

## WEEK – 3

### Spring Core and Maven

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

**File Name : pom.xml**

```
<dependencies>
  <!-- Spring Core Dependency -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.36</version>
  </dependency>
</dependencies>
```

**File Name : 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 -->
  <bean id="bookService" class="com.library.service.BookService">
    <property name="bookRepository" ref="bookRepository"/>
  </bean>

</beans>
```

**File Name : BookRepository.java**

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

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

**File Name : BookService.java**

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

```
import com.library.repository.BookRepository;
```

```
public class BookService {  
    private BookRepository bookRepository;  
  
    // Setter-based injection  
    public void setBookRepository(BookRepository bookRepository) {  
        this.bookRepository = bookRepository;  
    }  
  
    public void addBook(String bookName) {  
        System.out.println("Adding book: " + bookName);  
        bookRepository.saveBook(bookName);  
    }  
}
```

**File Name : LibraryApp.java**

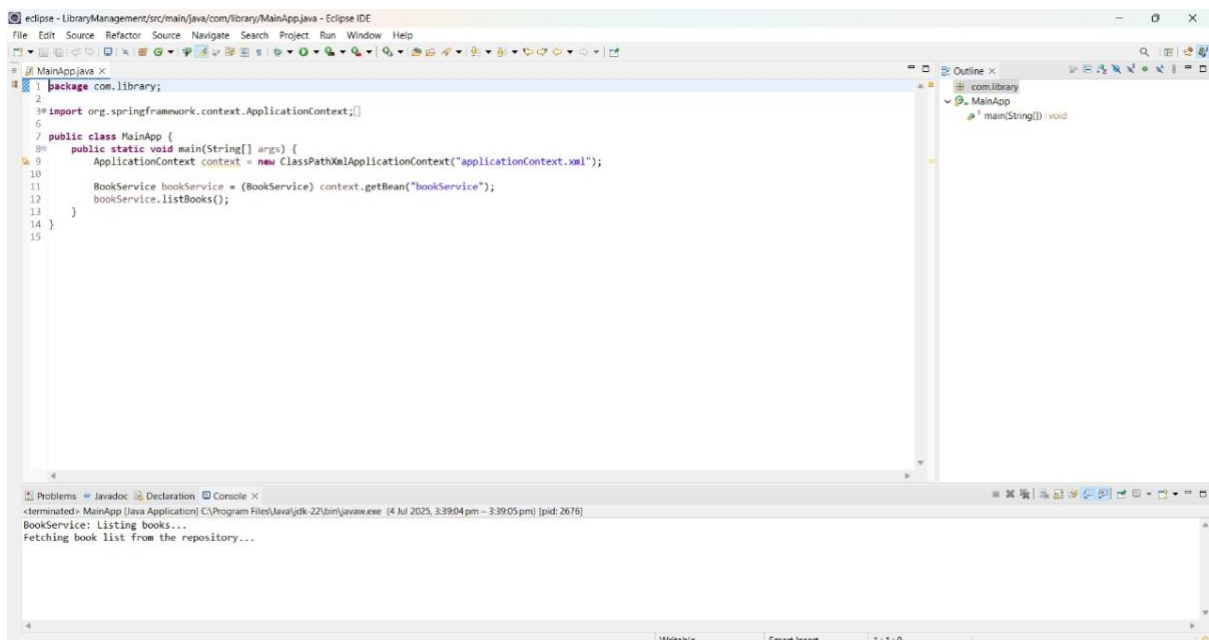
```
package com.library.main;
```

```
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 the BookService bean  
BookService bookService = (BookService) context.getBean("bookService");  
  
// Test it  
bookService.addBook("The Lord of the Rings");  
}  
}
```

## OUTPUT:



## Exercise 2: Implementing Dependency Injection

**File Name : 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>
```

### **File Name : BookService.java**

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

```
import com.library.repository.BookRepository;
```

```
public class BookService {
```

```
    private BookRepository bookRepository;
```

```
    // Setter required for setter-based DI
```

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

```
        this.bookRepository = bookRepository;
```

```
    }
```

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

```
        System.out.println("Adding book: " + bookName);
```

```
        bookRepository.saveBook(bookName);
```

```
    }
```

```
}
```

