

Book Hub – Spring Boot Backend (JWT + CRUD)

A **secure REST API** built using Spring Boot for managing a collection of books, with **JWT-based authentication**, **Swagger API documentation**, and full **CRUD** functionality for authenticated users.

Section 1: Project Overview

Description:




Book Hub is a Spring Boot-based backend project designed to manage books securely. Authenticated users can:





- Register and log in
- Add, view, update, and delete books
- Access a real-time dashboard showing stats like top authors and recent books

Technology Stack:

- Java 17
 - Spring Boot 3
 - Spring Security + JWT
 - Spring Data JPA & Hibernate
 - MySQL
 - Springdoc OpenAPI (Swagger)
-



Section 2: API Features

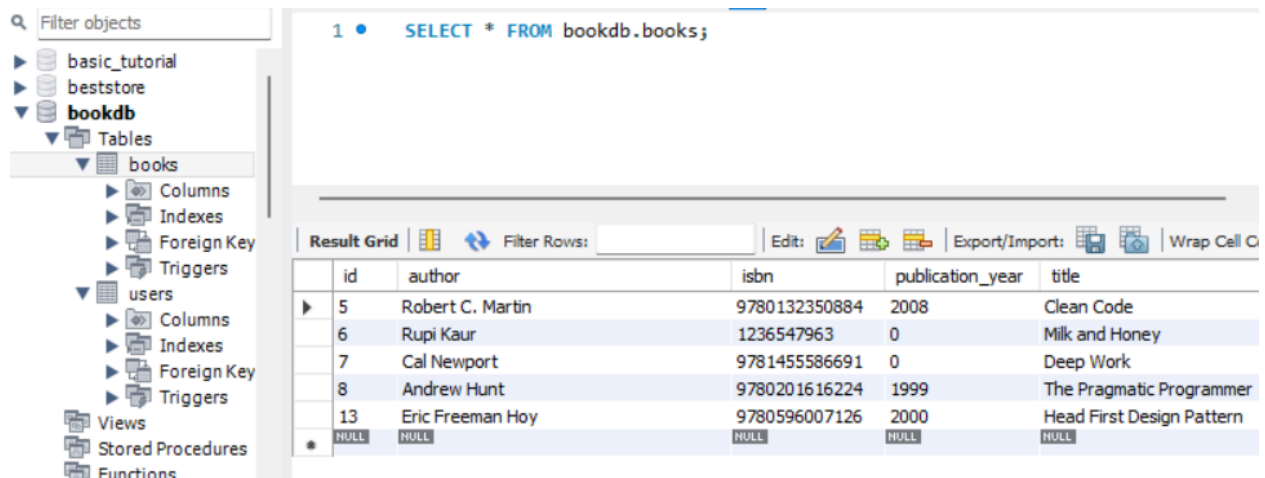
Feature	Description
 JWT Auth	Login generates token
 Add Book	<code>POST /api/books</code>
 Get Books	<code>GET /api/books</code>

 Edit Book	PUT <code>/api/books/{isbn}</code>
 Delete Book	DELETE <code>/api/books/{isbn}</code>
 Swagger Docs	<code>/swagger-ui/index.html</code>
 Dashboard	<code>/api/books/dashboard</code>

✓ Section 3: Screenshots (with Captions)

◆ 1. Database (MySQL Workbench)

-  Screenshot of book table
-  Screenshot of user table
- MySQL Database – Book and User Tables



	id	author	isbn	publication_year	title
▶	5	Robert C. Martin	9780132350884	2008	Clean Code
	6	Rupi Kaur	1236547963	0	Milk and Honey
	7	Cal Newport	9781455586691	0	Deep Work
	8	Andrew Hunt	9780201616224	1999	The Pragmatic Programmer
	13	Eric Freeman Hoy	9780596007126	2000	Head First Design Pattern
*	NULL	NULL	NULL	NULL	NULL

