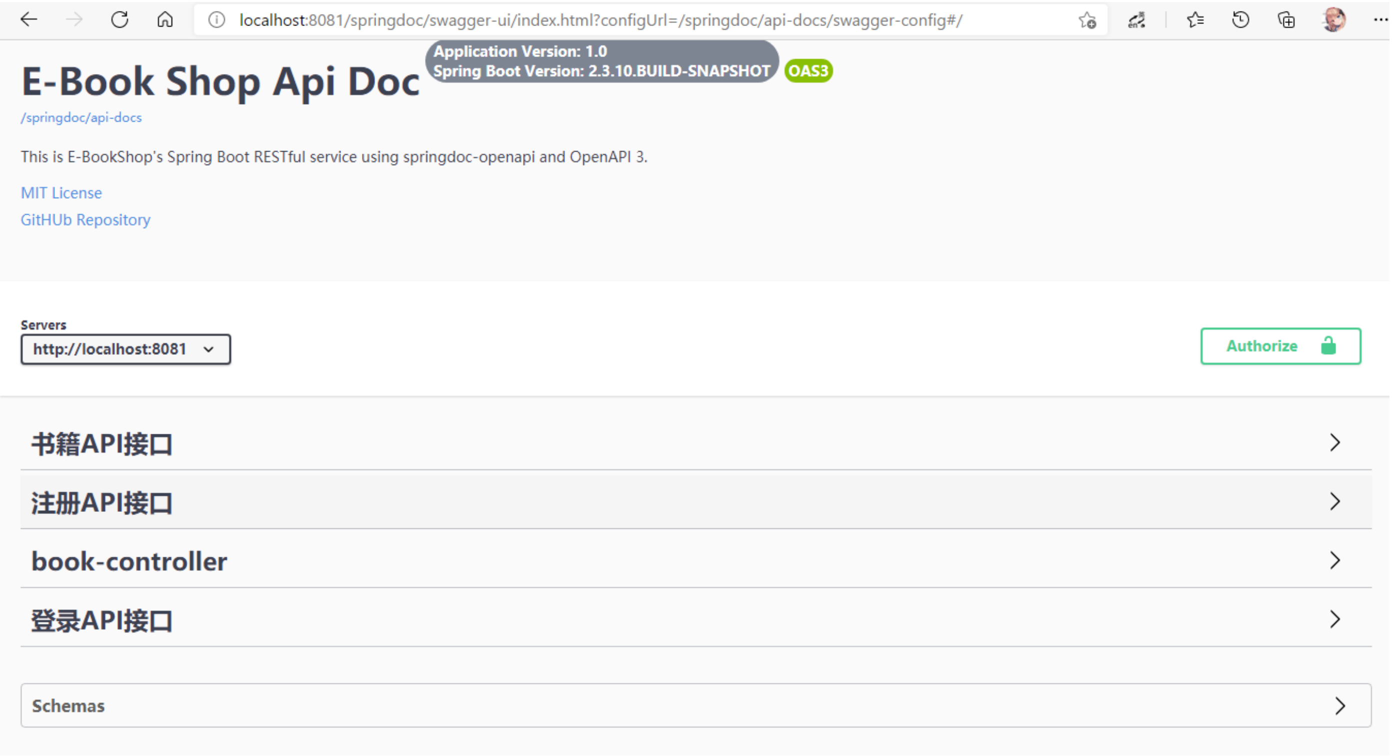
Web Session Stat View JSON API

N	SESSIONID	Principal	创建时间	最后访问时间	访问ip地址	请求次数	总共请求时间	执行中	最大并发	Jdbc执行数	Jdbc时间	事务提交数	事务回滚数	读取行数	更新行数
1	D4D03847059AABF359849001526A4D48		2,021-04-17 19:37:27	2,021-04-17 19:38:25	0:0:0:0:0:0:0:1	30	1,296		1	7	16			22	
2	C9673F7E6886EADEB23C990EA78F9084		2,021-04-17 20:30:12	2,021-04-17 20:30:49	0:0:0:0:0:0:0:1	12	1,892		1	2	86			11	

powered by Alibaba & sandzhang & melin & shrek.wang

3. Springdoc在线接口文档

- Online Document: http://localhost:8081/springdoc/index.html



localhost:8081/springdoc/swagger-ui/index.html?configUrl=/springdoc/api-docs/swagger-config#/

E-Book Shop Api Doc

Application Version: 1.0
Spring Boot Version: 2.3.10.BUILD-SNAPSHOT

/springdoc/api-docs

This is E-BookShop's Spring Boot RESTful service using springdoc-openapi and OpenAPI 3.