### **File Name :LibraryApp.java**

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

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

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

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

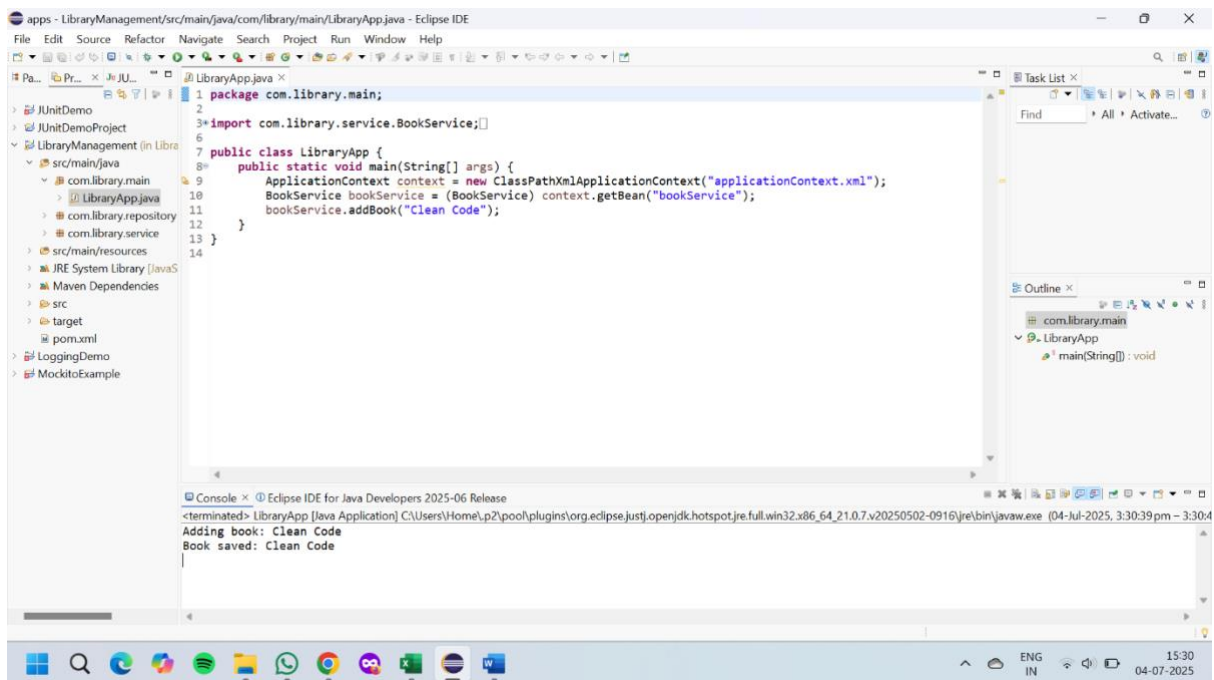
```
public class LibraryApp {
```

```

public static void main(String[] args) {
    ApplicationContext context = new
    ClassPathXmlApplicationContext("applicationContext.xml");
    BookService bookService = (BookService) context.getBean("bookService");
    bookService.addBook("Clean Code");
}
}

```

## OUTPUT:



## Exercise 4: Creating and Configuring a Maven Project

### File Name : 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>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.36</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>
    <version>5.3.36</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.36</version>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.10.1</version>
      <configuration>
        <source>1.8</source>
        <target>1.8</target>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>

```

**File Name : 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 -->
<bean id="bookService" class="com.library.service.BookService">
    <property name="bookRepository" ref="bookRepository"/>
</bean>

</beans>
```

**File Name : BookRepository.java**

```
package com.library.repository;

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

**File Name : BookService.java**

```
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    // Setter-based injection
    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }
}
```

```
public void addBook(String bookName) {  
    System.out.println("Adding book: " + bookName);  
    bookRepository.saveBook(bookName);  
}  
}
```

**File Name : LibraryApp.java**

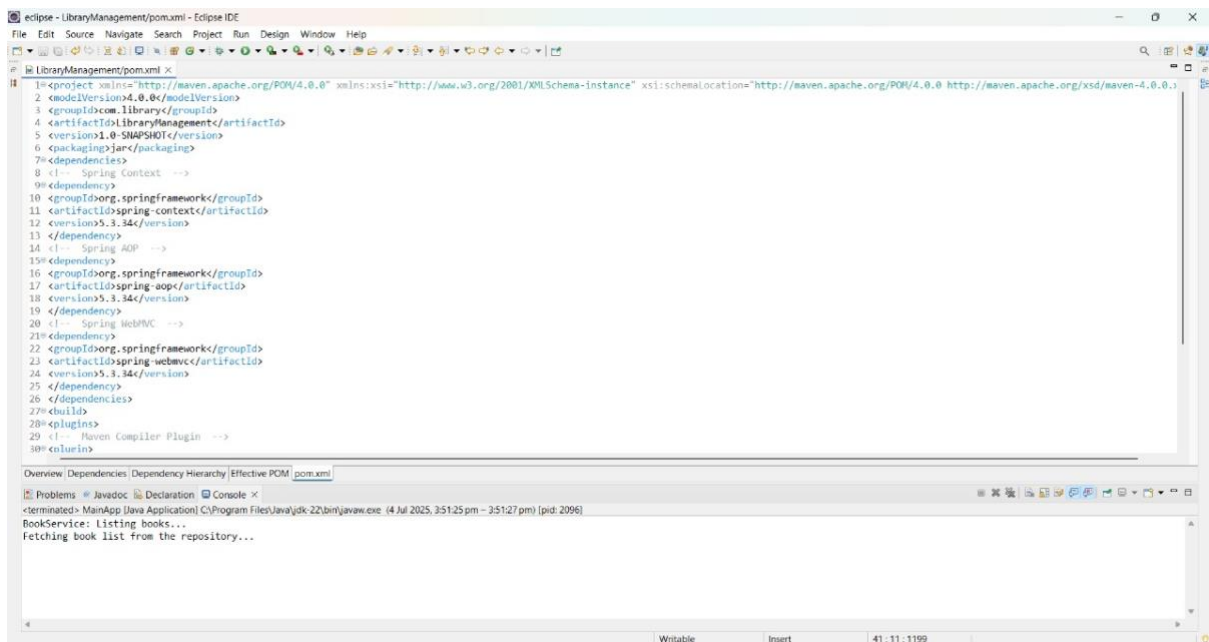
```
package com.library.main;
```