-  Screenshot of `/auth/register` with token respons

auth-controller

POST /auth/register

Parameters

No parameters

Request body required

application/json

```
{
  "username": "xyz",
  "password": "xyz123"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8081/auth/register' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "xyz",
    "password": "xyz123"
  }'
```

Request URL

```
http://localhost:8081/auth/register
```

Server response

Code

Details

200

Response body

Registration successfull

-  Screenshot of GET and POST PUT DELETE `/api/books`

DELETE

/api/books/{isbn}

GET

/api/books

Parameters

No parameters

Try it out

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header

Example Value | Schema

```
{  "id": 0,  "title": "string",  "author": "string",  "isbn": "0340442703",  "publicationYear": 0}
```

Available authorizations

bearer-key (http, Bearer)

Value:

eyJhbGciOiJIUzI1NiJ9.eyJzd

Authorize

Close

DELETE

/api/books/{isbn}

GET

/api/books

Parameters

No parameters

Try it out

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header

Example Value | Schema

```
{  "id": 0,  "title": "string",  "author": "string",  "isbn": "0325585850",  "publicationYear": 0}
```

Available authorizations

bearer-key (http, Bearer)

Authorized

Value: *****

Logout

Close

The screenshot displays a REST client interface with the following components:

- Method and URL:** A dropdown menu shows "GET" and the URL is "/api/books".
- Parameters:** A section labeled "Parameters" with a "Cancel" button. It indicates "No parameters".
- Buttons:** "Execute" and "Clear" buttons are present.
- Responses:** A section labeled "Responses" containing:
 - Curl:** A terminal window showing the curl command:

```
curl -X 'GET' \
  'http://localhost:8081/api/books' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJ3dCI6IjE4eXoKc1pYXQzOjE3NTIzMDk4MDMsImV4cCI6MTc1MjMONTg0M30.0nGN-ZlPekgx8lZEhk73RfWuy097V7e5nNzE08SA'
```
 - Request URL:** A field showing "http://localhost:8081/api/books".
 - Server response:** A section with a "Code" tab selected, showing a status of "200".
 - Response body:** A JSON array of book objects displayed in a dark-themed editor:

```
[
  {
    "id": 5,
    "title": "Clean Code",
    "author": "Robert C. Martin",
    "isbn": "9780132350884",
    "publicationYear": 2008
  },
  {
    "id": 6,
    "title": "Milk and Honey",
    "author": "Rupi Kaur",
    "isbn": "1236547863",
    "publicationYear": 0
  },
  {
    "id": 7,
    "title": "Deep Work",
    "author": "Cal Newport",
    "isbn": "9781455586691",
    "publicationYear": 0
  },
  {
    "id": 8
  }
]
```

POST

/api/books

Parameters

No parameters

Request body

required

application/json

```
{  
  "id": 14,  
  "title": "The Memoirs of Sherlock Holmes",  
  "author": "Sir Arthur Conan Doyle",  
  "isbn": "1316417620",  
  "publicationYear": 1955  
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \  
  'http://localhost:8081/api/books' \  
-H 'accept: */*' \  
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ6NDQwIm4iLCJ0eXAiOiJKV1QiLCJkaWkiOiJlbnRtZjB0LmN6Cj0tZIPakgs1ZEhk73RfnWuyos7V7cSnlzE8BSA' \  
-H 'Content-Type: application/json' \  
-d '{  
  "id": 14,  
  "title": "The Memoirs of Sherlock Holmes",  
  "author": "Sir Arthur Conan Doyle",  
  "isbn": "1316417620",  
  "publicationYear": 1955  
}'
```

Request URL

http://localhost:8081/api/books

Server response

Code

Details

200

Response body

```
{  
  "id": 14,  
  "title": "The Memoirs of Sherlock Holmes",  
  "author": "Sir Arthur Conan Doyle",  
  "isbn": "1316417620",  
  "publicationYear": 1955  
}
```

