

COGNIZANT DIGITAL NURTURE 4.0 (JAVA FSE)

Week 3 - Spring Core and Maven

1.Spring Core_Maven

i) 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</version>
  <dependencies>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.34</version>
    </dependency>
  </dependencies>
</project>
```

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

</beans>
```

BookRepository.java

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

```
public class BookRepository {  
    public String getBookName() {  
        return "Effective Java";  
    }  
}
```

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 displayBook() {  
        System.out.println("Book: " + bookRepository.getBookName());  
    }  
}
```

MainApp.java

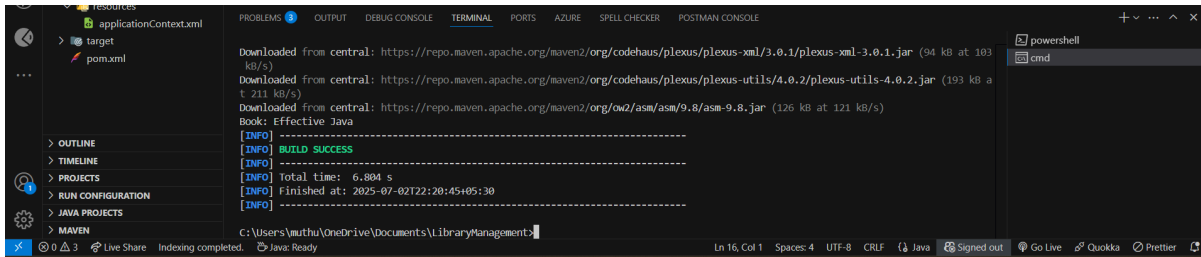
```
package com.library;
```

```
import com.library.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;
```

```
public class MainApp {  
    public static void main(String[] args) {  
        ApplicationContext context = new  
ClassPathXmlApplicationContext("applicationContext.xml");  
        BookService service = (BookService) context.getBean("bookService");  
        service.displayBook();  
    }  
}
```

```
}  
}
```

Output:



ii)Implementing Dependency Injection:

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.library.repository.BookRepository" />
```

```
<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;
```

```
public class BookService {  
    private BookRepository bookRepository;  
  
    public void setBookRepository(BookRepository bookRepository) {  
        this.bookRepository = bookRepository;  
    }  
  
    public void displayBook() {  
        System.out.println("Book: " + bookRepository.getBookName());  
    }  
}
```

BookRepository.java

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

```
public class BookRepository {  
    public String getBookName() {  
        return "Effective Java";  
    }  
}
```

MainApp.java

```
package com.library;
```

```

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

    public static void main(String[] args) {

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

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

        bookService.displayBook();

    }

}

```

Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE SPELL CHECKER POSTMAN CONSOLE
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-metadata.xml
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-metadata.xml (21 kB at 3.3 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-metadata.xml (14 kB at 2.3 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/exec-maven-plugin/maven-metadata.xml (1.0 kB at 748 B/s)
[INFO] -----< com.library:LibraryManagement >-----
[INFO] Building LibraryManagement 1.0
[INFO] from pom.xml
[INFO] Building LibraryManagement 1.0
[INFO] Building LibraryManagement 1.0
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec:3.5.1:java (default-cli) @ LibraryManagement ---
Book: Effective Java
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.348 s
[INFO] Finished at: 2025-07-03T18:32:47+05:30
[INFO] -----
C:\Users\muthu\OneDrive\Documents\LibraryManagement>

```

iii) Creating and Configuring a Maven Project:

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</version>
```

```
<dependencies>
```

```
  <!-- Spring Context -->
```

```
  <dependency>
```

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

```
    <artifactId>spring-context</artifactId>
```

```
    <version>5.3.34</version>
```

```
  </dependency>
```

```
  <!-- Spring AOP -->
```

```
  <dependency>
```

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

```
    <artifactId>spring-aop</artifactId>
```

```
    <version>5.3.34</version>
```

```
  </dependency>
```

```
  <!-- Spring WebMVC -->
```

```
  <dependency>
```

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

```
    <artifactId>spring-webmvc</artifactId>
```

```
    <version>5.3.34</version>
```

```
  </dependency>
```

```
</dependencies>
```

```
<build>
```

```
  <plugins>
```

```
    <!-- Compiler Plugin -->
```

```
    <plugin>
```

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

```
      <version>3.11.0</version>
```

```
      <configuration>
```

```
        <source>1.8</source>
```

```
        <target>1.8</target>
```

```
      </configuration>
```

```
    </plugin>
```

```
    <!-- Exec Plugin (for running Java main) -->
```

```
    <plugin>
```

```

<groupId>org.codehaus.mojo</groupId>
<artifactId>exec-maven-plugin</artifactId>
<version>3.1.0</version>
<configuration>
  <mainClass>com.library.MainApp</mainClass>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

MainApp.java