```
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 the BookService bean  
        BookService bookService = (BookService) context.getBean("bookService");  
  
        // Test it  
        bookService.addBook("The Lord of the Rings");  
    }  
}
```



## OUTPUT:



The screenshot shows the Eclipse IDE with a Maven POM file open in the editor. The POM file is for a project named 'LibraryManagement' with version '1.0-SNAPSHOT'. It includes dependencies for Spring Context, Spring AOP, Spring Framework, and Spring WebMVC. The console output at the bottom shows the following messages:

```
<terminated> MainApp [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (4 Jul 2025, 3:51:25 pm -- 3:51:27 pm) [pid: 2096]
BookService: Listing books...
Fetching book list from the repository...
```

### 1. spring-data-jpa-handson

#### Spring Data JPA - Quick Example

File Name : OrmLearnApplication.java

```
package com.cognizant.ormlearn;
```

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

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

```
import org.springframework.boot.SpringApplication;
```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

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

```
import com.cognizant.ormlearn.model.Country;
```

```
import com.cognizant.ormlearn.service.CountryService;
```

```
@SpringBootApplication
```

```
public class OrmLearnApplication {
```

```
    private          static          final          Logger          LOGGER          =
```

```
    LoggerFactory.getLogger(OrmLearnApplication.class);
```

```
    private static CountryService countryService;
```

```
public static void main(String[] args) {  
    ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);  
    countryService = context.getBean(CountryService.class);  
  
    LOGGER.info("Inside main");  
    testGetAllCountries();  
}
```

```
private static void testGetAllCountries() {  
    LOGGER.info("Start");  
    List<Country> countries = countryService.getAllCountries();  
    System.out.println("countries = " + countries); // Console Output  
    LOGGER.info("End");  
}  
}
```

**File Name : Country.java**

```
package com.cognizant.ormlearn.model;
```

```
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.Table;
```

```
@Entity  
@Table(name = "country")  
public class Country {
```

```
    @Id  
    @Column(name = "co_code")  
    private String code;
```

```
    @Column(name = "co_name")  
    private String name;
```

```
public String getCode() {  
    return code;  
}
```

```
public void setCode(String code) {  
    this.code = code;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
@Override
```

```
public String toString() {  
    return "Country [code=" + code + ", name=" + name + "];"  
}  
}
```

**File Name : CountryRepository.java**

```
package com.cognizant.ormlearn.repository;
```

```
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
import com.cognizant.ormlearn.model.Country;
```

```
@Repository
```

```
public interface CountryRepository extends JpaRepository<Country, String> {  
}
```

**File Name : CountryService.java**

```
package com.cognizant.ormlearn.service;
```

```
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.cognizant.ormlearn.model.Country;
import com.cognizant.ormlearn.repository.CountryRepository;
```

```
@Service
```

```
public class CountryService {
```

```
    @Autowired
```

```
    private CountryRepository countryRepository;
```

```
    @Transactional
```

```
    public List<Country> getAllCountries() {
```

```
        return countryRepository.findAll();
```

```
    }
```

```
}
```

**File Name : application.properties**

```
# Logging
```

```
logging.level.org.springframework=info
```

```
logging.level.com.cognizant=debug
```

```
logging.level.org.hibernate.SQL=trace
```

```
logging.level.org.hibernate.type.descriptor.sql=trace
```

```
logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-
25.25logger{25} %25M %4L %m%n
```