Download

GET

/api/books/{isbn}

Parameters

Cancel

Name	Description
isbn required	
string	1316417620
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8081/api/books/1316417620' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6ImlzbnQ1IiwiaWF0IjE5OTk1MjE0MjE0Lm51IiwiaXNjaW50IjpbImVudG97V7E5bWzE0BSA'
```

Request URL

http://localhost:8081/api/books/1316417620

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": 14, "title": "The Memoirs of Sherlock Holmes", "author": "Sir Arthur Conan Doyle", "isbn": "1316417620", "publicationYear": 1995 }</pre></div><div><div>Download</div></div></div>

PUT

/api/books/{isbn}

Parameters

NameDescription

isbn * requiredstring(path)1316417620

Request body requiredapplication/json

{
 "id": 14,
 "title": "The Memoirs of Sherlock Holmes",
 "author": "Arthur",
 "isbn": "1316417620",
 "publicationYear": 2025
}

Execute

Clear

Responses

Curlcurl -X 'PUT' \
 'http://localhost:8081/api/books/1316417620' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQ6ImV4cCI6IjEwMTZlOTYtMjAyMy0xMC0yMi0wNjc3ODk3MjA3LWVudC97V7e5nWzE0BSA' \
-d '{
 "id": 14,
 "title": "The Memoirs of Sherlock Holmes",
 "author": "Arthur",
 "isbn": "1316417620",
 "publicationYear": 2025
}'

Request URLhttp://localhost:8081/api/books/1316417620

Server responseCodeDetails200Response body{
 "id": 14,
 "title": "The Memoirs of Sherlock Holmes",
 "author": "Arthur",
 "isbn": "1316417620",
 "publicationYear": 2025
}

DELETE

/api/books/{isbn}

Parameters

Cancel

Name	Description
isbn <small>* required</small>	
string	9780201616224
(path)	

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:8081/api/books/9780201616224' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpYXBQI0JFE3NTIzMDk4NDMwImV4dCI6MTc1MjM0MTg0M30..n6cXN-ZlPekgx8lZEhK73RfmVuyo97V7e5nNzE0BSA'
```

Request URL

```
http://localhost:8081/api/books/9780201616224
```

Server response

CodeDetails

204

Undocumented

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
date: Sat,12 Jul 2025 09:34:26 GMT
expires: 0
keep-alives: timeout=60
pragma: no-cache
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
Content-Type: application/json
```

GET

/api/books/dashboard

Parameters

Cancel

No parameters

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8081/api/books/dashboard' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpYXBQI0JFE3NTIzMDk4NDMwImV4dCI6MTc1MjM0MTg0M30..n6cXN-ZlPekgx8lZEhK73RfmVuyo97V7e5nNzE0BSA'
```

Request URL

```
http://localhost:8081/api/books/dashboard
```

Server response

CodeDetails

200

Response body

```
{
  "totalBooks": 2,
  "topAuthors": [
    "Robert C. Martin (1)",
    "Cal Newport (1)"
  ],
  "recentBooks": [
    "Deep Work",
    "Clean Code"
  ]
}
```

Download

✔ Section 4: JWT Token Flow Diagram

User logs in

↓
Receives JWT token
↓
Token saved in localStorage (frontend)
↓
Frontend sends token in Authorization: Bearer <token>
↓
Secured endpoint validates token
→ If valid: Access granted
→ If invalid: 403 Forbidden

✓ Section 5: How to Run the Project

📁 Backend Setup

git clone <https://github.com/SanchitaDevkar/Backend-BookHub.git>
cd Backend-BookHub

Update database configuration in `application.properties`:

```
spring.datasource.url=jdbc:mysql://localhost:3306/bookhub  
spring.datasource.username=root  
spring.datasource.password=yourpassword
```


Run the project:

```
mvn spring-boot:run
```

Access Swagger UI:

<http://localhost:8081/swagger-ui/index.html>

📷 Screenshot 📷 Screenshot 1:




Create Your Account

Username

Password

Register

Already have an account? [Login here](#)



Welcome to Book Hub

Username

Password

Login

Don't have an account? [Register here](#)

Book List

[+ Add Book](#)

Clean Code by Robert C. Martin (ISBN: 9780132350884)	+ Edit Delete
Deep Work by Cal Newport (ISBN: 9781455586691)	+ Edit Delete

Add New Book

Title	<input type="text" value="The Alchemist"/>
Author	<input type="text" value="Paulo Coelho"/>
ISBN	<input type="text" value="123654789632"/>
Publication Year	<input type="text" value="2005"/>
<div>Add Book</div>	

← → ↻ 📄 localhost:4200/add-book 🔍 ⚙️ 📄 School ⋮

📖 Book Hub

👤 dhanaji@gmail.com 📖 Books 📊 Dashboard 🚪 Logout

📄 📖 📊 🚪

localhost:4200 says

✅ Book added successfully!

OK

Title

The Alchemist

Author

Paulo Coelho

ISBN

1236547896

Publication Year

2005

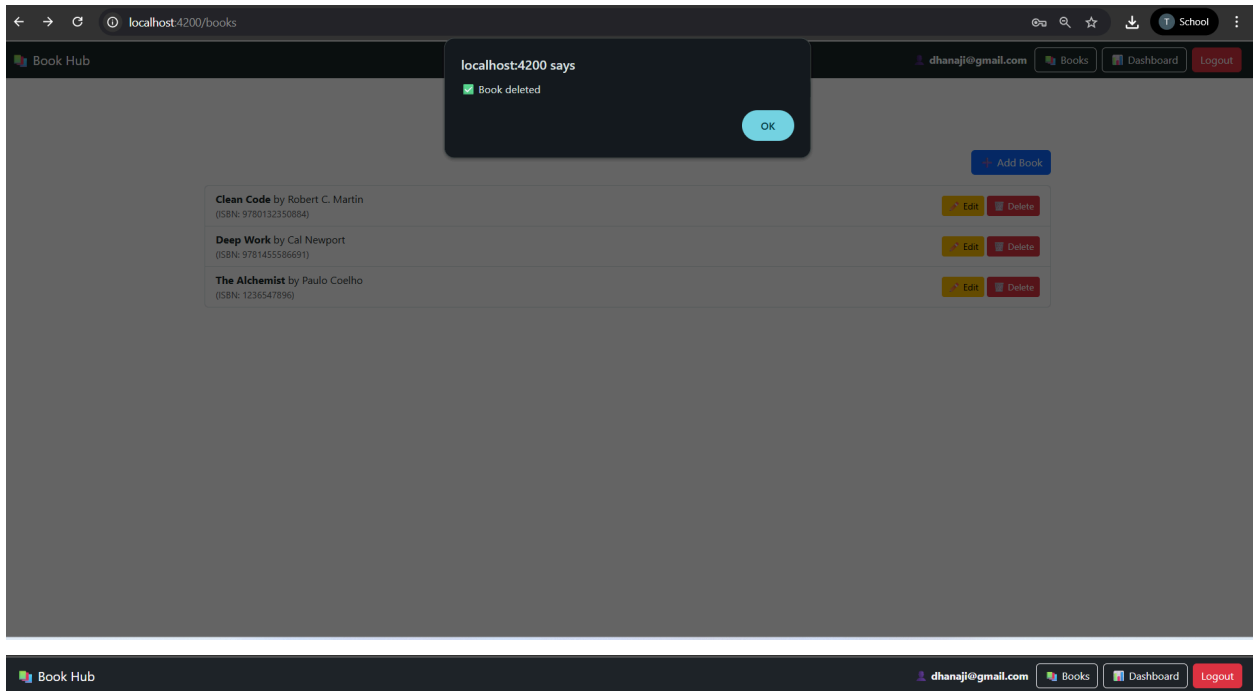
Add Book

📖 Book Hub

👤 dhanaji@gmail.com 📖 Books 📊 Dashboard 🚪 Logout

📖 Book List

		+ Add Book
Clean Code by Robert C. Martin (ISBN: 9780132350884)	🔍 Edit	🗑 Delete
Deep Work by Cal Newport (ISBN: 9781455586691)	🔍 Edit	🗑 Delete
The Alchemist by Paulo Coelho (ISBN: 1236547896)	🔍 Edit	🗑 Delete




