

WEEK 3

Spring Core and Maven, Spring Data JPA with Spring Boot, Hibernate

Spring Core and Maven

1. Configuring a Basic Spring Application:

Create a Maven Project LMS

Add Spring Dependency in Pom.xml:

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

Create 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.cognizant.lms.repo.BookRepository"
/>

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

Create BookRepository.java:

```
package com.cognizant.lms.repo;

public class BookRepository {
    public void saveBook(String bookName) {
        System.out.println("Book '" + bookName + "' saved to the repository.");
    }
}
```

Create BookService.java:

```
package com.cognizant.lms.service;

import com.cognizant.lms.repo.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("Adding book: " + bookName);
        bookRepository.saveBook(bookName);
    }
}
```

Create App.java:

```
package com.cognizant.lms;

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

import com.cognizant.lms.service.BookService;

public class App {
    public static void main(String[] args) {
        ApplicationContext context = new
        ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.addBook("Spring in Action");
    }
}
```

Output:

```
1
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 import com.cognizant.lms.service.BookService;
7
8 public class App {
9     public static void main(String[] args) {
10         ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");
11
12         BookService bookService = (BookService) context.getBean("bookService");
13         bookService.addBook("Spring in Action");
14     }
15 }
16
```

Problems | Javadoc | Declaration | Console x

<terminated> App [Java Application] D:\FSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.jst.j2ee.openjdk.hotspot.jre.full.win32.x86_64

Adding book: Spring in Action
Book 'Spring in Action' saved to the repository.

2. Implementing Dependency Injection:

Create a Maven Project LMS

Add Spring Dependency in Pom.xml:

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

Create 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.cognizant.lms.repo.BookRepository"
    />

    <!-- BookService Bean with Dependency Injection -->
```

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

</beans>
```

Create BookRepository.java:

```
package com.cognizant.lms.repo;

public class BookRepository {
    public void saveBook(String bookName) {
        System.out.println("Book " + bookName + " saved to the repository.");
    }
}
```

Create BookService.java:

```
package com.cognizant.lms.service;

import com.cognizant.lms.repo.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);
    }
}
```

Create App.java:

```
package com.cognizant.lms;

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

import com.cognizant.lms.service.BookService;
```

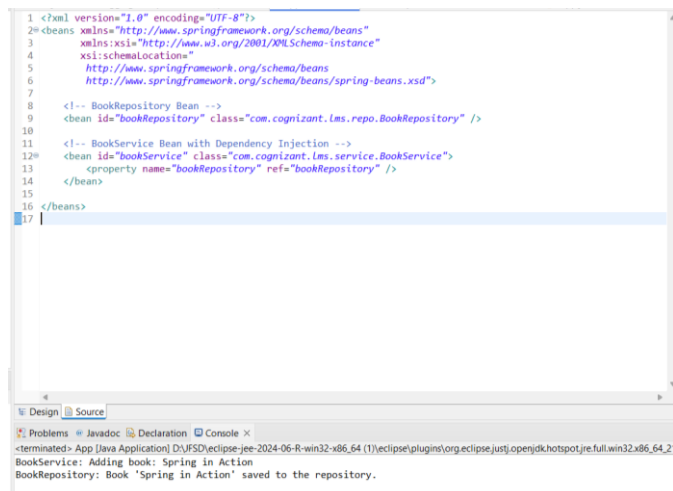
```

public class App {
    public static void main(String[] args) {
        ApplicationContext context = new
        ClassPathXmlApplicationContext("applicationContext.xml");

        BookService bookService = (BookService) context.getBean("bookService");
        bookService.addBook("Spring in Action");
    }
}

```

Output:



The screenshot shows an IDE window with XML code and a console output. The XML code defines two beans: a BookRepository and a BookService. The BookService bean has a dependency on the BookRepository bean. The console output shows the application running successfully and adding a book to the repository.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="
5         http://www.springframework.org/schema/beans
6         http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8     <!-- BookRepository Bean -->
9     <bean id="bookRepository" class="com.cognizant.lms.repo.BookRepository" />
10
11     <!-- BookService Bean with Dependency Injection -->
12     <bean id="bookService" class="com.cognizant.lms.service.BookService">
13         <property name="bookRepository" ref="bookRepository" />
14     </bean>
15
16 </beans>
17

```

Problems Javadoc Declaration Console X
 <terminated> App [Java Application] [D:\JSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.jee\openjdk-hotspot-jre\full\win32-x86_64_21
 BookService: Adding book: Spring in Action
 BookRepository: Book 'Spring in Action' saved to the repository.

3. Creating and Configuring a Maven Project

Create a maven project Library Management.

Add Spring Dependencies in Pom.xml :

```

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.32</version>
</dependency>

<!-- Spring AOP -->
<dependency>
    <groupId>org.springframework</groupId>

```

```

    <artifactId>spring-aop</artifactId>
    <version>5.3.32</version>
</dependency>

<!-- Spring Web MVC -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.32</version>
</dependency>

<!-- Servlet API (provided by container like Tomcat) -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>

```

Configure Maven Compiler Plugin:

```

<plugin>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.10.1</version>
    <configuration>
        <source>1.8</source>
        <target>1.8</target>
    </configuration>
</plugin>

```

Output:



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4    <modelVersion>4.0.0</modelVersion>
5
6    <groupId>com.cognizant</groupId>
7    <artifactId>LibraryManagement</artifactId>
8    <version>0.0.1-SNAPSHOT</version>
9
10   <name>LibraryManagement</name>
11   <!-- TODO: change it to the project's website -->
12   <url>http://www.example.com/</url>
13
14   <properties>
15     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16     <maven.compiler.release>17</maven.compiler.release>
17   </properties>
18
19   <dependencyManagement>
20     <dependencies>
21       <dependency>
22         <groupId>org.junit</groupId>
23         <artifactId>junit-bom</artifactId>
24         <version>5.11.0</version>
25         <type>pom</type>
26         <scope>import</scope>
27       </dependency>
28     </dependencies>
29   </dependencyManagement>
30
31   <dependencies>
32     <dependency>
33       <groupId>org.springframework</groupId>
34       <artifactId>spring-context</artifactId>
35       <version>5.3.32</version>
36     </dependency>

```

```
39# <dependency>
40#   <groupId>org.springframework</groupId>
41#   <artifactId>spring-aop</artifactId>
42#   <version>5.3.32</version>
43# </dependency>
44#
45# <!-- Spring Web MVC -->
46# <dependency>
47#   <groupId>org.springframework</groupId>
48#   <artifactId>spring-webmvc</artifactId>
49#   <version>5.3.32</version>
50# </dependency>
51#
52# <!-- Servlet API (provided by container like Tomcat) -->
53# <dependency>
54#   <groupId>javax.servlet</groupId>
55#   <artifactId>javax.servlet-api</artifactId>
56#   <version>4.0.1</version>
57#   <scope>provided</scope>
58# </dependency>
59# <dependency>
60#   <groupId>org.junit.jupiter</groupId>
61#   <artifactId>junit-jupiter-api</artifactId>
62#   <scope>test</scope>
63# </dependency>
64# <!-- Optionally: parameterized tests support -->
65# <dependency>
66#   <groupId>org.junit.jupiter</groupId>
67#   <artifactId>junit-jupiter-params</artifactId>
68#   <scope>test</scope>
69# </dependency>
70# </dependencies>
71#
72# <build>
73#   <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
74#   <plugins>
75#     <plugin>
76#       <artifactId>maven-compiler-plugin</artifactId>
77#       <version>3.10.1</version>
78#       <configuration>
79#         <source>1.8</source>
80#         <target>1.8</target>
81#       </configuration>
82#     </plugin>
83#     <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_lifecycle -->
84#     <plugin>
85#       <artifactId>maven-clean-plugin</artifactId>
86#       <version>3.4.0</version>
87#     </plugin>
88#     <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bins.html#jar-packaging -->
89#     <plugin>
90#       <artifactId>maven-resources-plugin</artifactId>
91#       <version>3.3.1</version>
92#     </plugin>
93#     <plugin>
94#       <artifactId>maven-compiler-plugin</artifactId>
95#       <version>3.13.0</version>
96#     </plugin>
97#     <plugin>
98#       <artifactId>maven-surefire-plugin</artifactId>
99#       <version>3.3.0</version>
100#     </plugin>
101#     <plugin>
102#       <artifactId>maven-jar-plugin</artifactId>
103#       <version>3.4.2</version>
104#     </plugin>
105#     <plugin>
106#       <artifactId>maven-install-plugin</artifactId>
107#       <version>3.1.2</version>
108#     </plugin>
109#     <plugin>
110#       <artifactId>maven-deploy-plugin</artifactId>
111#       <version>3.1.2</version>
112#     </plugin>
113#     <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site_lifecycle -->
114#     <plugin>
115#       <artifactId>maven-site-plugin</artifactId>
116#       <version>3.12.1</version>
117#     </plugin>
118#     <plugin>
119#       <artifactId>maven-project-info-reports-plugin</artifactId>
120#       <version>3.6.1</version>
121#     </plugin>
122#   </plugins>
123# </pluginManagement>
124# </build>
125# </project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Problems | Javadoc | Declaration | Console x

terminated> D:\JSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre

[INFO] -----