```
# Database
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn
```

```
spring.datasource.username=root
```

```
spring.datasource.password=root
```

```
# Hibernate
```

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

**File Name : 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.cognizant</groupId>
```

```
<artifactId>orm-learn</artifactId>
```

```
<version>0.0.1-SNAPSHOT</version>
```

```
<packaging>jar</packaging>
```

```
<name>orm-learn</name>
```

```
<description>Demo project for Spring Data JPA and Hibernate</description>
```

```
<parent>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-parent</artifactId>
```

```
    <version>2.7.0</version>
```

```
    <relativePath/>
```

```
</parent>
```

```
<properties>
```

```
<java.version>11</java.version>

</properties>

<dependencies>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-data-jpa</artifactId>

  </dependency>

  <dependency>

    <groupId>mysql</groupId>

    <artifactId>mysql-connector-java</artifactId>

    <scope>runtime</scope>

  </dependency>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-devtools</artifactId>

    <scope>runtime</scope>

  </dependency>

  <dependency>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-test</artifactId>

    <scope>test</scope>

  </dependency>

</dependencies>
```

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

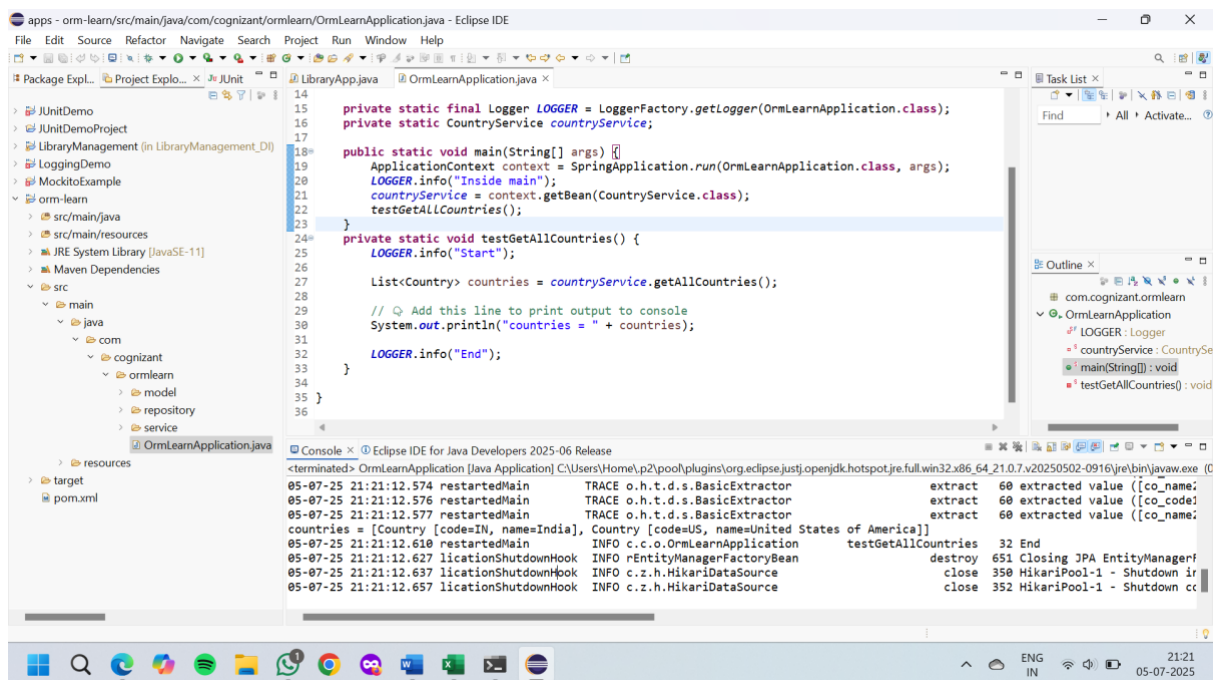
</plugin>

</plugins>

</build>

</project>

## OUTPUT:



The screenshot shows the Eclipse IDE interface. The main editor displays the `OrmLearnApplication.java` file. The code includes a `main` method that runs a Spring application and a `testGetAllCountries` method that prints the output of `countryService.getAllCountries()` to the console. The console output shows the application starting, logging the start and end times, and printing the list of countries: `countries = [Country {code=IN, name=India}, Country {code=US, name=United States of America}]`. The console also shows the application shutting down and closing the JPA EntityManagerFactory and HikariPool-1.

```
private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);
private static CountryService countryService;

public static void main(String[] args) {
    ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);
    LOGGER.info("Inside main");
    countryService = context.getBean(CountryService.class);
    testGetAllCountries();
}

private static void testGetAllCountries() {
    LOGGER.info("Start");

    List<Country> countries = countryService.getAllCountries();

    // Add this line to print output to console
    System.out.println("countries = " + countries);

    LOGGER.info("End");
}
```