MIT License

GitHUb Repository

Servers: http://localhost:8081

书籍API接口

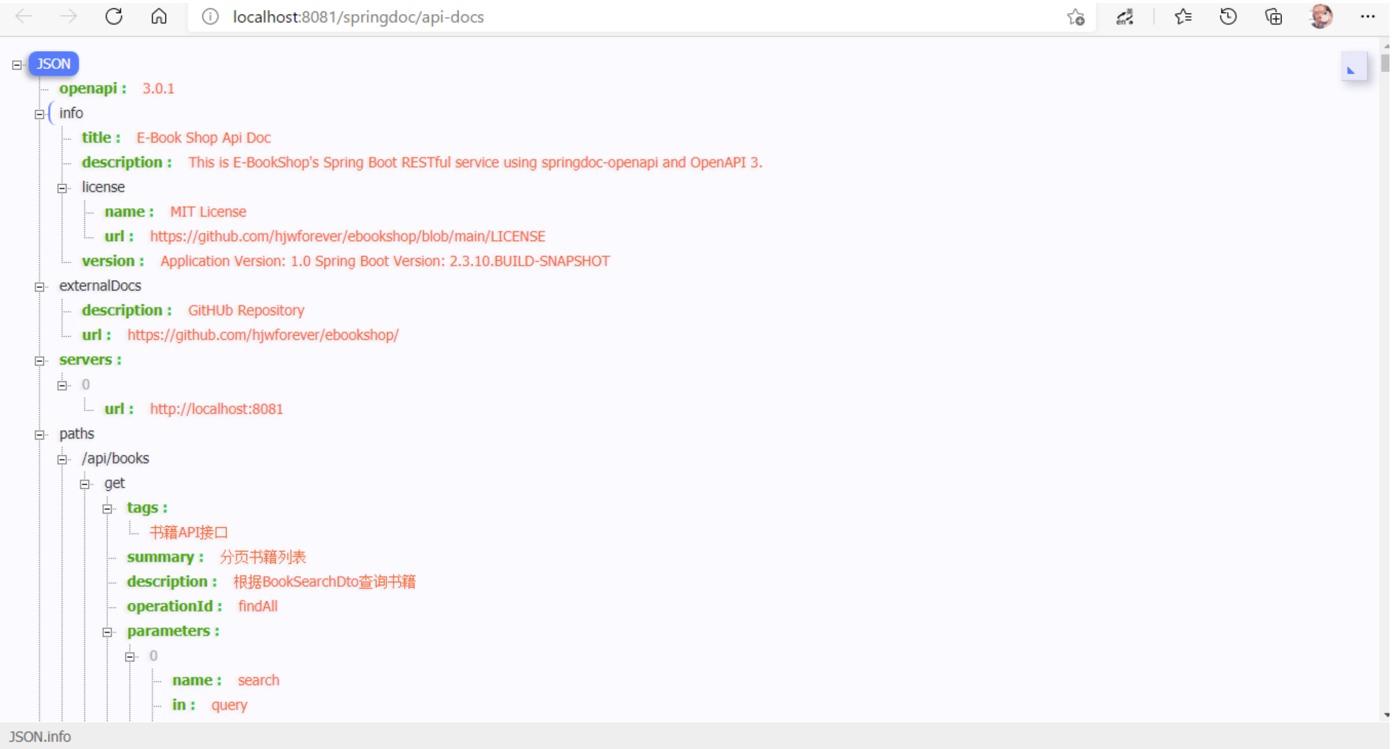
注册API接口

book-controller

登录API接口

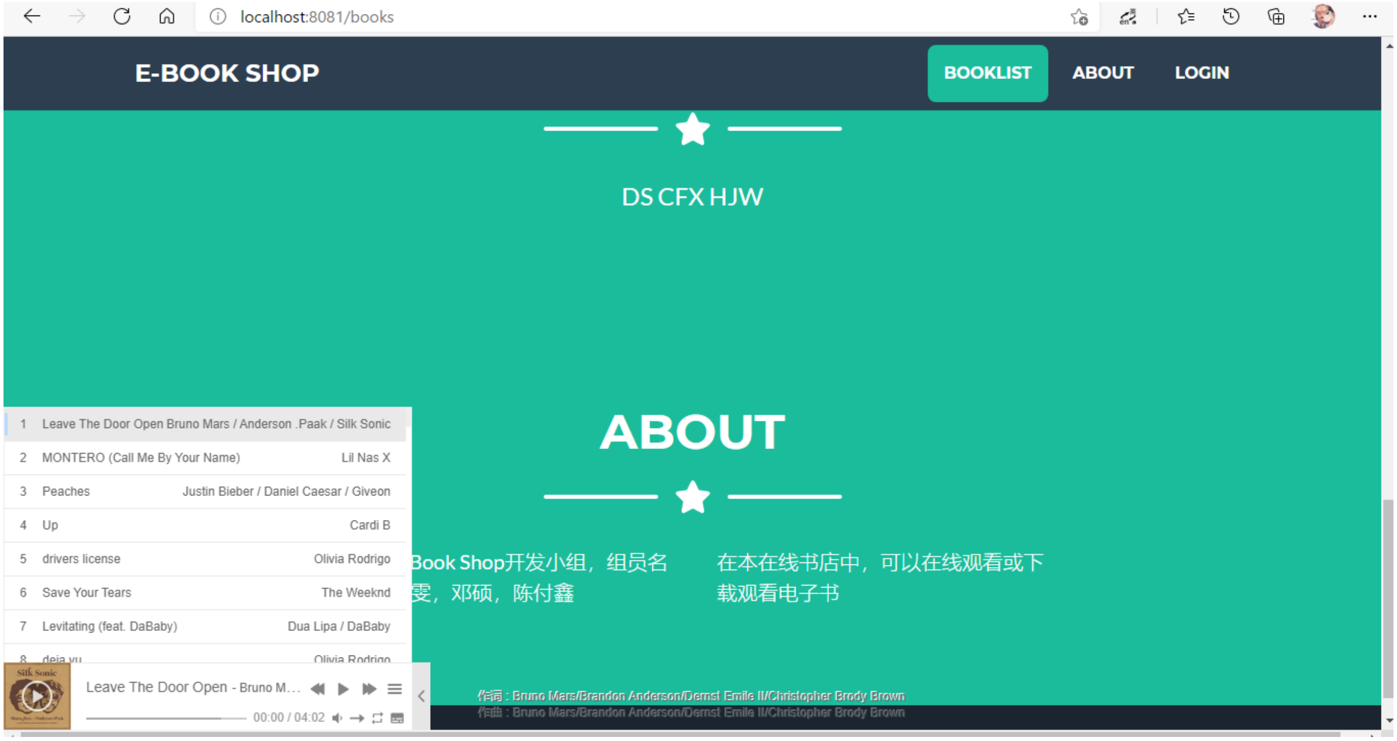
Schemas

- JSON Document: http://localhost:8081/springdoc/api-docs



```
JSON
  openapi : 3.0.1
  info
    title : E-Book Shop Api Doc
    description : This is E-BookShop's Spring Boot RESTful service using springdoc-openapi and OpenAPI 3.
    license
      name : MIT License
      url : https://github.com/hjwforever/ebookshop/blob/main/LICENSE
    version : Application Version: 1.0 Spring Boot Version: 2.3.10.BUILD-SNAPSHOT
  externalDocs
    description : GitHUb Repository
    url : https://github.com/hjwforever/ebookshop/
  servers :
    0
      url : http://localhost:8081
  paths
    /api/books
      get
        tags :
          书籍API接口
        summary : 分页书籍列表
        description : 根据BookSearchDto查询书籍
        operationId : findAll
        parameters :
          0
            name : search
            in : query
```

4. 音乐播放器



DS CFX HJW

ABOUT

Book Shop开发小组，组员名雯，邓硕，陈付鑫

在本在线书店中，可以在线观看或下载观看电子书

1	Leave The Door Open	Bruno Mars / Anderson .Paak / Silk Sonic
2	MONTERO (Call Me By Your Name)	Lil Nas X
3	Peaches	Justin Bieber / Daniel Caesar / Giveon
4	Up	Cardi B
5	drivers license	Olivia Rodrigo
6	Save Your Tears	The Weeknd
7	Levitating (feat. DaBaby)	Dua Lipa / DaBaby
8	deja vu	Olivia Rodrigo

