**Exercise 1: Online Bookstore - Setting Up RESTful Services**

Business Scenario:

You are tasked with developing a RESTful service for an online bookstore. The service will manage books, authors, and customers.

1. **Setup Spring Boot Project:**

* Using Spring Initializr (start.spring.io), I initialized a new Spring BookStoreAPI and added the following dependencies:
* Spring Web: For creating RESTful web services.
* Spring Boot DevTools: For automatic restarts and other development conveniences.
* Lombok: To reduce boilerplate code by using annotations like @Data, @Getter, @Setter, etc.

 Go to Spring Initializr.

Set the project name as BookstoreAPI.

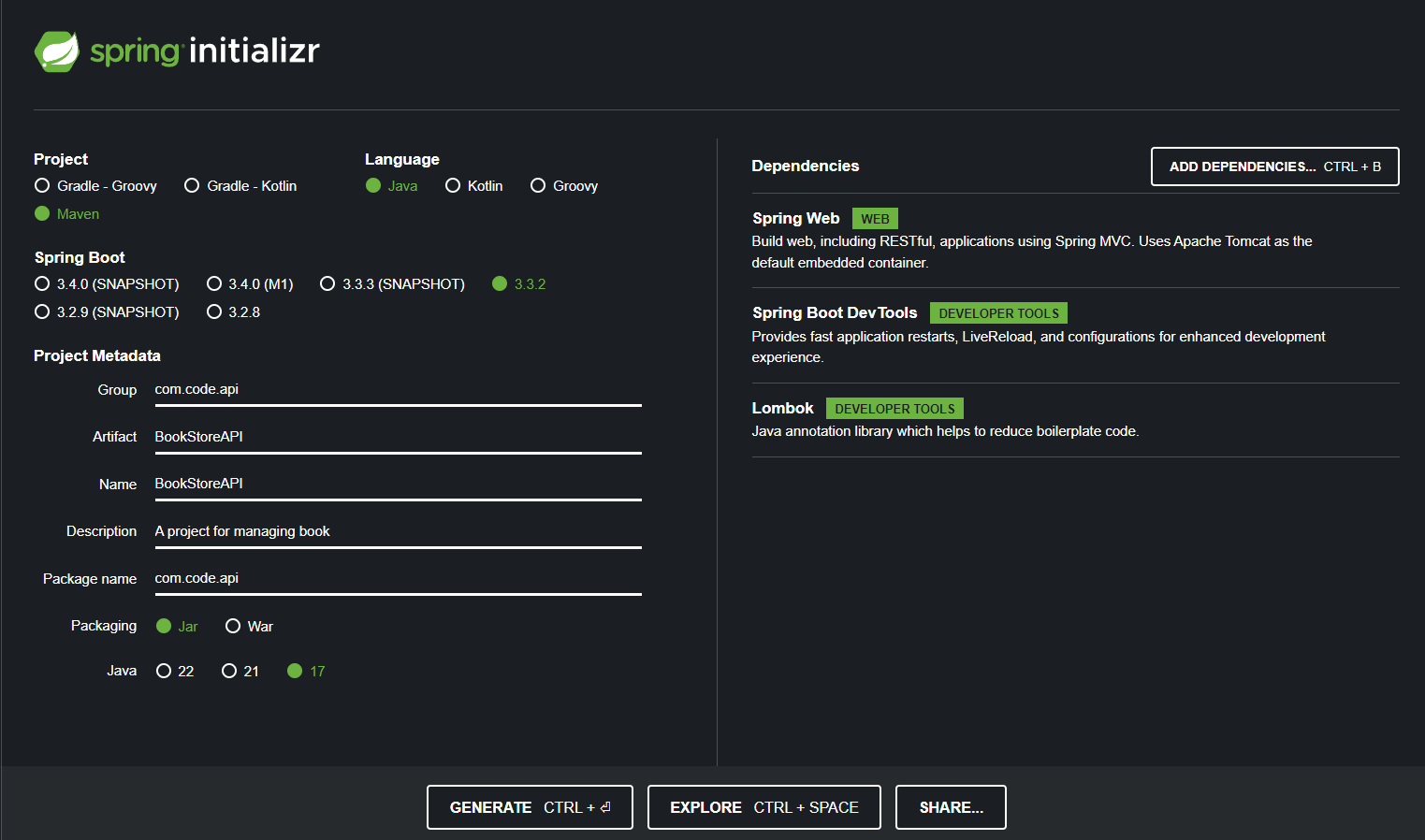
Choose the following options:

* + **Project:** Maven Project
  + **Language:** Java
  + **Spring Boot Version:** 3.3.2
  + **Packaging:** Jar
  + **Java Version:** 17

 **Add Dependencies:**

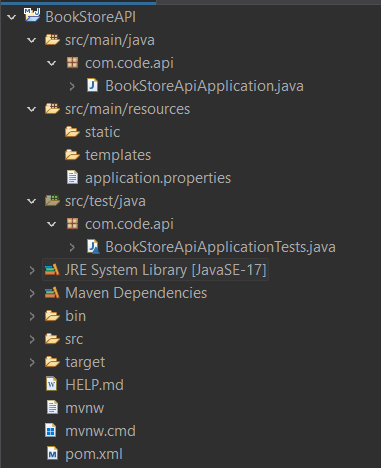
 **Generate the Project:**

* Click on "Generate" to download the project.
* Unzipped the downloaded project and opened it in Eclipse IDE.