Console Output:

```
<terminated> OrmLearnApplication [Java Application] C:\Users\Home\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.21.0.7.v20250502-0916\jre\bin\javaw.exe (C
05-07-25 21:21:12.574 restartedMain TRACE o.h.t.d.s.BasicExtractor extract 60 extracted value {[co_name:
05-07-25 21:21:12.576 restartedMain TRACE o.h.t.d.s.BasicExtractor extract 60 extracted value {[co_code:
05-07-25 21:21:12.577 restartedMain TRACE o.h.t.d.s.BasicExtractor extract 60 extracted value {[co_name:
countries = [Country {code=IN, name=India}, Country {code=US, name=United States of America}]
05-07-25 21:21:12.610 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 32 End
05-07-25 21:21:12.627 licationShutdownHook INFO r.EntityManagerFactoryBean destroy 651 Closing JPA EntityManagerF
05-07-25 21:21:12.637 licationShutdownHook INFO c.z.h.HikariDataSource close 350 HikariPool-1 - Shutdown ir
05-07-25 21:21:12.657 licationShutdownHook INFO c.z.h.HikariDataSource close 352 HikariPool-1 - Shutdown cc
```

apps - orm-learn/src/main/java/com/cognizant/ormlearn/OrmLearnApplication.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Exp... Project Explo... JUnit LibraryApp.java OrmLearnApplication.java

JUnitDemo JUnitDemoProject LibraryManagement (in LibraryManagement\_D) LoggingDemo MockitoExample orm-learn src/main/java src/main/resources JRE System Library [JavaSE-11] Maven Dependencies src main java com cognizant ormlearn model repository service resources target pom.xml

Console x Eclipse IDE for Java Developers 2025-06 Release

<terminated> OrmLearnApplication [Java Application] C:\Users\Home\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64.21.0.7.v20250502-0916\jre\bin\javaw.exe (C: 21:21:06.612 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader

Spring Boot :: (v2.7.0)

```
05-07-25 21:21:07.220 restartedMain INFO c.c.o.OrmLearnApplication logStarting 55 Starting OrmLearnApplicati
05-07-25 21:21:07.227 restartedMain DEBUG c.c.o.OrmLearnApplication logStarting 56 Running with Spring Boot \
05-07-25 21:21:07.229 restartedMain INFO c.c.o.OrmLearnApplication logStartupProfileInfo 634 No active profile set, fal
05-07-25 21:21:07.318 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 255 Devtools property default:
05-07-25 21:21:08.010 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 132 Bootstrapping Spring Data
05-07-25 21:21:08.089 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 201 Finished Spring Data repo:
05-07-25 21:21:08.801 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 31 HH0000204: Processing
05-07-25 21:21:08.905 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 44 HH0000412: Hibernate ORM (
05-07-25 21:21:09.202 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 56 HCANN000001: Hibernate Con
05-07-25 21:21:09.527 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 110 HikariPool-1 - Starting...
05-07-25 21:21:10.358 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 123 HikariPool-1 - Start compl
05-07-25 21:21:10.391 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 175 HH0000400: Using dialect:
05-07-25 21:21:11.594 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 52 HH0000490: Using JtaPlatf
05-07-25 21:21:11.610 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 437 Initialized JPA Enti
05-07-25 21:21:12.285 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 58 LiveReload server is runni
05-07-25 21:21:12.322 restartedMain INFO c.c.o.OrmLearnApplication logStarting 61 Started OrmLearnApplicati
05-07-25 21:21:12.337 restartedMain INFO c.c.o.OrmLearnApplication logStarting 20 Inside main
05-07-25 21:21:12.345 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 25 Start
05-07-25 21:21:12.551 restartedMain DEBUG org.hibernate.SQL logStatement 144 select country@_co_code i
05-07-25 21:21:12.563 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_code:
05-07-25 21:21:12.574 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_name:
05-07-25 21:21:12.574 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_name:
```

apps - orm-learn/src/main/java/com/cognizant/ormlearn/OrmLearnApplication.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Exp... Project Explo... JUnit LibraryApp.java OrmLearnApplication.java

JUnitDemo JUnitDemoProject LibraryManagement (in LibraryManagement\_D) LoggingDemo MockitoExample orm-learn src/main/java src/main/resources JRE System Library [JavaSE-11] Maven Dependencies src main java com cognizant ormlearn model repository service resources target pom.xml

Console x Eclipse IDE for Java Developers 2025-06 Release

<terminated> OrmLearnApplication [Java Application] C:\Users\Home\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64.21.0.7.v20250502-0916\jre\bin\javaw.exe (C: 21:21:06.612 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader

Spring Boot :: (v2.7.0)

```
05-07-25 21:21:07.220 restartedMain INFO c.c.o.OrmLearnApplication logStarting 55 Starting OrmLearnApplicati
05-07-25 21:21:07.227 restartedMain DEBUG c.c.o.OrmLearnApplication logStarting 56 Running with Spring Boot \
05-07-25 21:21:07.229 restartedMain INFO c.c.o.OrmLearnApplication logStartupProfileInfo 634 No active profile set, fal
05-07-25 21:21:07.318 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 255 Devtools property default:
05-07-25 21:21:08.010 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 132 Bootstrapping Spring Data
05-07-25 21:21:08.089 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 201 Finished Spring Data repo:
05-07-25 21:21:08.801 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 31 HH0000204: Processing
05-07-25 21:21:08.905 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 44 HH0000412: Hibernate ORM (
05-07-25 21:21:09.202 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 56 HCANN000001: Hibernate Con
05-07-25 21:21:09.527 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 110 HikariPool-1 - Starting...
05-07-25 21:21:10.358 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 123 HikariPool-1 - Start compl
05-07-25 21:21:10.391 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 175 HH0000400: Using dialect:
05-07-25 21:21:11.594 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 52 HH0000490: Using JtaPlatf
05-07-25 21:21:11.610 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 437 Initialized JPA Enti
05-07-25 21:21:12.285 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 58 LiveReload server is runni
05-07-25 21:21:12.322 restartedMain INFO c.c.o.OrmLearnApplication logStarting 61 Started OrmLearnApplicati
05-07-25 21:21:12.337 restartedMain INFO c.c.o.OrmLearnApplication logStarting 20 Inside main
05-07-25 21:21:12.345 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 25 Start
05-07-25 21:21:12.551 restartedMain DEBUG org.hibernate.SQL logStatement 144 select country@_co_code i
05-07-25 21:21:12.563 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_code:
05-07-25 21:21:12.574 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_name:
05-07-25 21:21:12.576 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_code:
05-07-25 21:21:12.577 restartedMain TRACE org.hibernate.SQL extract 60 extracted value ([co_name:
05-07-25 21:21:12.610 restartedMain countries = [Country [code=IN, name=India], Country [code=US, name=United States of America]]
05-07-25 21:21:12.627 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 32 End
05-07-25 21:21:12.637 restartedMain INFO org.springframework.boot.devtools.restart.classloader.RestartClassLoader 651 Closing JPA EntityManager!
05-07-25 21:21:12.637 restartedMain INFO c.c.o.OrmLearnApplication destroy 350 HikariPool-1 - Shutdown in
05-07-25 21:21:12.637 restartedMain INFO c.c.o.OrmLearnApplication destroy 352 HikariPool-1 - Shutdown cc
```



## **Difference between JPA, Hibernate and Spring Data JPA**

### **Hibernate Example**

#### **File Name : pom.xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0">
```

```
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>com.example</groupId>
```

```
  <artifactId>HibernateExample</artifactId>
```

```
  <version>1.0</version>
```

```
  <dependencies>
```

```
    <dependency>
```

```
      <groupId>org.hibernate</groupId>
```

```
      <artifactId>hibernate-core</artifactId>
```

```
      <version>5.6.15.Final</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>mysql</groupId>
```

```
      <artifactId>mysql-connector-java</artifactId>
```

```
      <version>8.0.30</version>
```

```
    </dependency>
```

```
  </dependencies>
```

```
</project>
```

#### **File Name : hibernate.cfg.xml**

```
<!DOCTYPE hibernate-configuration PUBLIC
```

```
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
```

```

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

    <session-factory>

        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>

        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/testdb</property>

        <property name="hibernate.connection.username">root</property>

        <property name="hibernate.connection.password">root</property>

        <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>

        <property name="hibernate.hbm2ddl.auto">update</property>

        <mapping class="com.example.Employee"/>

    </session-factory>

</hibernate-configuration>

```

**File Name : Employee.java**

```

package com.example;

import javax.persistence.*;

@Entity

@Table(name = "employee")

public class Employee {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    private String name;

    private String department;

```

```
// Getters and setters

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}
```

**File Name : MainApp.java**

```
package com.example;

import org.hibernate.*;

import org.hibernate.cfg.Configuration;

public class MainApp {

    public static void main(String[] args) {

        SessionFactory factory = new Configuration().configure().buildSessionFactory();

        Session session = factory.openSession();

        Transaction tx = null;

        try {

            tx = session.beginTransaction();

            Employee emp = new Employee();

            emp.setName("John");

            emp.setDepartment("IT");

            session.save(emp);

            tx.commit();

        }

    }

}
```

```

        System.out.println("Employee saved with ID: " + emp.getId());
    } catch (Exception e) {

        if (tx != null) tx.rollback();

        e.printStackTrace();

    } finally {

        session.close();

        factory.close();

