RESTful Bookstore API

Project Overview

Project Title: RESTful Bookstore API

Objective:

To build a fully functional backend RESTful API for managing books and authors in a bookstore. This project demonstrates key skills needed in backend development, such as REST API design, CRUD operations, validation, exception handling, authentication, and database integration using Spring Boot and MySQL.

Technologies Used:

- Programming Language: Java 17
- Framework: Spring Boot 3.5.3
- Database: MySQL
- Build Tool: Maven
- API Testing: Postman
- API Documentation: Swagger UI
- Authentication: Spring Security (Basic Auth)
- Other: Lombok

Tools Used:

- Eclipse IDE 2025-03
- Spring Initializr
- MySQL Workbench
- Postman

Package Structure:

- `controller`: REST controllers for Book and Author
- `service`: Interfaces and implementations of business logic
- `repository`: Spring Data JPA interfaces
- `entity`: Book and Author models
- `dto`: Request/Response data transfer objects
- `exception`: Custom exceptions and global error handler

Features & Setup Guide

Key Features:

- CRUD operations for Book and Author
- Input validation using Jakarta Bean Validation
- Global exception handling with meaningful messages
- Clean and RESTful API design
- Basic Authentication using Spring Security
- Swagger UI integration for live API testing

Authentication Credentials: Username: `admin` , Password: `admin@123`

Example Endpoints:

- `GET /api/books` - Fetch all books
- `POST /api/books` - Add a new book
- `GET /api/books/{id}` - Get book by ID
- `PUT /api/books/{id}` - Update book
- `DELETE /api/books/{id}` - Delete book
- `GET /api/author` - Fetch all authors
- `POST /api/author` - Add a new author
- `GET /api/author/{id}` - Get author by ID
- `DELETE /api/author/{id}` - Delete author

How to Run:

1. Import the project into Eclipse via `File → Import → Existing Maven Project`
2. Configure MySQL DB:
   - Create DB `bookstore_db`
   - Update DB credentials in `application.properties`
3. Run the application from `BookstoreApplication.java`
4. Access APIs via Postman or Swagger UI

Swagger UI: http://localhost:8080/swagger-ui.html

Database Configuration (`application.properties`):

```
spring.datasource.url=jdbc:mysql://localhost:3306/bookstore_db
spring.datasource.username=root
spring.datasource.password=your_password
spring.jpa.hibernate.ddl-auto=update
```

Conclusion: This project serves as a resume-ready showcase of full-stack backend development using Spring Boot. It includes secure API design, proper exception management, modular code structure, and production-ready practices such as layered architecture and DTO-based data transfer.