**Exercise 1: Configuring a Basic Spring Application**

**Scenario:**

Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

**BookRepository.java**

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook(String bookName) {

System.***out***.println("BookRepository: Saving book - " + bookName);

}

}

**BookService.java**

**package** com.library.service;

**import** com.library.repository.BookRepository;

**public** **class** BookService {

**private** BookRepository bookRepository;

// Setter for dependency injection

**public** **void** setBookRepository(BookRepository bookRepository) {

**this**.bookRepository = bookRepository;

}

**public** **void** addBook(String bookName) {

System.***out***.println("BookService: Adding book - " + bookName);

bookRepository.saveBook(bookName);

}

}

**MainApp.java**

**package** com.library;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.library.service.BookService;

**public** **class** MainApp {

**public** **static** **void** main(String[] args) {

// Load Spring container from applicationContext.xml

ApplicationContext context = **new** ClassPathXmlApplicationContext("applicationContext.xml");

// Get BookService bean

BookService bookService = context.getBean("bookService", BookService.**class**);

// Call method

bookService.addBook("Spring in Action");

}

}