```
71# <build>
72#   <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
73#   <plugins>
74#     <plugin>
75#       <artifactId>maven-compiler-plugin</artifactId>
76#       <version>3.10.1</version>
77#       <configuration>
78#         <source>1.8</source>
79#         <target>1.8</target>
80#       </configuration>
81#     </plugin>
82#     <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean_lifecycle -->
83#     <plugin>
84#       <artifactId>maven-clean-plugin</artifactId>
85#       <version>3.4.0</version>
86#     </plugin>
87#     <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bins.html#jar-packaging -->
88#     <plugin>
89#       <artifactId>maven-resources-plugin</artifactId>
90#       <version>3.3.1</version>
91#     </plugin>
92#     <plugin>
93#       <artifactId>maven-compiler-plugin</artifactId>
94#       <version>3.13.0</version>
95#     </plugin>
96#     <plugin>
97#       <artifactId>maven-surefire-plugin</artifactId>
98#       <version>3.3.0</version>
99#     </plugin>
100#     <plugin>
101#       <artifactId>maven-jar-plugin</artifactId>
102#       <version>3.4.2</version>
103#     </plugin>
104#     <plugin>
105#       <artifactId>maven-install-plugin</artifactId>
106#       <version>3.1.2</version>
107#     </plugin>
108#     <plugin>
109#       <artifactId>maven-deploy-plugin</artifactId>
110#       <version>3.1.2</version>
111#     </plugin>
112#     <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site_lifecycle -->
113#     <plugin>
114#       <artifactId>maven-site-plugin</artifactId>
115#       <version>3.12.1</version>
116#     </plugin>
117#     <plugin>
118#       <artifactId>maven-project-info-reports-plugin</artifactId>
119#       <version>3.6.1</version>
120#     </plugin>
121#   </plugins>
122# </pluginManagement>
123# </build>
124# </project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Problems | Javadoc | Declaration | Console x

terminated> App (1) [Java Application] D:\JSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre

[INFO] -----

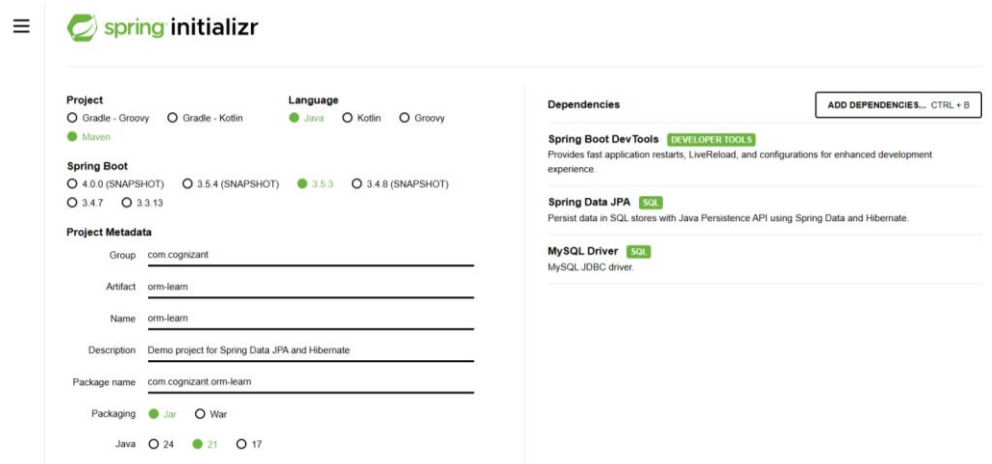
Hello World!

Spring Data JPA with Spring Boot, Hibernate

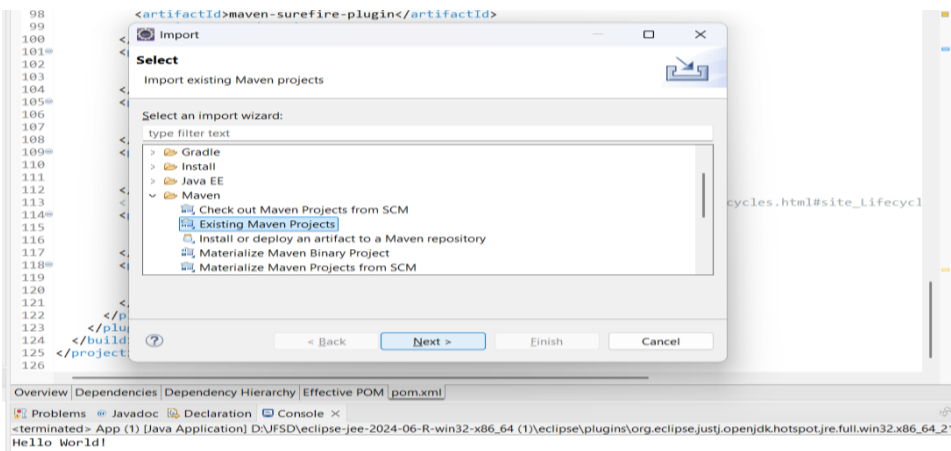
4. Spring Data JPA - Quick Example:

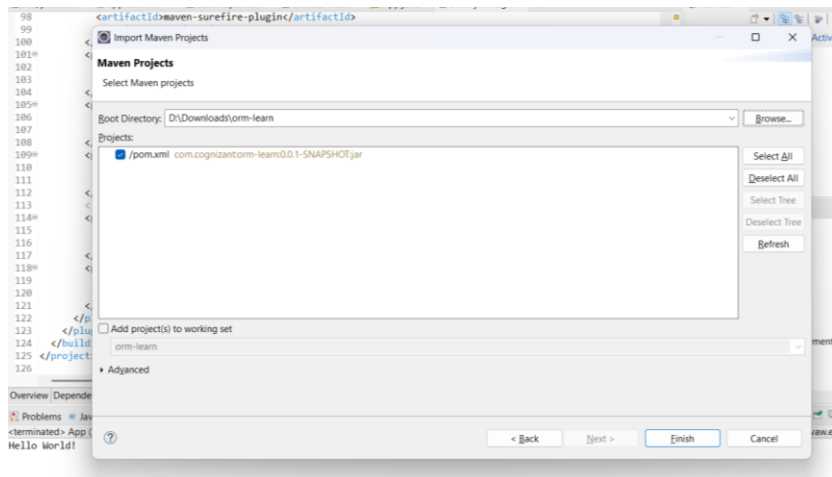
Execution:

Steps :-



The image shows the Spring Initializr web interface for configuring a new project. The 'Project' section has 'Maven' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '3.5.3' selected. The 'Project Metadata' section includes fields for Group (com.cognizant), Artifact (orm-learn), Name (orm-learn), Description (Demo project for Spring Data JPA and Hibernate), and Package name (com.cognizant.orm-learn). The 'Packaging' section has 'Jar' selected. The 'Dependencies' section lists 'Spring Boot DevTools', 'Spring Data JPA', and 'MySQL Driver'.





