Practice Exercise: Building a Bookstore Management API

Objective

Develop a RESTful API using .NET Core and SQL Server to manage a bookstore database. This API will allow operations on books and authors, supporting CRUD operations for both entities.

Requirements

- Database Setup: Design and create tables for Books and Authors in SQL Server.
- Entity Framework: Use the Database First approach to scaffold models and DbContext.
- Repository Pattern: Implement repository classes to manage database operations.
- API Endpoints: Create controllers to handle HTTP methods for CRUD operations.

Detailed Steps

Database Design

Authors Table

```
CREATE TABLE Authors (
    AuthorId int IDENTITY(1,1) PRIMARY KEY,
    Name nvarchar(255) NOT NULL,
    Biography nvarchar(MAX)
);
```

Books Table

```
CREATE TABLE Books (
    BookId int IDENTITY(1,1) PRIMARY KEY,
    Title nvarchar(255) NOT NULL,
    YearPublished int,
    Genre nvarchar(100),
    AuthorId int FOREIGN KEY REFERENCES Authors(AuthorId)
);
```

Populate the tables with sample data.

Set Up Entity Framework

Use the `Scaffold-DbContext` command to generate models and DbContext from the existing database schema. Configure the connection string in `appsettings.json`.

Create Repository Classes

Implement `AuthorRepository` and `BookRepository` with methods for each CRUD operation. Use dependency injection to manage repository instances.

Building the API Controllers

Create `AuthorsController` and `BooksController` with methods for GET, POST, PUT, and DELETE operations. Ensure each method utilizes the appropriate repository methods to interact with the database.

Testing and Validation

Test the API using Swagger UI to ensure all endpoints function correctly. Validate input data and handle errors gracefully to ensure the API's robustness.