Book-Author App Documentation

Github URL

https://github.com/ashar340codes/simple-web-app

Overview

Spring Boot 3.3 application that manages Authors and Books with CRUD (Create, Read, Update) plus a join view of book titles and author names. Uses Spring MVC, Spring Data JPA, JSP views, and an in-memory H2 database seeded with 10 authors and 10 books on startup.

Entity Design

- **Author** (`src/main/java/com/example/bookauthorapp/model/Author.java`): `id`, `name`, `biography`, one-to-many `books`.
- **Book** (`src/main/java/com/example/bookauthorapp/model/Book.java`): `id`, `title`, `genre`, many-to-one `author` via `author_id`.
- Relationship: one Author to many Books; Books reference their Author.

Implementation Layers

- **Repository**: `AuthorRepository`, `BookRepository` (includes `findBooksWithAuthors()` JPQL inner join).
- **Service**: `AuthorService`, `BookService` wrap repository access.
- **Controller**: `AuthorController`, `BookController` drive list/create/edit flows and bind data to JSPs.
- **Views**: JSPs under `src/main/webapp/WEB-INF/views` (`authors.jsp`, `books.jsp`, `author-form.jsp`, `book-form.jsp`) plus `css/styles.css`.
- **Bootstrap/Seed**: `BookAuthorAppApplication` seeds 10 authors and 10 books if DB is empty.

Key Endpoints

- `GET /authors` – list authors
- `GET /authors/new` – author form
- `POST /authors` or `POST /authors/new` – create author
- `GET /authors/edit/{id}` – edit author form
- `POST /authors/edit/{id}` – update author
- `GET /books` – list books and joined data
- `GET /books/new` – book form
- `POST /books` or `POST /books/new` – create book
- `GET /books/edit/{id}` – edit book form
- `POST /books/edit/{id}` – update book
- H2 console: `http://localhost:8080/h2-console` (JDBC URL `jdbc:h2:mem:testdb`, user `sa`, password `password`)

Build & Run

- Prereqs: JDK 17+ (code uses manual getters/setters), Maven 3.9+ available on PATH.
- Build/tests: `mvn clean test`
- Run (dev): `mvn spring-boot:run`
- Run (jar): `mvn clean package` then `java -jar target/book-author-app-0.0.1-SNAPSHOT.jar`
- Access app at `http://localhost:8080/authors` or `/books`.

Testing

- Repository tests validate CRUD and join query.
- Service tests use Mockito to verify service-level behavior.
- Run with `mvn test`.

CSS/UX

Basic styling at `src/main/webapp/css/styles.css` for tables, forms, and links. Reusable forms post to `/authors` or `/books` and also accept `/new//edit/{id}` POST routes to avoid 405 errors.

Challenges & Solutions

- **405 on edit/new routes**: Added explicit POST mappings for `/edit/{id}` and `/new` plus form actions to matching endpoints.
- **Lombok vs JDK 25**: Removed Lombok; added explicit getters/setters to keep compilation stable on newer JDKs.
- **Tooling availability**: Ensure Maven 3.9+ is installed and `mvn` is on PATH.

Screenshots

- `![Authors List](screenshots/author-create-form.png)`
- `![Books List and Join](screenshots/book-list-joined.png)`
- `![Book Create Form](screenshots/book-create-form.png)`
- `![Author Create Form](screenshots/author-create-form.png)`

File Map (high level)

- `src/main/java/com/example/bookauthorapp/` – app entry, controllers, services, repositories, models
- `src/main/resources/application.properties` – datasource and view config
- `src/main/webapp/WEB-INF/views/` – JSP views
- `src/main/webapp/css/styles.css` – styling
- `src/test/java/com/example/bookauthorapp/` – tests
- `.gitignore` – ignores build output and IDE/OS junk

How to Extend

- Add Delete flows (controller, service, view buttons).
- Swap H2 for MySQL/PostgreSQL by updating `application.properties` and dependencies.
- Add validation annotations and form error display for better UX.