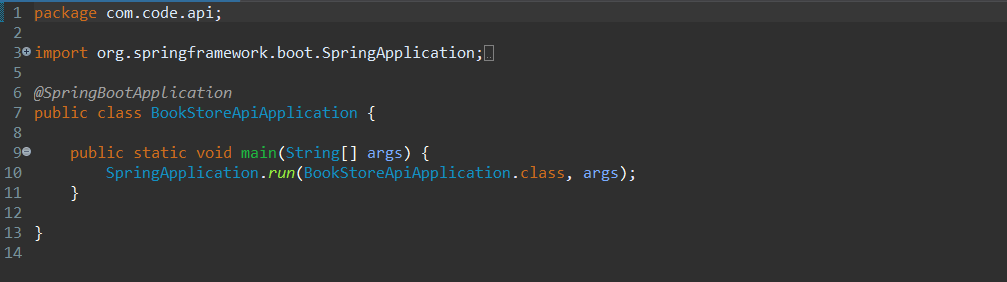
**2. Project Structure:**

* The generated project contains the following structure:



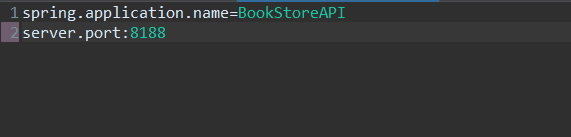
 **src/main/java:**

* Contains the main application class (BookstoreApiApplication.java) under the package (com.code.api).



 **src/main/resources:**

* application.properties for configuration.



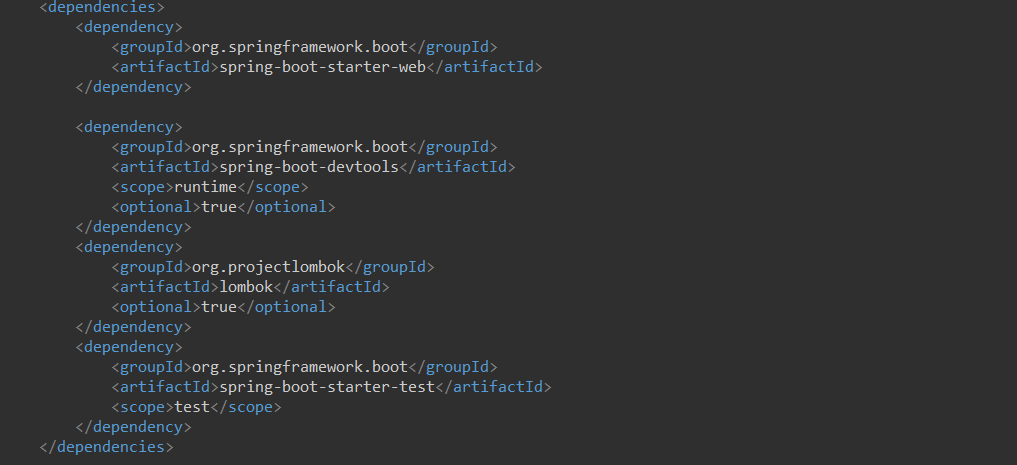
* static/: For static resources like HTML, CSS, JavaScript.
* templates/: For Thymeleaf templates if using a view layer.

 **src/test/java:**

* For writing unit and integration tests.

 **pom.xml:**

* Contains the heart of the project.
* The following dependencies that were added during the generation of the project can also be viewed in pom.xml file.



* 1. **What’s New in Spring Boot 3:**

Spring Boot 3 introduces several new features and improvements:

* **Java 17 Support:** Full support for Java 17, the latest LTS version, including new language features.
* **Spring Framework 6:** Built on top of Spring Framework 6, which brings Jakarta EE 9 and 10 support (transitioning from javax.\* to jakarta.\* packages).
* **Native Image Support:** Improved support for building native images using GraalVM, making applications more lightweight and faster to start.
* **Enhanced Observability:** New features for better observability, including improved metrics, tracing, and logging.
* **Updated Dependency Versions:** Dependencies like Spring Data, Spring Security, and Spring Batch have been updated to their latest versions, offering new features and improvements.
* **Configuration Improvements:** Enhanced configuration properties binding and validation, along with better handling of configuration properties in the application environment.
* **Performance Optimizations:** Various performance enhancements have been introduced, reducing startup time and memory consumption.

**Output:**