- orm-learn [boot] [devtools]
 - src/main/java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-21]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml

```
MySQL 8.0 Command Line CLI x + - x
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2255
Server version: 8.0.37 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create schema ormlern;
Query OK, 1 row affected (0.09 sec)

mysql> |
```

```

1 spring.application.name=orm-learn
2
3 # Spring framework and application log
4 logging.level.org.springframework=info
5 logging.level.com.cognizant=debug
6
7 # Hibernate logs for displaying executed SQL, input and output
8 logging.level.org.hibernate.SQL=trace
9 logging.level.org.hibernate.type.descriptor.sql=trace
10
11 # Log pattern
12 logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %s-20.20thread %5p %s-25.25logger(25) %25n %4l %n%n
13
14 # Database configuration
15 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
16 spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn
17 spring.datasource.username=root
18 spring.datasource.password=root
19
20 # Hibernate configuration
21 spring.jpa.hibernate.ddl-auto=validate
22 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
23

```

Problems | Javadoc | Declaration | Console X
 -terminated> App (1) [Java Application] D:\FSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justopenjdk.hotspot.jre.full\win32.x86_64\Hello World!

The screenshot shows an IDE with a Java application configuration and its execution output. The configuration includes logging levels and database settings. The console output shows the application starting successfully, with various warnings and information messages. The application is named 'orm-learn' and is running on a Windows system.

```

1 spring.application.name=orm-learn
2
3 # Spring framework and application log
4 logging.level.org.springframework=info
5 logging.level.com.cognizant=debug
6
7 # Hibernate logs for displaying executed SQL, input and output
8 logging.level.org.hibernate.SQL=trace
9 logging.level.org.hibernate.type.descriptor.sql=trace
10
11 # Log pattern
12 logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %s-20.20thread %5p %s-25.25logger(25) %25n %4l %n%n
13
14 # Database configuration
15 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
16 spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn
17 spring.datasource.username=root
18 spring.datasource.password=root
19
20 # Hibernate configuration
21 spring.jpa.hibernate.ddl-auto=validate
22 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
23

```

Administrator: C:\WINDOWS\system32\cmd.exe X
 system unit 'default'
 06-07-25 00:12:49.770 main INFO :OrmLearnApplicationTests logStarted 59 Started OrmLearnApplicationTests in 4.767 seconds
 (process running for 6.488)
 Mockito is currently self-attaching to enable the inline-mock-maker. This will no longer work in future releases of the JDK. Please add Mockito as an agent to your build as described in Mockito's documentation: <https://javadoc.io/doc/org.mockito/mockito-core/latest/org.mockito/org/mockito/Mockito.html#0.3>
 WARNING: A Java agent has been loaded dynamically (C:\Users\HPA\.m2\repository\net\bytebuddy\byte-buddy-agent\1.17.6\byte-buddy-agent-1.17.6.jar)
 WARNING: If a serviceability tool is in use, please run with -XX:EnableDynamicAgentLoading to hide this warning
 WARNING: Dynamic loading of agents will be disallowed by default in a future release
 OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 6.423 s -- in com.cognizant.orm_learn.OrmLearnApplicationTests
 [INFO] Results:
 [INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
 [INFO] --- jar:3.4.2:jar (default-jar) @ orm-learn ---
 [INFO] Building jar: D:\Downloads\orm-learn\target\orm-learn-0.0.1-SNAPSHOT.jar
 [INFO] --- spring-boot:3.5.3:repackage (repackage) @ orm-learn ---
 [INFO] Replicating main artifact D:\Downloads\orm-learn\target\orm-learn-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
 [INFO] The original artifact has been renamed to D:\Downloads\orm-learn\target\orm-learn-0.0.1-SNAPSHOT.jar.original
 [INFO] BUILD SUCCESS
 [INFO] Total time: 13.050 s
 [INFO] Finished at: 2025-07-06T00:12:52+05:30
 [INFO]