Leave The Door Open - Bruno M…

00:00 / 04:02

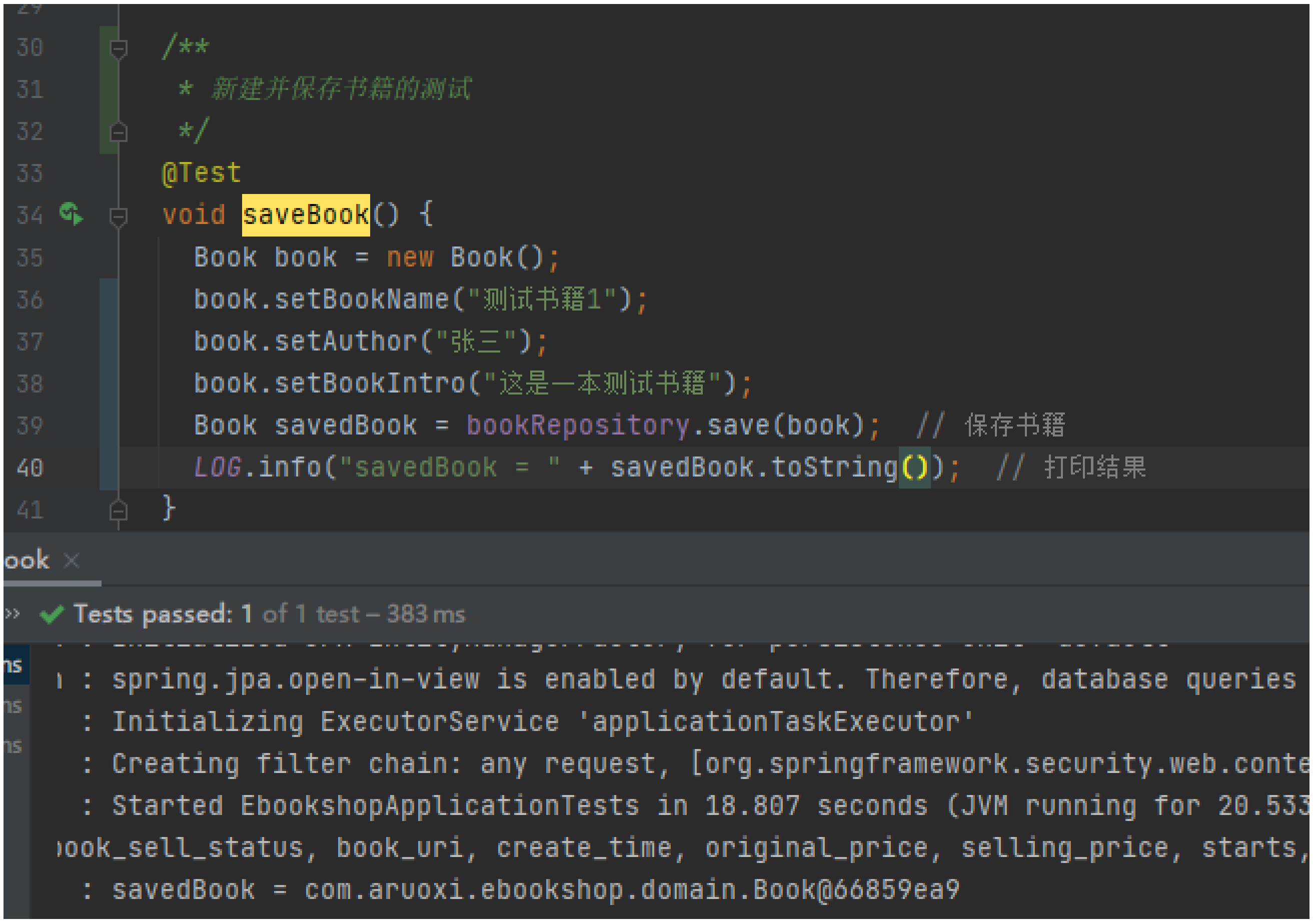
作词 : Bruno Mars/Brandon Anderson/Dernst Emile II/Christopher Brody Brown

作曲 : Bruno Mars/Brandon Anderson/Dernst Emile II/Christopher Brody Brown

Unit Test

Repository Test

新建并保存书籍

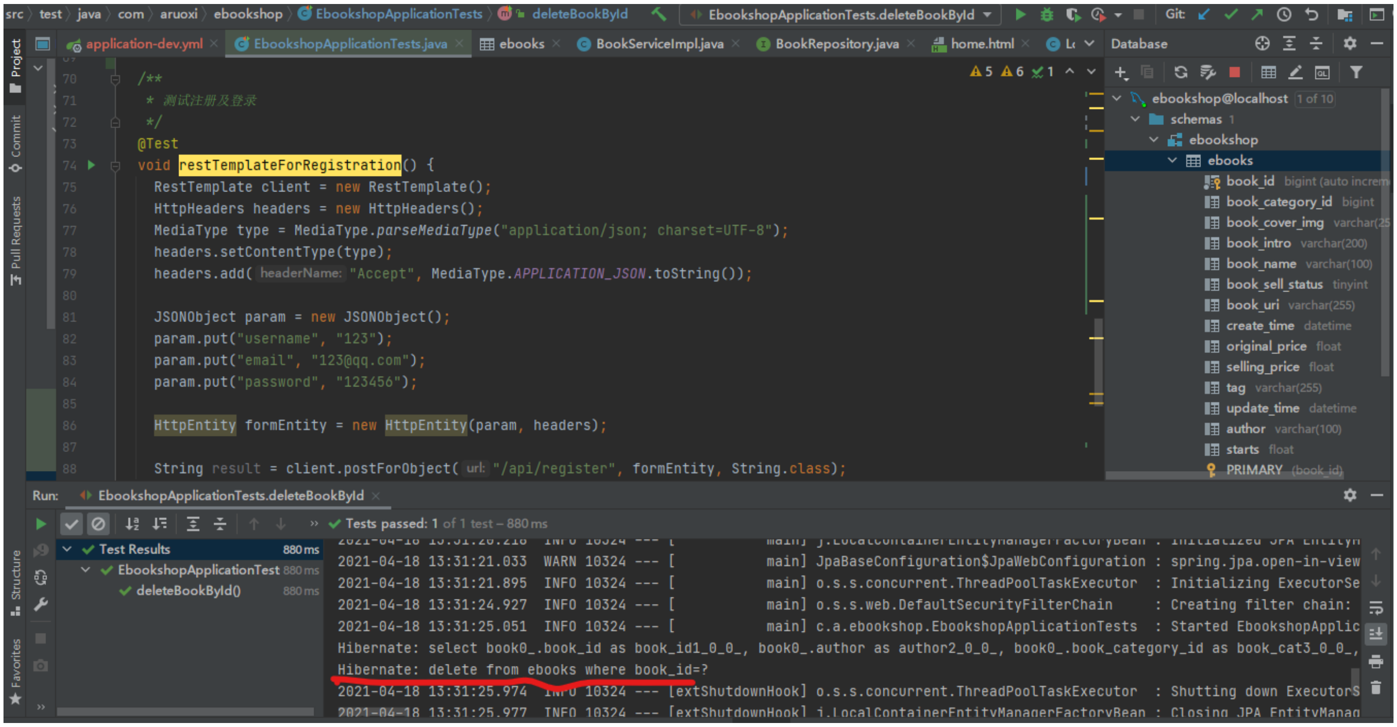


```
    /**
     * 新建并保存书籍的测试
     */
    @Test
    void saveBook() {
        Book book = new Book();
        book.setBookName("测试书籍1");
        book.setAuthor("张三");
        book.setBookIntro("这是一本测试书籍");
        Book savedBook = bookRepository.save(book);  // 保存书籍
        LOG.info("savedBook = " + savedBook.toString());  // 打印结果
    }
```

```
: spring.jpa.open-in-view is enabled by default. Therefore, database queries
: Initializing ExecutorService 'applicationTaskExecutor'
: Creating filter chain: any request, [org.springframework.security.web.conte
: Started EbookshopApplicationTests in 18.807 seconds (JVM running for 20.533
ook_sell_status, book_uri, create_time, original_price, selling_price, starts,
: savedBook = com.aruoxi.ebookshop.domain.Book@66859ea9
```

