

SPRING CORE AND MAVEN

- Exercise 1: Configuring a Basic Spring Application

→ solution:

Dependencies in pom.xml

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.22</version>
</dependency>
```

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>5.3.22</version>
</dependency>
```

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">
    <bean id="bookRepository"
        class="com.shom.library.repository.BookRepository" />

    <bean id="bookService" class="com.shom.library.service.BookService"/>
</beans>
```

BookRepository.java

```
package com.shom.library.repository;

public class BookRepository {
    public void getBooks() {
        System.out.println("Getting books--");
    }
}
```

BookService.java

```
package com.shom.library.service;

public class BookService {
    public void displayBooks() {
        System.out.println("Showing books from book repository---");
    }
}
```

Main.java

```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.shom.library.repository.BookRepository;
import com.shom.library.service.BookService;

public class Main {
    public static void main(String[] args) {

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

        BookRepository bookRepository = (BookRepository)
        context.getBean("bookRepository");
        bookRepository.getBooks();

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.displayBooks();
    }
}
```

```
<terminated> Main [Java Application] C:\eclipse\eclipse\plugins\org.eclipse.ju
Getting books--
Showing books from book repository---
```

Exercise 2: Implementing Dependency Injection

➔solution:

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">
  <bean id="bookRepository"
        class="com.shom.library.repository.BookRepository" />

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

BookRepository.java

```
package com.shom.library.repository;

public class BookRepository {
  public void getBooks() {
    System.out.println("Getting books--");
  }
}
```

BookService.java

```
package com.shom.library.service;

import com.shom.library.repository.BookRepository;

public class BookService {

  private BookRepository bookRepository;

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

```

    public void displayBooks() {
        bookRepository.getBooks();
        System.out.println("Showing books from book repository---");
    }
}

```

Main.java

```

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

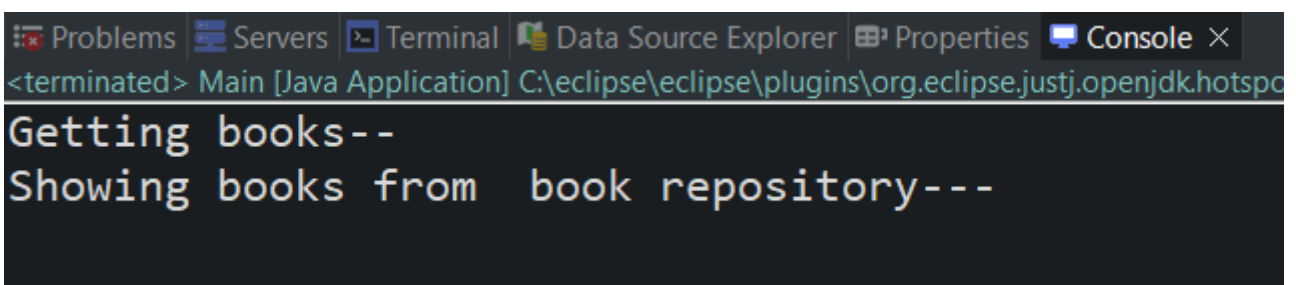
import com.shom.library.service.BookService;

public class Main {
    public static void main(String[] args) {

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

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.displayBooks();
    }
}

```



```

<terminated> Main [Java Application] C:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspc
Getting books--
Showing books from book repository---

```

Exercise 4: Creating and Configuring a Maven Project

➔ solution

New Maven project

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

☒ run archetype generation interactively

Properties available from archetype:

Name	Value
javaCompilerVer...	17
junitVersion	5.11.0

Add... Remove

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.shom</groupId>
  <artifactId>LibraryManagement1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>LibraryManagement1</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
```

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.22</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.22</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.22</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.22</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>