```
package com.library;
```

```

public class MainApp {
    public static void main(String[] args) {
        System.out.println("✅ Spring Project Setup Successful!");
    }
}

```

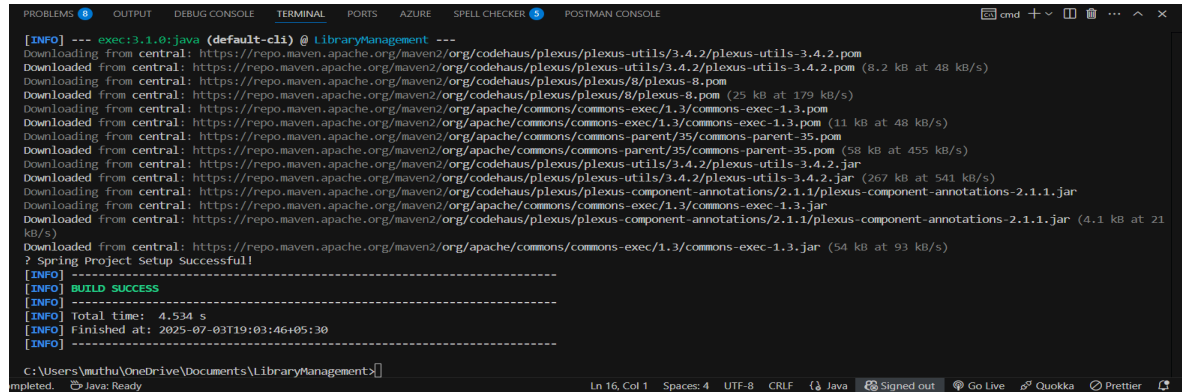
applicationContext.xml

```

<?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">
</beans>

```

Output:



2.Spring Data JPA with Spring Boot, Hibernate spring-data-jpa-handson

i)Spring Data JPA - Quick Example:

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

DATASOURCE

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

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;

// getters, setters, toString()

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 + "]";
}
}

```

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> { }

```

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();
```

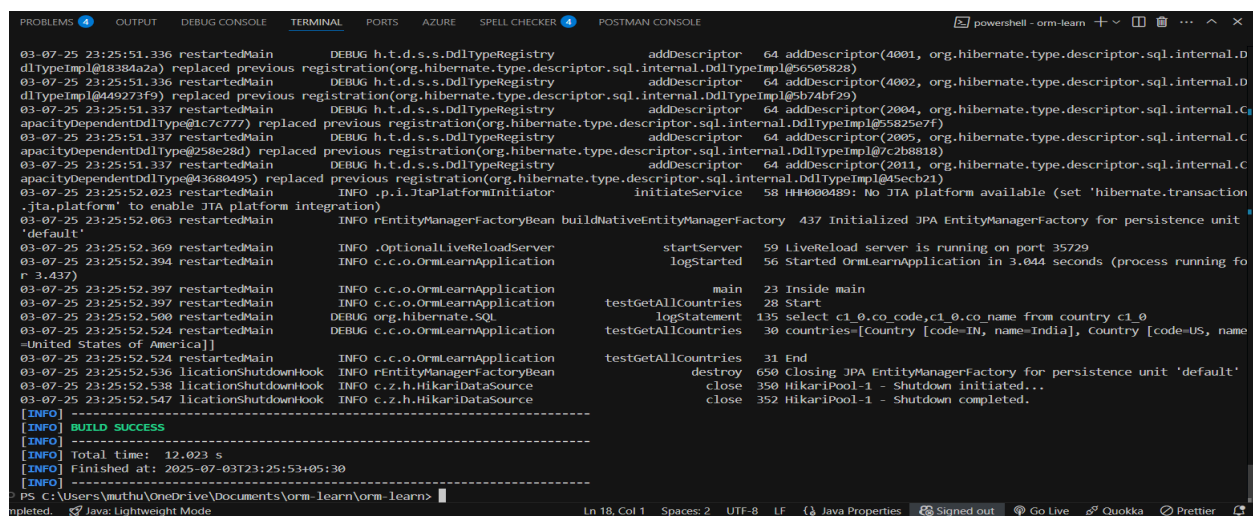
```
        LOGGER.debug("countries={}", countries);
```