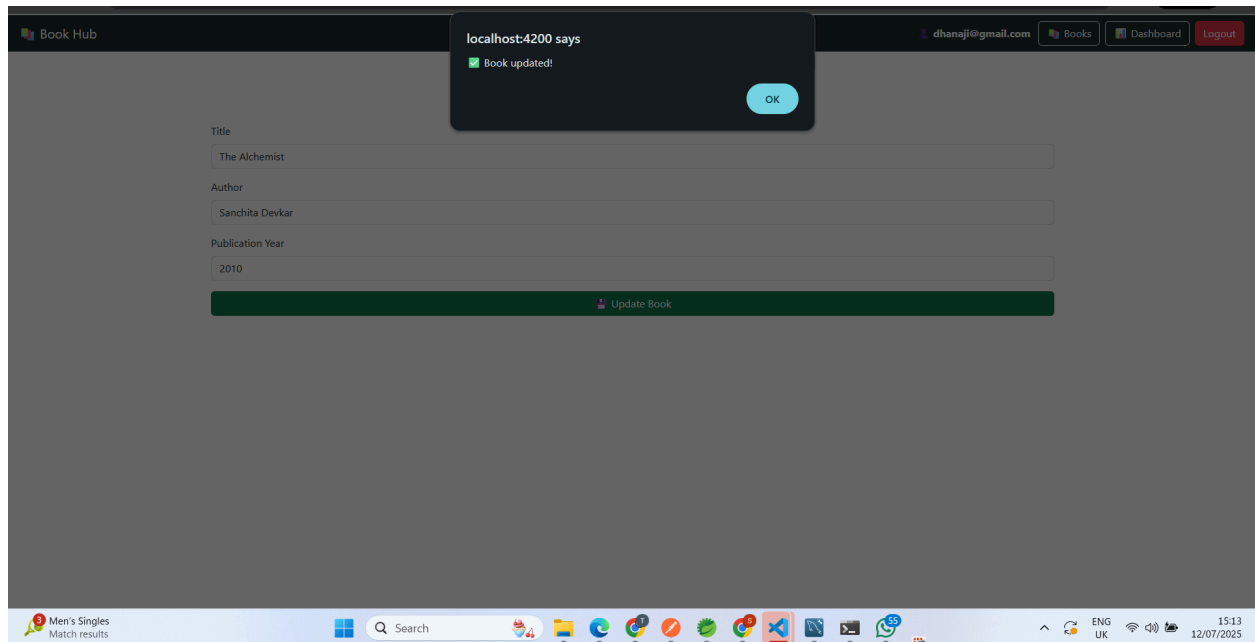
Edit Book

Title

Author

Publication Year

 Update Book



Additions Based on Full Stack Build

- 🔑 JWT Auth enabled login and registration (single user role)
- ✨ Dashboard endpoint `/api/books/dashboard` returns:
 - Total books
 - Top 3 authors
 - Recently added 5 books
- 📊 Shown beautifully in Angular frontend with Bootstrap cards

Dashboard

Total Books 2	Top Authors Robert C. Martin (1) Sanchita Devkar (1)	Recent Books The Alchemist Clean Code
-------------------------	-------------------------------------------------------------------	----------------------------------------------------

Author

Sanchita Devkar

Full Stack Java Developer | Angular + Spring Boot + JWT