根据id删除书籍

- 根据id删除书籍

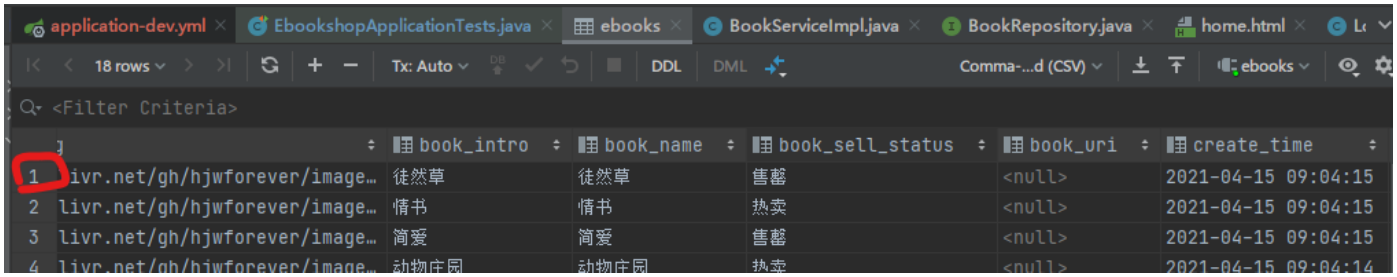


```
    /**
     * 测试注册及登录
     */
    @Test
    void restTemplateForRegistration() {
        RestTemplate client = new RestTemplate();
        HttpHeaders headers = new HttpHeaders();
        MediaType type = MediaType.parseMediaType("application/json; charset=UTF-8");
        headers.setContentType(type);
        headers.add( headerName: "Accept", MediaType.APPLICATION_JSON.toString());

        JSONObject param = new JSONObject();
        param.put("username", "123");
        param.put("email", "123@qq.com");
        param.put("password", "123456");

        HttpEntity formEntity = new HttpEntity(param, headers);

        String result = client.postForObject( url: "/api/register", formEntity, String.class);
```

Run: EbookshopApplicationTests.deleteBookById
Tests passed: 1 of 1 test – 880 ms
Test Results 880 ms
EbookshopApplicationTest 880 ms
deleteBookById() 880 ms

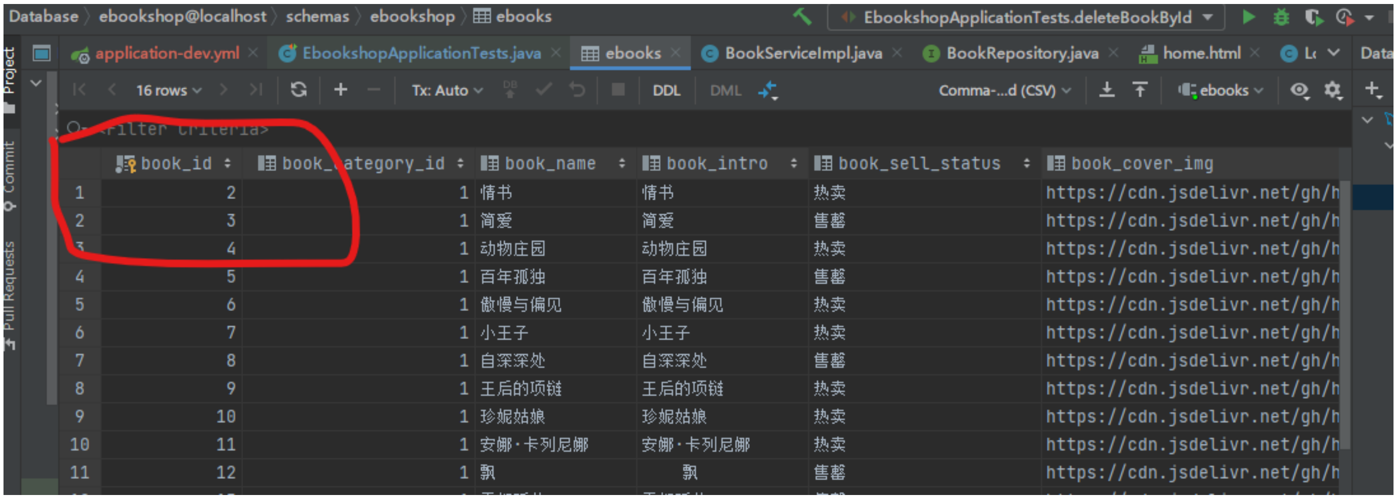
```
2021-04-18 13:31:21.033  WARN 10324 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view
2021-04-18 13:31:21.895  INFO 10324 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorSe
2021-04-18 13:31:24.927  INFO 10324 --- [           main] o.s.s.web.DefaultSecurityFilterChain     : Creating filter chain:
2021-04-18 13:31:25.051  INFO 10324 --- [           main] c.a.ebookshop.EbookshopApplicationTests  : Started EbookshopApplic
Hibernate: select book0_.book_id as book_id1_0_0_, book0_.author as author2_0_0_, book0_.book_category_id as book_cat3_0_0_,
Hibernate: delete from ebooks where book_id=?
2021-04-18 13:31:25.974  INFO 10324 --- [extShutdownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorS
2021-04-18 13:31:25.977  INFO 10324 --- [extShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManag
```