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

```
    }
```

```
}
```

Output:



```
03-07-25 23:25:51.336 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4001, org.hibernate.type.descriptor.sql.internal.D
ddlTypeImpl@18384a2a) replaced previous registration(org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@56505828)
03-07-25 23:25:51.336 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(4002, org.hibernate.type.descriptor.sql.internal.D
ddlTypeImpl@949273f0) replaced previous registration(org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@5b74bf29)
03-07-25 23:25:51.337 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2004, org.hibernate.type.descriptor.sql.internal.C
apacityDependentDdlType@1c7c777) replaced previous registration(org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@55825e7f)
03-07-25 23:25:51.337 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2005, org.hibernate.type.descriptor.sql.internal.C
apacityDependentDdlType@258e28d) replaced previous registration(org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@7c2b8818)
03-07-25 23:25:51.337 restartedMain DEBUG h.t.d.s.s.DdlTypeRegistry addDescriptor 64 addDescriptor(2011, org.hibernate.type.descriptor.sql.internal.C
apacityDependentDdlType@43680495) replaced previous registration(org.hibernate.type.descriptor.sql.internal.DdlTypeImpl@45ecb21)
03-07-25 23:25:52.023 restartedMain INFO p.i.JtaPlatformInitiator initiateService 58 HHH000489: No JTA platform available (set 'hibernate.transaction
.jta.platform' to enable JTA platform integration)
03-07-25 23:25:52.063 restartedMain INFO r.EntityManagerFactoryBean buildNativeEntityManagerFactory 437 Initialized JPA EntityManagerFactory for persistence unit
'default'
03-07-25 23:25:52.369 restartedMain INFO .OptionalLiveReloadServer startServer 59 LiveReload server is running on port 35729
03-07-25 23:25:52.394 restartedMain INFO c.c.o.OrmLearnApplication logStarted 56 Started OrmLearnApplication in 3.044 seconds (process running fo
r 3.437)
03-07-25 23:25:52.397 restartedMain INFO c.c.o.OrmLearnApplication main 23 Inside main
03-07-25 23:25:52.397 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 28 Start
03-07-25 23:25:52.500 restartedMain DEBUG org.hibernate.SQL logStatement 135 select c1_0.co_code,c1_0.co_name from country c1_0
03-07-25 23:25:52.524 restartedMain DEBUG c.c.o.OrmLearnApplication testGetAllCountries 30 countries=[Country [code=IN, name=India], Country [code=US, name
=United States of America]]
03-07-25 23:25:52.524 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries 31 End
03-07-25 23:25:52.536 restartedMain INFO r.EntityManagerFactoryBean destroy 650 Closing JPA EntityManagerFactory for persistence unit 'default'
03-07-25 23:25:52.538 restartedMain INFO c.z.h.HikariDataSource close 350 HikariPool-1 - Shutdown initiated...
03-07-25 23:25:52.547 restartedMain INFO c.z.h.HikariDataSource close 352 HikariPool-1 - Shutdown completed.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12.023 s
[INFO] Finished at: 2025-07-03T23:25:53+05:30
[INFO] -----
PS C:\Users\muthu\OneDrive\Documents\orm-learn\orm-learn>
```

ii) Difference between JPA, Hibernate and Spring Data JPA:

```
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;  
}
```

EmployeeRepository.java

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

EmployeeService.java

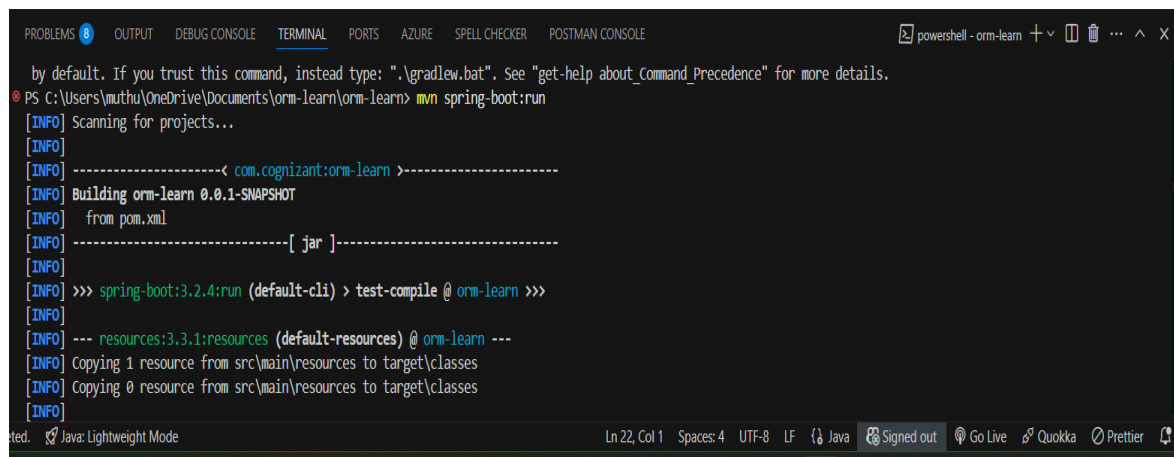
@Autowired

```
private EmployeeRepository employeeRepository;
```

@Transactional

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

Output:



```
by default. If you trust this command, instead type: ".\gradlew.bat". See "get-help about_Command_Precedence" for more details.  
PS C:\Users\muthu\OneDrive\Documents\orm-learn\orm-learn> mvn spring-boot:run  
[INFO] Scanning for projects...  
[INFO]  
[INFO] -----< com.cognizant:orm-learn >-----  
[INFO] Building orm-learn 0.0.1-SNAPSHOT  
[INFO] from pom.xml  
[INFO] -----[ jar ]-----  
[INFO]  
[INFO] >>> spring-boot:3.2.4:run (default-cli) > test-compile @ orm-learn >>>  
[INFO]  
[INFO] --- resources:3.3.1:resources (default-resources) @ orm-learn ---  
[INFO] Copying 1 resource from src/main/resources to target/classes  
[INFO] Copying 0 resource from src/main/resources to target/classes  
[INFO]
```