The screenshot shows an IDE with a Java application configuration and its execution output. The configuration includes logging levels and database settings. The console output shows the application starting successfully, with various warnings and information messages. The application is named 'orm-learn' and is running on a Windows system.

```

1 import org.springframework.boot.SpringApplication;
2
3 # Spring framework and application log
4 logging.level.org.springframework=info
5 logging.level.com.cognizant=debug
6
7 # Hibernate logs for displaying executed SQL, input and output
8 logging.level.org.hibernate.SQL=trace
9 logging.level.org.hibernate.type.descriptor.sql=trace
10
11 # Log pattern
12 logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %s-20.20thread %5p %s-25.25logger(25) %25n %4l %n%n
13
14 # Database configuration
15 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
16 spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn
17 spring.datasource.username=root
18 spring.datasource.password=root
19
20 # Hibernate configuration
21 spring.jpa.hibernate.ddl-auto=validate
22 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
23

```

Administrator: C:\WINDOWS\system32\cmd.exe X
 [INFO]
 [INFO] --- jar:3.4.2:jar (default-jar) @ orm-learn ---
 [INFO] Building jar: D:\Downloads\orm-learn\target\orm-learn-0.0.1-SNAPSHOT.jar
 [INFO] --- spring-boot:3.5.3:repackage (repackage) @ orm-learn ---
 [INFO] Replicating main artifact D:\Downloads\orm-learn\target\orm-learn-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
 [INFO] The original artifact has been renamed to D:\Downloads\orm-learn\target\orm-learn-0.0.1-SNAPSHOT.jar.original
 [INFO] BUILD SUCCESS
 [INFO] Total time: 13.050 s
 [INFO] Finished at: 2025-07-06T00:12:52+05:30
 [INFO]


```
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6 import com.cognizant.ormlearn.model.Country;
7
8 @Repository
9 public interface CountryRepository extends JpaRepository<Country, String> {
10 }
11
12 }
```

Find

Outline

- com.cognizant.ormlearn
- CountryRepository

Problems | Javadoc | Declaration | Console | Terminal

<terminated> orm-learn - OrmLearnApplication [Spring Boot App] D:\FSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.5\jre\bin\java.exe (org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@229d17)

06-07-25 00:15:25.829 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4003,

```
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import com.cognizant.ormlearn.model.Country;
9 import com.cognizant.ormlearn.repository.CountryRepository;
10
11 import jakarta.transaction.Transactional;
12
13 @Service
14 public class CountryService {
15
16     @Autowired
17     CountryRepository countryRepository;
18
19     @Transactional
20     public List<Country> getAllCountries() {
21         return countryRepository.findAll();
22     }
23 }
24 }
```

Find

Outline

- com
- CountryService
- CountryRepository

Problems | Javadoc | Declaration | Console | Terminal

<terminated> orm-learn - OrmLearnApplication [Spring Boot App] D:\FSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.5\jre\bin\java.exe (org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@229d17)

06-07-25 00:15:25.829 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4003,

```
18 private static CountryService countryService;
19
20 private static void testGetAllCountries() {
21     LOGGER.info("Start");
22     List<Country> countries = countryService.getAllCountries();
23     for(Country c : countries) {
24         System.out.println(c);
25     }
26     LOGGER.debug("countries={}", countries);
27     LOGGER.info("End");
28 }
29
30 public static void main(String[] args) {
31     ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);
32     countryService = context.getBean(CountryService.class);
33     testGetAllCountries();
34     LOGGER.info("Inside main");
35 }
36 }
```