- **删除前** 数据库-ebooks-#1书籍**存在**



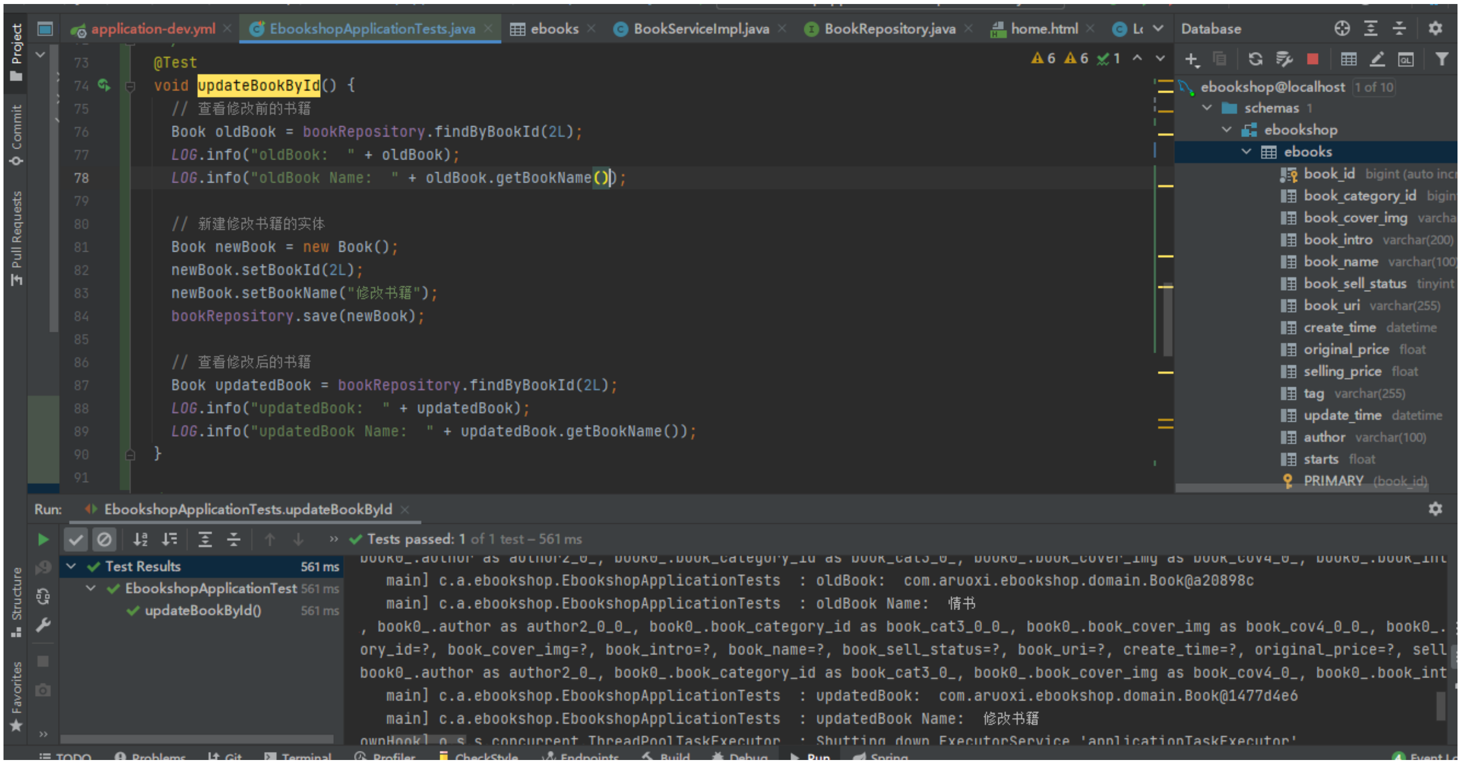
	j	book_intro	book_name	book_sell_status	book_uri	create_time
1	ivr.net/gh/hjwforever/image…	徒然草	徒然草	售罄	<null>	2021-04-15 09:04:15
2	livr.net/gh/hjwforever/image…	情书	情书	热卖	<null>	2021-04-15 09:04:15
3	livr.net/gh/hjwforever/image…	简爱	简爱	售罄	<null>	2021-04-15 09:04:15
4	livr.net/gh/hjwforever/image…	动物庄园	动物庄园	热卖	<null>	2021-04-15 09:04:14

- **删除后** 数据库-ebooks-#1书籍**不存在**



	book_id	book_category_id	book_name	book_intro	book_sell_status	book_cover_img
1	2	1	情书	情书	热卖	https://cdn.jsdelivr.net/gh/h
2	3	1	简爱	简爱	售罄	https://cdn.jsdelivr.net/gh/h
3	4	1	动物庄园	动物庄园	热卖	https://cdn.jsdelivr.net/gh/h
4	5	1	百年孤独	百年孤独	售罄	https://cdn.jsdelivr.net/gh/h
5	6	1	傲慢与偏见	傲慢与偏见	热卖	https://cdn.jsdelivr.net/gh/h
6	7	1	小王子	小王子	热卖	https://cdn.jsdelivr.net/gh/h
7	8	1	自深深处	自深深处	售罄	https://cdn.jsdelivr.net/gh/h
8	9	1	王后的项链	王后的项链	热卖	https://cdn.jsdelivr.net/gh/h
9	10	1	珍妮姑娘	珍妮姑娘	热卖	https://cdn.jsdelivr.net/gh/h
10	11	1	安娜·卡列尼娜	安娜·卡列尼娜	热卖	https://cdn.jsdelivr.net/gh/h
11	12	1	飘	飘	售罄	https://cdn.jsdelivr.net/gh/h

根据id修改书籍



```java
@Test
void updateBookById() {
    // 查看修改前的书籍
    Book oldBook = bookRepository.findByBookId(2L);
    LOG.info("oldBook:  " + oldBook);
    LOG.info("oldBook Name:  " + oldBook.getBookName());

    // 新建修改书籍的实体
    Book newBook = new Book();
    newBook.setBookId(2L);
    newBook.setBookName("修改书籍");
    bookRepository.save(newBook);

    // 查看修改后的书籍
    Book updatedBook = bookRepository.findByBookId(2L);
    LOG.info("updatedBook:  " + updatedBook);
    LOG.info("updatedBook Name:  " + updatedBook.getBookName());
}
```

