

WEEK 3 - Spring Core and Maven

Exercise 1: Configuring a Basic Spring Application

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.library</groupId>
    <artifactId>LibraryManagement</artifactId>
    <version>1.0-SNAPSHOT</version>

    <dependencies>

        <!-- Spring Core -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>5.3.30</version>
        </dependency>

        <!-- Spring Context -->
        <dependency>
            <groupId>org.springframework</groupId>
```

```
        <artifactId>spring-context</artifactId>
        <version>5.3.30</version>
    </dependency>
</dependencies>
</project>
```

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">

    <!-- Define BookRepository bean -->
    <bean id="bookRepository" class="com.library.repository.BookRepository"/>

    <!-- Define BookService bean and inject BookRepository -->
    <bean id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository"/>
    </bean>
</beans>
```

BookRepository.java

```
package com.library.repository;

import java.util.Arrays;
import java.util.List;
```

```
public class BookRepository {

    public List<String> findAllBooks() {
        return Arrays.asList("The Hobbit", "1984", "Clean Code", "Spring in Action");
    }
}
```

BookService.java

```
package com.library.service;

import com.library.repository.BookRepository;
import java.util.List;

public class BookService {

    private BookRepository bookRepository;

    // Setter injection

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

    public void displayBooks() {
        List<String> books = bookRepository.findAllBooks();
        System.out.println("Books available in the library:");
        for (String book : books) {
            System.out.println("- " + book);
        }
    }
}
```

```
}
```

LibraryApp.java

```
package com.library.main;
```

```
import com.library.service.BookService;
```

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

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

```
public class LibraryApp {
```

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

```
        // Load Spring Context
```

```
        ApplicationContext context = new
```

```
ClassPathXmlApplicationContext("applicationContext.xml");
```

```
        // Get BookService bean
```

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

```
        // Call service method
```

```
        bookService.displayBooks();
```

```
    }
```

```
}
```

OUTPUT:

Books available in the library:

- The Hobbit
- 1984
- Clean Code
- Spring in Action

Exercise 2: Implementing Dependency Injection

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"/>

    <!-- BookService bean with dependency injected -->

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

</beans>
```

BookService.java

```
package com.library.service;

import com.library.repository.BookRepository;
import java.util.List;

public class BookService {

    private BookRepository bookRepository;

    // Setter injection

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }


    public void displayBooks() {

        List<String> books = bookRepository.findAllBooks();

        System.out.println("Books available in the library:");

        for (String book : books) {

            System.out.println("- " + book);

        }

    }

}
```

LibraryManagementApplication.java

```
package com.library.main;

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) {

        // Load Spring application context

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

        // Get BookService bean

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

        // Call method to verify DI

        bookService.displayBooks();

    }

}
```

Output:

Books available in the library:

- The Hobbit
- 1984
- Clean Code
- Spring in Action

Exercise 4: Creating and Configuring a Maven Project

```
mvn archetype:generate -DgroupId=com.library \
    -DartifactId=LibraryManagement \
    -DarchetypeArtifactId=maven-archetype-quickstart \
    -DinteractiveMode=false
```

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.library</groupId>
    <artifactId>LibraryManagement</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>

    <properties>
        <java.version>1.8</java.version>
        <spring.version>5.3.30</spring.version>
    </properties>
```


<dependencies>

<!-- Spring Context (Core IoC container) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>\${spring.version}</version>

</dependency>

<!-- Spring AOP (for aspect-oriented programming) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>\${spring.version}</version>

</dependency>

<!-- Spring Web MVC (for building web applications) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>\${spring.version}</version>

</dependency>

<!-- Servlet API (required for Web MVC) -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

```
        <scope>provided</scope>
    </dependency>

    <!-- JUnit (for unit testing) -->
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.13.2</version>
        <scope>test</scope>
    </dependency>
</dependencies>

<!-- Build configuration -->
<build>
    <plugins>
        <!-- Maven Compiler Plugin -->
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
            <configuration>
                <source>${java.version}</source>
                <target>${java.version}</target>
            </configuration>
        </plugin>
    </plugins>
</build>
```

</project>

mvn clean install