Find

Outline

- com.cognizant.ormlearn
- OrmLearnApplication
- LOGGER: Logger
- CountryService: CountryService
- testGetAllCountries(): void
- main(String[]): void

Problems | Javadoc | Declaration | Console | Terminal

<terminated> orm-learn - OrmLearnApplication [Spring Boot App] D:\FSD\ eclipse-jee-2024-06-R-win32-x86_64 (1)\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.5\jre\bin\java.exe (org.hibernate.type.descriptor.sql.internal.CapacityDependentDdlType@229d17)

06-07-25 00:36:28.679 restartedMain INFO p.i.JtaPlatformInitiator initiateService 59 H00000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)

06-07-25 00:36:28.733 restartedMain INFO EntityManagerFactoryBean buildNativeEntityManagerFactory 447 Initialized JPA EntityManagerFactory for persistence unit 'default'

06-07-25 00:36:29.052 restartedMain INFO OptionalLiveReloadServer startServer 59 LiveReload server is running on port 35729

06-07-25 00:36:29.080 restartedMain INFO c.c.o.OrmLearnApplication logStarted 59 Started OrmLearnApplication in 3.036 seconds (process running for 3.732)

06-07-25 00:36:29.085 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 22 Start

06-07-25 00:36:29.223 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0 where c1_0.co_code=? and c1_0.co_name=?

Country [code=IN, name=India, getCode()=IN, getName()=India]

Country [code=US, name=United States of America, getCode()=US, getName()=United States of America]

06-07-25 00:36:29.264 restartedMain DEBUG c.c.o.OrmLearnApplication testGetAllCountries 27 countries=[Country [code=IN, name=India, getCode()=IN, getName()=India], Country [code=US, getName()=United States of America, getCode()=US, getName()=United States of America]]

06-07-25 00:36:29.264 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 28 End

06-07-25 00:36:29.264 restartedMain INFO c.c.o.OrmLearnApplication main 36 Inside main

06-07-25 00:36:29.885 restartedMain INFO EntityManagerFactoryBean destroy 660 Closing JPA EntityManagerFactory for persistence unit 'default'

06-07-25 00:36:29.277 licationShutdownHook INFO c.z.h.HikariDataSource close 349 HikariPool-1 - Shutdown initiated...

06-07-25 00:36:29.289 licationShutdownHook INFO c.z.h.HikariDataSource close 351 HikariPool-1 - Shutdown completed.

In Final Output, we retrieved the data from database.

5. Difference between JPA, Hibernate and Spring Data JPA:

Java Persistence API (JPA):

- A *specification* (interface-only) for object-relational mapping (ORM) in Java.
- Contains only interfaces, annotations like @Entity, @Id, @OneToMany, etc.
- Does not include an implementation – it relies on providers like Hibernate or EclipseLink.

Hibernate :

- A *concrete implementation* of the JPA specification.
- Also includes its own APIs, like Session, Criteria, HQL (Hibernate Query Language).

Implementation :

```
public Integer addEmployee(Employee employee) {
    Session session = factory.openSession();
    Transaction tx = null;
    Integer employeeID = null;

    try {
        tx = session.beginTransaction();
        employeeID = (Integer) session.save(employee);
        tx.commit();
    } catch (HibernateException e) {
        if (tx != null) tx.rollback();
        e.printStackTrace();
    } finally {
        session.close();
    }
    return employeeID;
}
```

Spring Data JPA :

- A *Spring-based abstraction* on top of JPA providers (like Hibernate).
- Does not implement JPA but relies on providers like Hibernate.
- Removes boilerplate code.

EmployeeRepository.java:

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer>
{
}
```

EmployeeService.java:

```
@Autowired
```

```
private EmployeeRepository employeeRepository;
```

```
@Transactional
```

```
public void addEmployee(Employee employee) {
    employeeRepository.save(employee);
}
```