Book Store API Project Report

Introduction

The Book Store API is a RESTful web service designed to manage books and their authors efficiently. It provides a seamless way to register, retrieve, update, and delete book records along with their associated author details. The API supports essential features like pagination, sorting, and validation to enhance user experience and data integrity.

Abstract

This project implements a backend system for a book store using Spring Boot. It allows users to perform CRUD (Create, Read, Update, Delete) operations on books and authors. The API ensures data validation using Spring Boot Starter Validation and offers interactive API documentation through Swagger. Pagination and sorting features help in managing large datasets effectively.

Tools Used

- **Spring Boot**: Framework for building the REST API.
- **Spring Data JPA**: For database operations and ORM.
- **Hibernate**: JPA implementation for managing entities.
- **MySQL**: Relational database to store book and author data.
- **Spring Boot Starter Validation**: To validate incoming data.
- **Swagger (OpenAPI)**: For API documentation and testing.
- **Lombok**: To reduce boilerplate code for getters, setters, and constructors.
- **Maven**: Build and dependency management tool.

Modules and Functionality

The project is divided into several key modules:

- **Entity Module**: Defines `Author` and `Book` entities with a one-to-many relationship and metadata fields for timestamps.
- **Model Module**: Contains data transfer objects with validation annotations to ensure data correctness.
- **Repository Module**: Interfaces extending `JpaRepository` for database access.
- **Service Module**: Implements business logic for registering, retrieving, updating, and deleting books and authors.

- **Controller Module**: REST controllers expose endpoints with support for pagination, sorting, and Swagger documentation.
- **Exception Handling Module**: Global exception handler manages validation errors and custom exceptions.

Key Features

- **Pagination and Sorting**: Enables users to request specific pages of book data sorted by any field.
- **Validation**: Ensures input data meets required criteria using annotations.
- **Bidirectional Relationship**: Maintains consistency between books and authors with cascading saves and deletes.
- **Swagger API Documentation**: Provides an interactive UI to explore and test API endpoints.
- **Exception Handling**: Returns meaningful HTTP responses for errors and validation failures.

Steps Involved in Building the Project

1. Setup Spring Boot project with required dependencies.
2. Design entity classes for `Author` and `Book`.
3. Create model classes with validation constraints.
4. Implement repository interfaces for database access.
5. Develop service layer for business logic.
6. Build REST controllers with Swagger annotations.
7. Add global exception handling.
8. Configure application properties for database and server.
9. Test API endpoints using Swagger UI and Postman.
10. Document the API and finalize the project.

Conclusion

The Book Store API project successfully demonstrates a full-stack backend application using Spring Boot. It efficiently manages book and author data with robust validation, pagination, sorting, and clear API documentation. This project can be extended further with features like user authentication, advanced search, and frontend integration.

— *End of Report* —