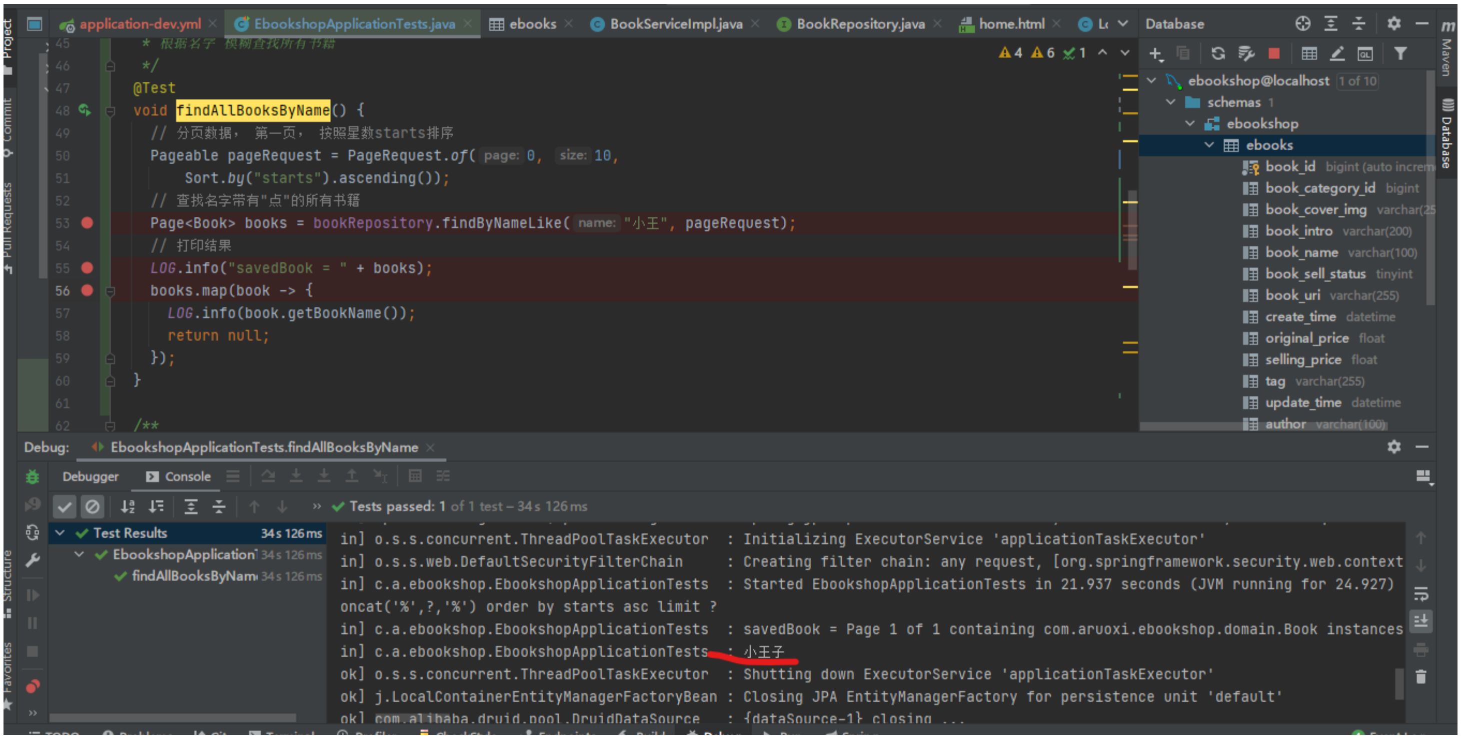
ebookshop@localhost 1 of 10
schemas 1
ebookshop
ebooks
book_id bigint (auto incr
book_category_id bigin
book_cover_img varcha
book_intro varchar(200)
book_name varchar(100
book_sell_status tinyint
book_uri varchar(255)
create_time datetime
original_price float
selling_price float
tag varchar(255)
update_time datetime
author varchar(100)
starts float
PRIMARY (book_id)

Tests passed: 1 of 1 test – 561 ms

Test Results 561 ms
EbookshopApplicationTest 561 ms
updateBookById() 561 ms

```
book0_.author as author2_0_, book0_.book_category_id as book_cat3_0_, book0_.book_cover_img as book_cov4_0_, book0_.book_int
   main] c.a.ebookshop.EbookshopApplicationTests  : oldBook:  com.aruoxi.ebookshop.domain.Book@a20898c
   main] c.a.ebookshop.EbookshopApplicationTests  : oldBook Name:  情书
, book0_.author as author2_0_0_, book0_.book_category_id as book_cat3_0_0_, book0_.book_cover_img as book_cov4_0_0_, book0_.
ory_id=?, book_cover_img=?, book_intro=?, book_name=?, book_sell_status=?, book_uri=?, create_time=?, original_price=?, sell
book0_.author as author2_0_, book0_.book_category_id as book_cat3_0_, book0_.book_cover_img as book_cov4_0_, book0_.book_int
   main] c.a.ebookshop.EbookshopApplicationTests  : updatedBook:  com.aruoxi.ebookshop.domain.Book@1477d4e6
   main] c.a.ebookshop.EbookshopApplicationTests  : updatedBook Name:  修改书籍
ownHook] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService 'applicationTaskExecutor'
```

根据 book_name 模糊查找所有相关书籍



```java
     * 根据名字 模糊查找所有书籍
     */
    @Test
    void findAllBooksByName() {
        // 分页数据， 第一页， 按照星数starts排序
        Pageable pageRequest = PageRequest.of( 0, 10,
            Sort.by("starts").ascending());
        // 查找名字带有"点"的所有书籍
        Page<Book> books = bookRepository.findByNameLike( "小王", pageRequest);
        // 打印结果
        LOG.info("savedBook = " + books);
        books.map(book -> {
          LOG.info(book.getBookName());
          return null;
        });
    }

    /**
```