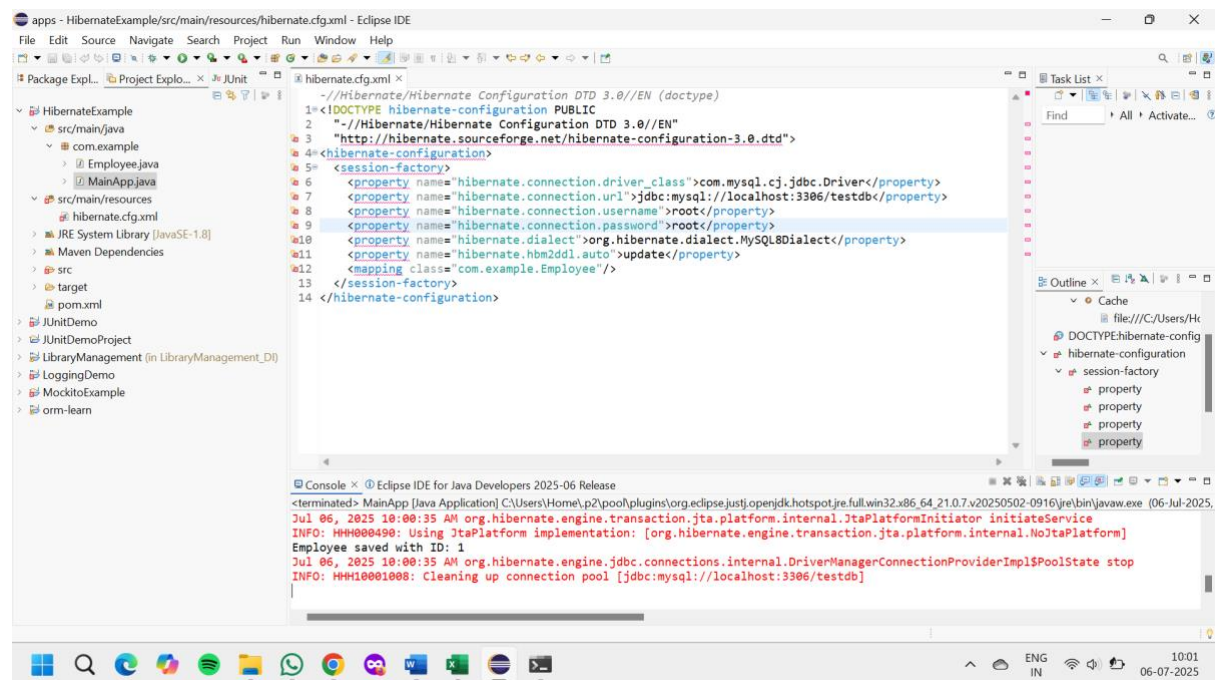
    }

}

}

```

## OUTPUT:



## Spring Data JPA Example

**File Name : pom.xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>SpringDataJPAExample</artifactId>

  <version>1.0</version>

  <parent>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-starter-parent</artifactId>

    <version>2.7.15</version>

  </parent>

  <dependencies>

    <dependency>

      <groupId>org.springframework.boot</groupId>

      <artifactId>spring-boot-starter-data-jpa</artifactId>

    </dependency>

    <dependency>

      <groupId>mysql</groupId>

      <artifactId>mysql-connector-java</artifactId>

      <version>8.0.30</version>

    </dependency>

  </dependencies>

</project>
```

**File Name : application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/testdb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

**File Name : Employee.java**

package com.example.entity;

import javax.persistence.\*;

@Entity

public class Employee {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;

    private String name;

    private String department;

    // Getters and setters

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public String getDepartment() { return department; }

    public void setDepartment(String department) { this.department = department; }

}

**File Name : EmployeeRepository.java**

```
package com.example.repository;

import com.example.entity.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}
```

**File Name : EmployeeService.java**

```
package com.example.service;

import com.example.entity.Employee;

import com.example.repository.EmployeeRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

@Service

public class EmployeeService {

    @Autowired

    private EmployeeRepository employeeRepository;

    @Transactional

    public void addEmployee(Employee emp) {

        employeeRepository.save(emp);

    }

}
```

