# Spring Core Maven Exercises Documentation

## Exercise 1: Configuring a Basic Spring Application

**- Project Name: LibraryManagement**

**Add Spring Core Dependency in pom.xml:**

<dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
</dependency>

**Create XML Configuration File:**

File: applicationContext.xml  
Location: src/main/resources

<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">  
  
 <bean id="bookRepository" class="com.library.repository.BookRepository"/>  
 <bean id="bookService" class="com.library.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

**Create Java Classes:**

* **BookRepository.java**

package com.library.repository;  
  
public class BookRepository {  
 public void getBookDetails() {  
 System.out.println("Fetching book details from repository...");  
 }  
}

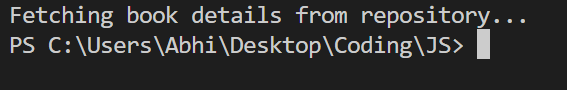
* **BookService.java**

package com.library.service;  
  
import com.library.repository.BookRepository;  
  
public class BookService {  
 private BookRepository bookRepository;  
  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void displayBooks() {  
 bookRepository.getBookDetails();  
 }  
}

* **LibraryManagementApplication.java**

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

**output:**



## Exercise 2: Implementing Dependency Injection

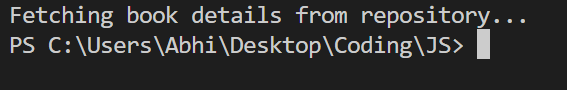
**Update applicationContext.xml:**  
<property name="bookRepository" ref="bookRepository"/>

**Ensure Setter Injection already included in BookService.java**

public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
}

**0utput:**

Run Application using LibraryManagementApplication.java



## Exercise 4: Creating and Configuring a Maven Project

**Project:  
 - Name: LibraryManagement**

**Add Dependencies in pom.xml:**

<dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
</dependencies>

**Configure Maven Compiler Plugin:**

<build>  
 <plugins>  
 <plugin>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.1</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
</build>

**Output:**