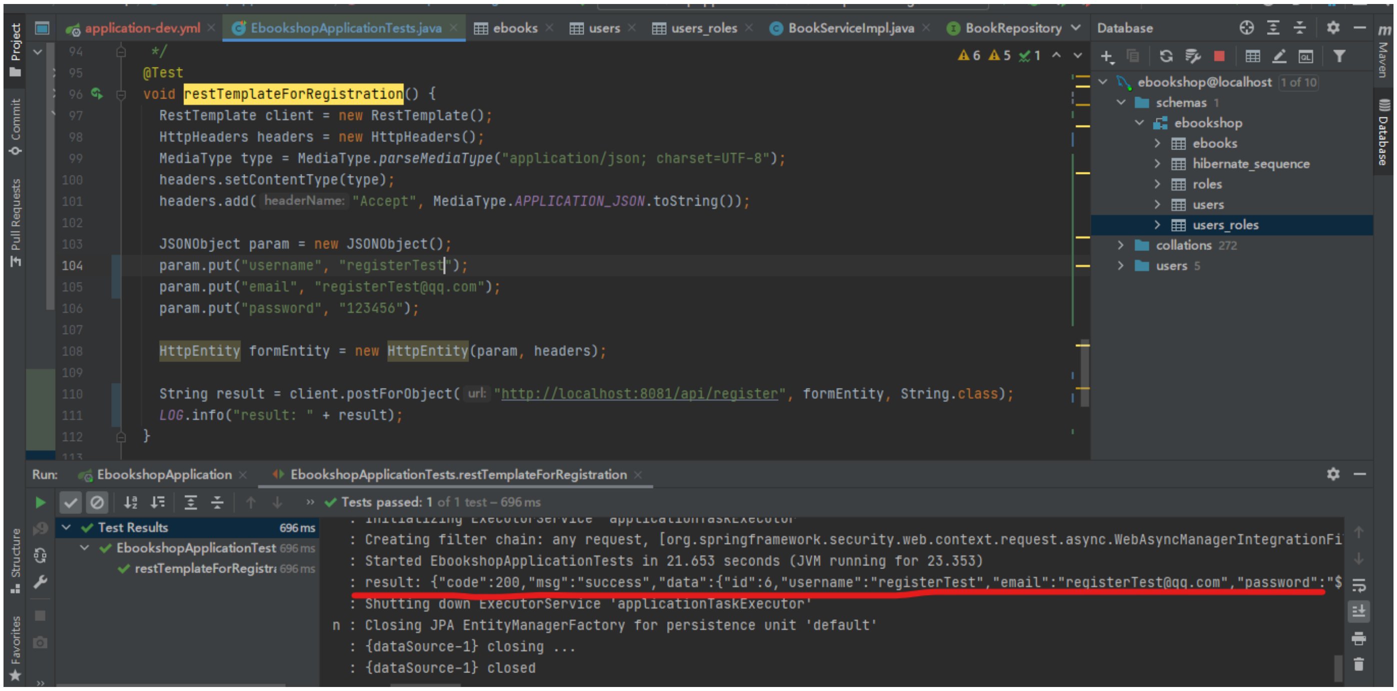
Tests passed: 1 of 1 test – 34 s 126 ms

```
in] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
in] o.s.s.web.DefaultSecurityFilterChain     : Creating filter chain: any request, [org.springframework.security.web.context
in] c.a.ebookshop.EbookshopApplicationTests  : Started EbookshopApplicationTests in 21.937 seconds (JVM running for 24.927)
oncat('%',?,'%') order by starts asc limit ?
in] c.a.ebookshop.EbookshopApplicationTests  : savedBook = Page 1 of 1 containing com.aruoxi.ebookshop.domain.Book instances
in] c.a.ebookshop.EbookshopApplicationTests  : 小王子
ok] o.s.s.concurrent.ThreadPoolTaskExecutor  : Shutting down ExecutorService 'applicationTaskExecutor'
ok] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'
ok] com.alibaba.druid.pool.DruidDataSource   : {dataSource-1} closing ...
```

Controller Test

用户注册

- register Test



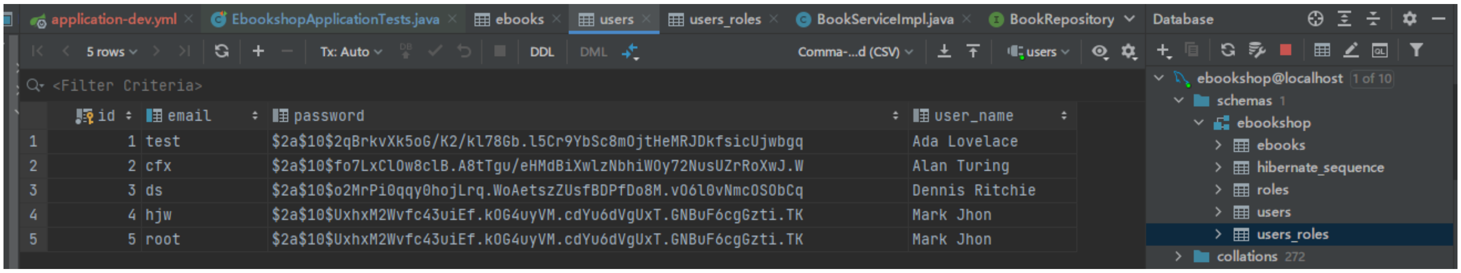
application-dev.yml
EbookshopApplicationTests.java
ebooks
users
users_roles
BookServiceImpl.java
BookRepository
Database
ebookshop@localhost 1 of 10
schemas 1
ebookshop
ebooks
hibernate_sequence
roles
users
users_roles
collations 272
users 5
Run: EbookshopApplication
EbookshopApplicationTests.restTemplateForRegistration
Test Results 696 ms
EbookshopApplicationTest 696 ms
restTemplateForRegistr 696 ms

```java
     */
    @Test
    void restTemplateForRegistration() {
      RestTemplate client = new RestTemplate();
      HttpHeaders headers = new HttpHeaders();
      MediaType type = MediaType.parseMediaType("application/json; charset=UTF-8");
      headers.setContentType(type);
      headers.add( "Accept", MediaType.APPLICATION_JSON.toString());

      JSONObject param = new JSONObject();
      param.put("username", "registerTest");
      param.put("email", "registerTest@qq.com");
      param.put("password", "123456");

      HttpEntity formEntity = new HttpEntity(param, headers);

      String result = client.postForObject( "http://localhost:8081/api/register", formEntity, String.class);
      LOG.info("result: " + result);
    }
```

