# **Exercise 7: Implementing Constructor and Setter Injection**

## **Scenario**

The Library Management application now needs both constructor and setter injection for better control over bean initialization and dependency management.

## **Step 1: Configure Constructor Injection**

## **Author.java**

**Path: src/main/java/com/library/Author.java**

package com.library;  
  
public class Author {  
 private String name;  
  
 public Author(String name) {  
 this.name = name;  
 }  
  
 public String getAuthorName() {  
 return name;  
 }  
}

This class represents an Author dependency that will be injected using a constructor.



## 

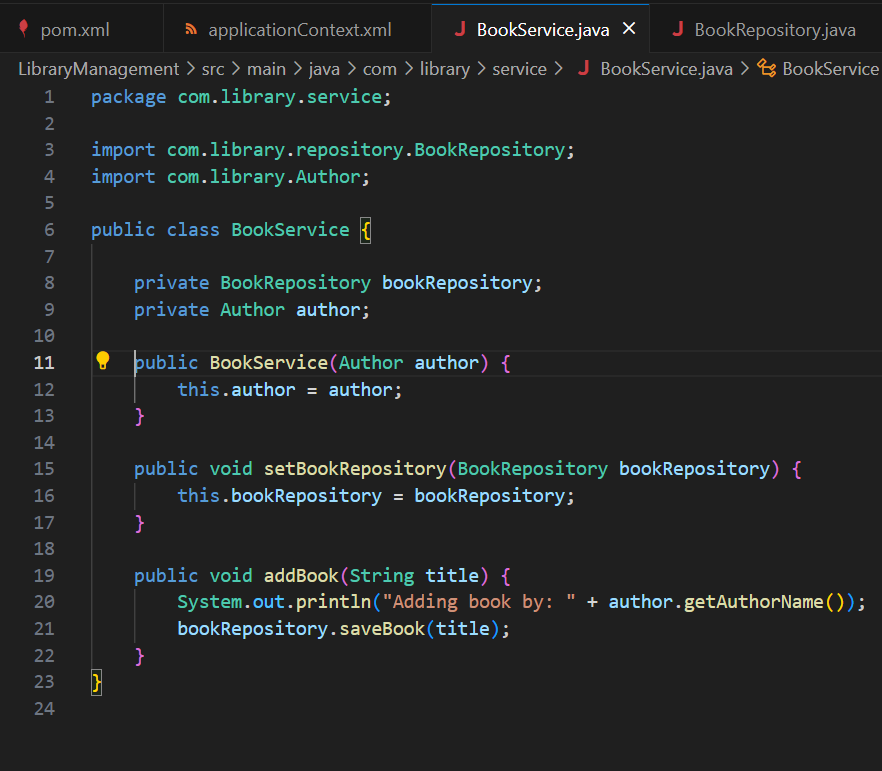
## **Step 2: Configure Setter Injection in BookService**

### **BookService.java**

**Path: src/main/java/com/library/service/BookService.java**

package com.library.service;  
  
import com.library.repository.BookRepository;  
import com.library.Author;  
  
public class BookService {  
  
 private BookRepository bookRepository;  
 private Author author;  
  
 // Constructor for Author  
 public BookService(Author author) {  
 this.author = author;  
 }  
  
 // Setter for BookRepository  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void addBook(String title) {  
 System.out.println("Adding book by: " + author.getAuthorName());  
 bookRepository.saveBook(title);  
 }  
}

This combines both constructor injection (for Author) and setter injection (for BookRepository).



### 

### **BookRepository.java**

**Path: src/main/java/com/library/repository/BookRepository.java**

package com.library.repository;  
  
public class BookRepository {  
  
 public void saveBook(String title) {  
 System.out.println("Book saved: " + title);  
 }  
}

This repository class saves books.