**File Name : SpringDataJpaApp.java**

```
package com.example;

import com.example.entity.Employee;
```

```
import com.example.service.EmployeeService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringDataJpaApp implements CommandLineRunner {

    @Autowired

    private EmployeeService employeeService;

    public static void main(String[] args) {

        SpringApplication.run(SpringDataJpaApp.class, args);

    }

    @Override

    public void run(String... args) {

        Employee emp = new Employee();

        emp.setName("Alice");

        emp.setDepartment("HR");

        employeeService.addEmployee(emp);

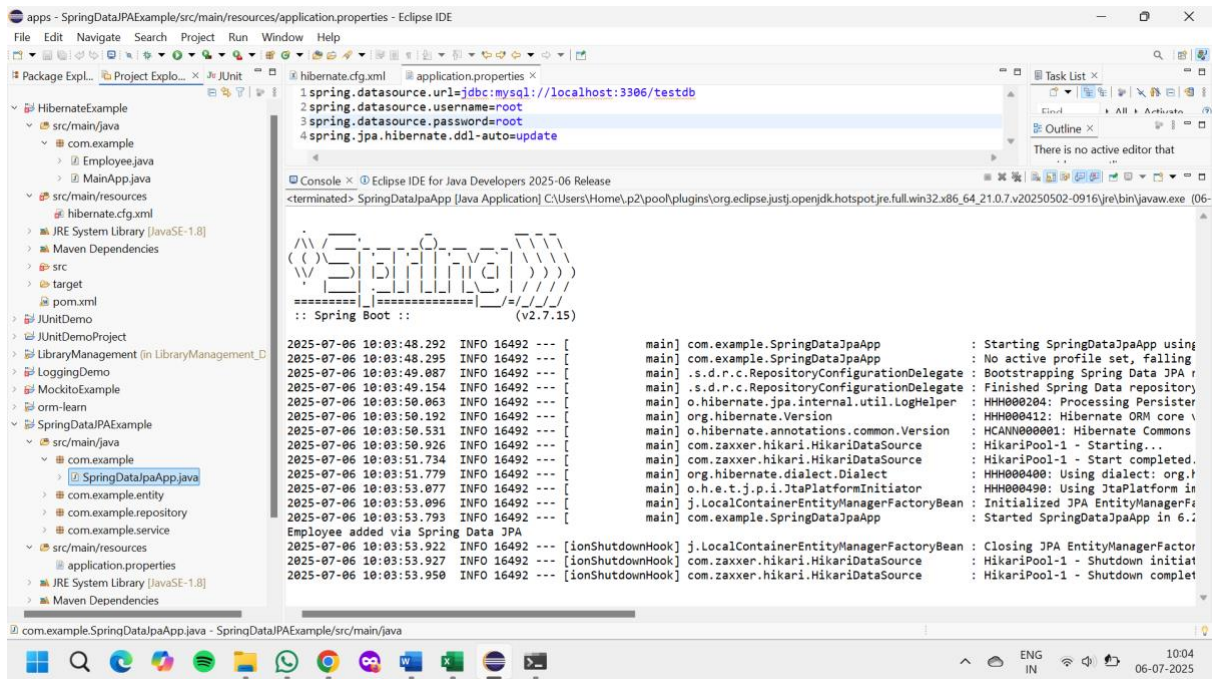
        System.out.println("Employee added via Spring Data JPA");

    }

}
```



## OUTPUT:



```
apps - SpringDataJPAExample/src/main/resources/application.properties - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Expl... Project Expl... x JUnit
hibernate.cfg.xml application.properties x
1 spring.datasource.url=jdbc:mysql://localhost:3306/testdb
2 spring.datasource.username=root
3 spring.datasource.password=root
4 spring.jpa.hibernate.ddl-auto=update

Console x Eclipse IDE for Java Developers 2025-06 Release
<terminated> SpringDataJpaApp [Java Application] C:\Users\Home\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (06-

Spring Boot :: (v2.7.15)

2025-07-06 10:03:48.292 INFO 16492 --- [main] com.example.SpringDataJpaApp : Starting SpringDataJpaApp using
2025-07-06 10:03:48.295 INFO 16492 --- [main] com.example.SpringDataJpaApp : No active profile set, falling
2025-07-06 10:03:49.087 INFO 16492 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA r
2025-07-06 10:03:49.154 INFO 16492 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository
2025-07-06 10:03:50.063 INFO 16492 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000284: Processing Persister
2025-07-06 10:03:50.192 INFO 16492 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core \
2025-07-06 10:03:50.531 INFO 16492 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons
2025-07-06 10:03:50.926 INFO 16492 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-07-06 10:03:51.734 INFO 16492 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-07-06 10:03:51.779 INFO 16492 --- [main] org.hibernate.dialect.Dialect : HHH000490: Using dialect: org.h
2025-07-06 10:03:53.077 INFO 16492 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform in
2025-07-06 10:03:53.096 INFO 16492 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFi
2025-07-06 10:03:53.793 INFO 16492 --- [main] com.example.SpringDataJpaApp : Started SpringDataJpaApp in 6.2
Employee added via Spring Data JPA
2025-07-06 10:03:53.922 INFO 16492 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactor
2025-07-06 10:03:53.927 INFO 16492 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiat
2025-07-06 10:03:53.950 INFO 16492 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown complet

com.example.SpringDataJpaApp.java - SpringDataJPAExample/src/main/java
```