Tests passed: 1 of 1 test – 696 ms

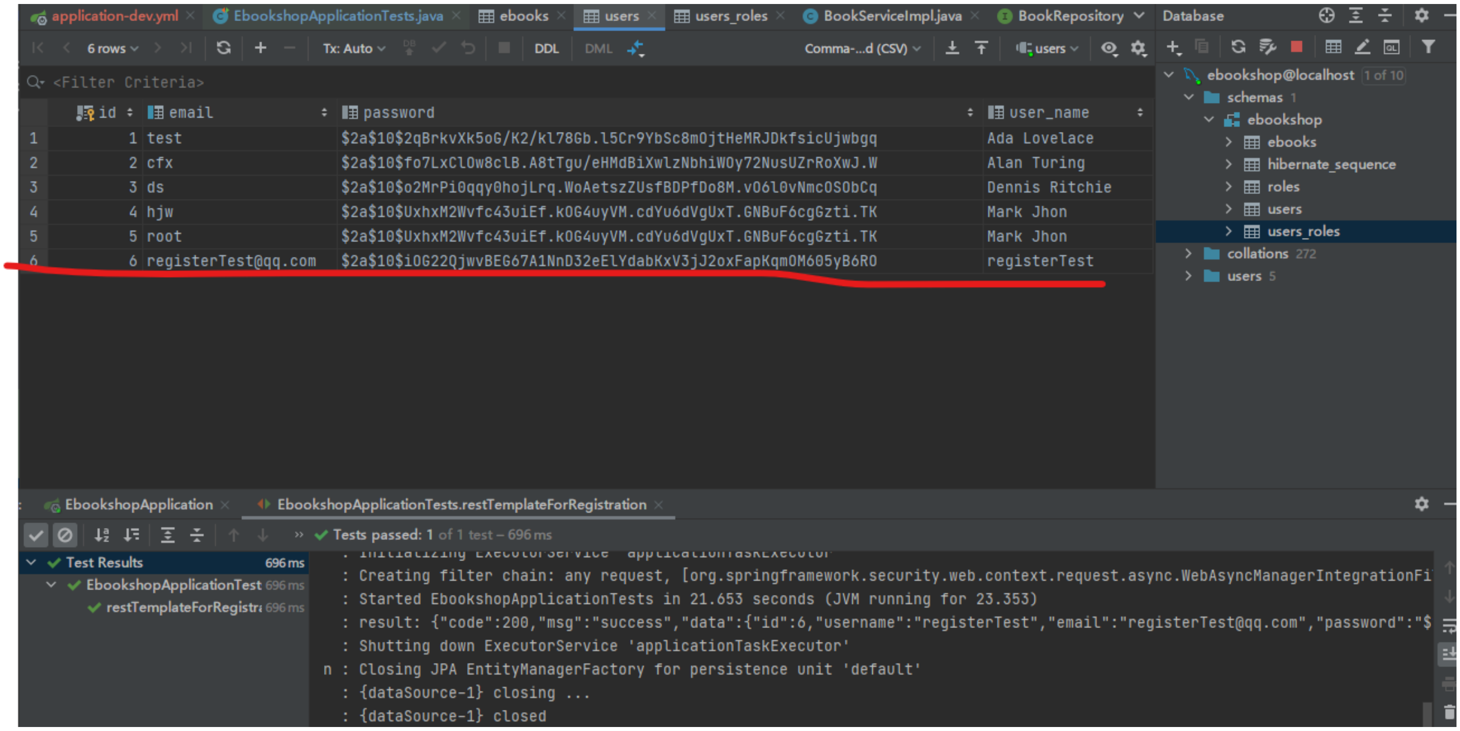
```
  : Initializing ExecutorService 'applicationTaskExecutor'
  : Creating filter chain: any request, [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFi
  : Started EbookshopApplicationTests in 21.653 seconds (JVM running for 23.353)
  : result: {"code":200,"msg":"success","data":{"id":6,"username":"registerTest","email":"registerTest@qq.com","password":"$
  : Shutting down ExecutorService 'applicationTaskExecutor'
n : Closing JPA EntityManagerFactory for persistence unit 'default'
  : {dataSource-1} closing ...
  : {dataSource-1} closed
```

- **注册前**的数据库用户列表



	id	email	password	user_name
1	1	test	$2a$10$2qBrkvXk5oG/K2/kl78Gb.l5Cr9YbSc8mOjtHeMRJDkfsicUjwbgq	Ada Lovelace
2	2	cfx	$2a$10$fo7LxClOw8clB.A8tTgu/eHMdBiXwlzNbhiWOy72NusUZrRoXwJ.W	Alan Turing
3	3	ds	$2a$10$o2MrPi0qqy0hojLrq.WoAetszZUsfBDPfDo8M.vO6l0vNmcOSObCq	Dennis Ritchie
4	4	hjw	$2a$10$UxhxM2Wvfc43uiEf.kOG4uyVM.cdYu6dVgUxT.GNBuF6cgGzti.TK	Mark Jhon
5	5	root	$2a$10$UxhxM2Wvfc43uiEf.kOG4uyVM.cdYu6dVgUxT.GNBuF6cgGzti.TK	Mark Jhon

- **注册后**的数据库用户列表



	id	email	password	user_name
1	1	test	$2a$10$2qBrkvXk5oG/K2/kl78Gb.l5Cr9YbSc8mOjtHeMRJDkfsicUjwbgq	Ada Lovelace
2	2	cfx	$2a$10$fo7LxClOw8clB.A8tTgu/eHMdBiXwlzNbhiWOy72NusUZrRoXwJ.W	Alan Turing
3	3	ds	$2a$10$o2MrPi0qqy0hojLrq.WoAetszZUsfBDPfDo8M.vO6l0vNmcOSObCq	Dennis Ritchie
4	4	hjw	$2a$10$UxhxM2Wvfc43uiEf.kOG4uyVM.cdYu6dVgUxT.GNBuF6cgGzti.TK	Mark Jhon
5	5	root	$2a$10$UxhxM2Wvfc43uiEf.kOG4uyVM.cdYu6dVgUxT.GNBuF6cgGzti.TK	Mark Jhon
6	6	registerTest@qq.com	$2a$10$iOG22QjwvBEG67A1NnD32eELYdabKxV3jJ2oxFapKqmOM605yB6RO	registerTest

```
  : Initializing ExecutorService 'applicationTaskExecutor'
  : Creating filter chain: any request, [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFi
  : Started EbookshopApplicationTests in 21.653 seconds (JVM running for 23.353)
  : result: {"code":200,"msg":"success","data":{"id":6,"username":"registerTest","email":"registerTest@qq.com","password":"$
  : Shutting down ExecutorService 'applicationTaskExecutor'
n : Closing JPA EntityManagerFactory for persistence unit 'default'
  : {dataSource-1} closing ...
  : {dataSource-1} closed
```