## **Step 3: applicationContext.xml**

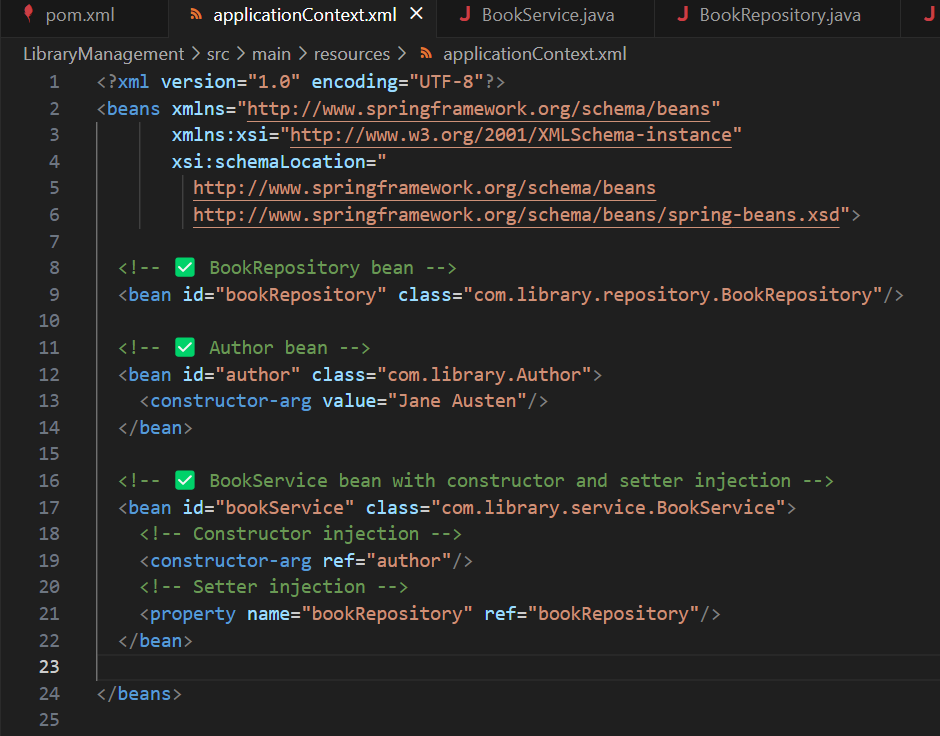
### **applicationContext.xml**

**Path: src/main/resources/applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <!-- BookRepository bean -->  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
  
 <!-- Author bean -->  
 <bean id="author" class="com.library.Author">  
 <constructor-arg value="Jane Austen"/>  
 </bean>  
  
 <!-- BookService bean with constructor and setter injection -->  
 <bean id="bookService" class="com.library.service.BookService">  
 <!-- Constructor injection -->  
 <constructor-arg ref="author"/>  
 <!-- Setter injection -->  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
  
</beans>

This XML configures:

* Author via <constructor-arg>
* BookRepository via <property>



## 

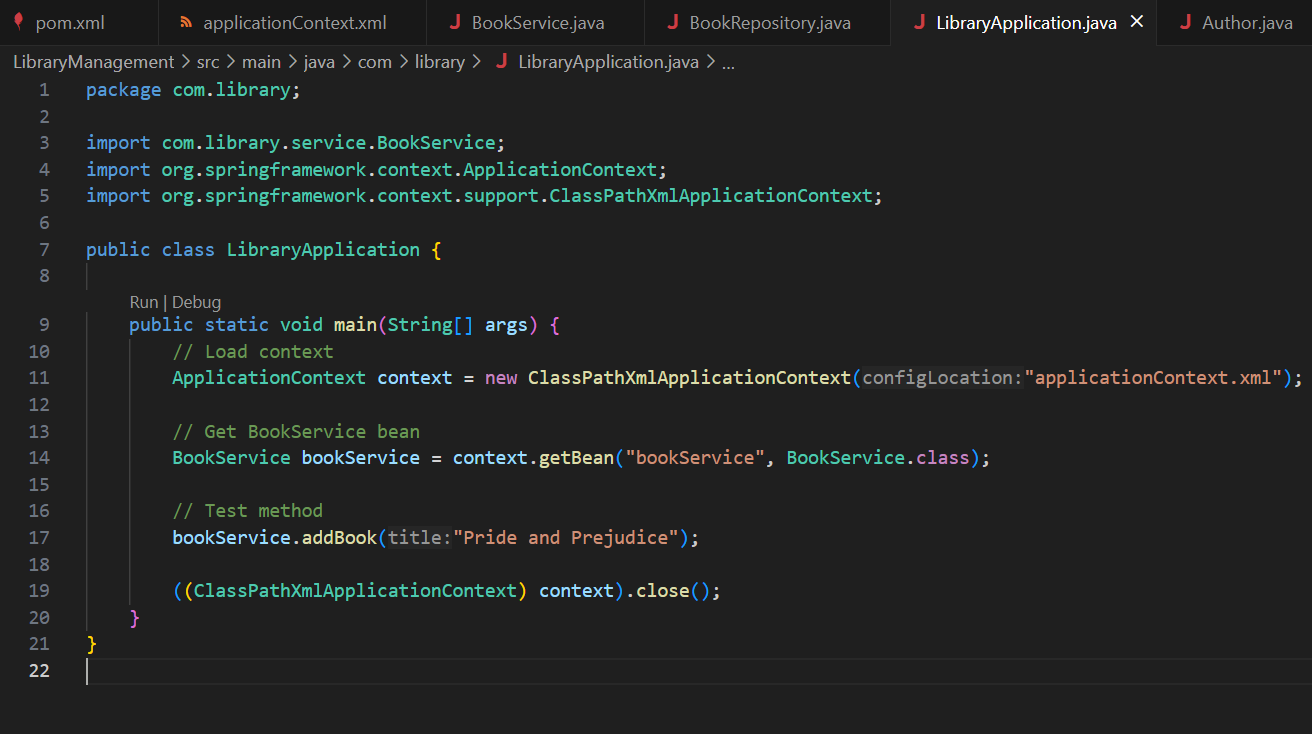
## **Step 4: Run the Application**

### **LibraryApplication.java**

**Path: src/main/java/com/library/LibraryApplication.java**

package com.library;  
  
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class LibraryApplication {  
  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = context.getBean("bookService", BookService.class);  
  
 bookService.addBook("Pride and Prejudice");  
  
 ((ClassPathXmlApplicationContext) context).close();  
 }  
}

This main class loads the Spring context and verifies both injection types work.



## How to Run

In your project root:

mvn compile  
mvn exec:java

Expected Output:

Adding book by: Jane Austen  
Book saved: Pride and Prejudice

