**Superset ID:** **6403222**

**MODULE 3 SOLUTIONS**

**Spring Core and Maven**

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

**//HelloWorld.java**

package org.example;  
  
public class HelloWorld {  
 private String message;  
  
 public void setMessage(String message){  
 this.message = message;  
 }  
  
 public void getMessage(){  
 System.*out*.println("Your Message: " + message);  
 }  
}

**//MainApp.java**

package org.example;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class MainApp {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("Beans.xml");  
 HelloWorld obj = (HelloWorld) context.getBean("helloWorld");  
 obj.getMessage();  
 }  
}

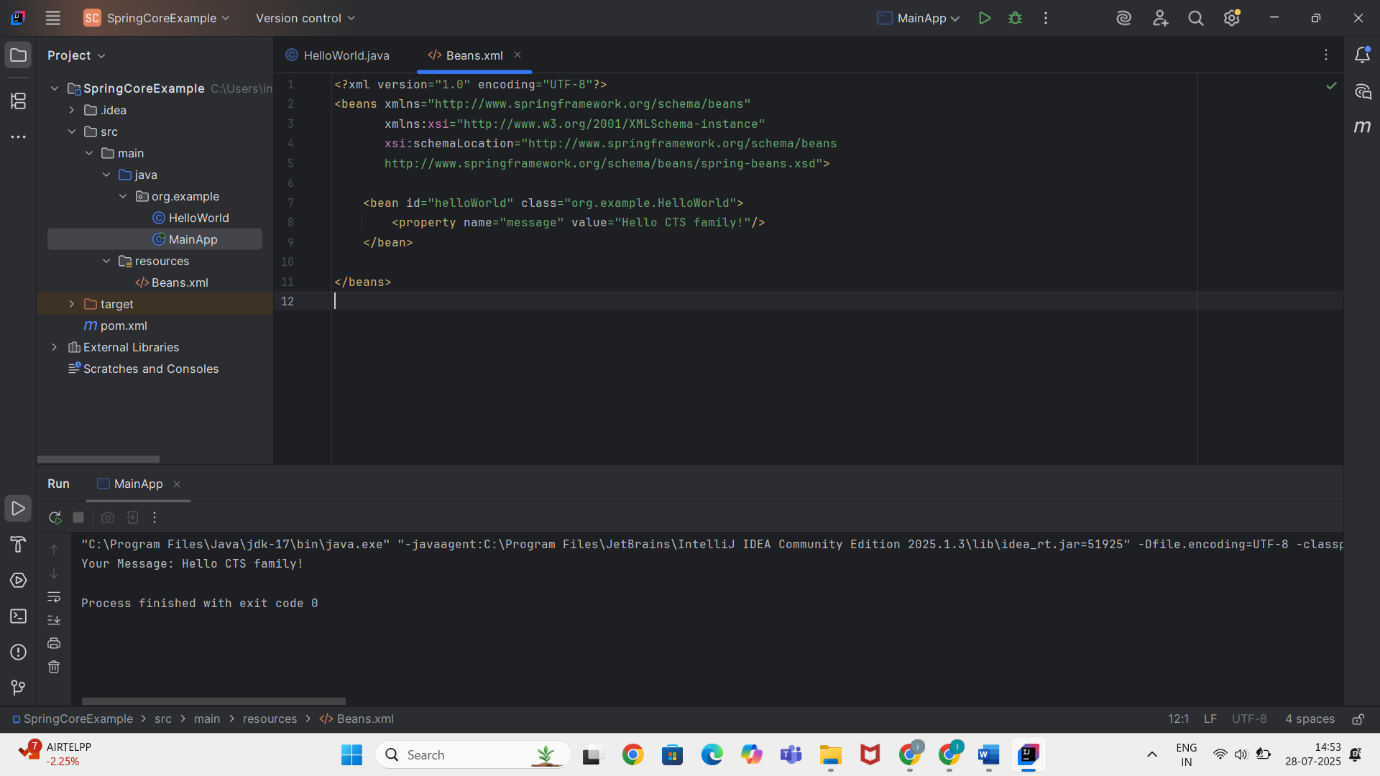
**//pom.xml**

<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>org.example</groupId>  
 <artifactId>SpringCoreExample</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.22</version>  
 </dependency>  
 </dependencies>  
</project>

**//Beans.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="helloWorld" class="org.example.HelloWorld">  
 <property name="message" value="Hello CTS family!"/>  
 </bean>  
  
</beans>

**Output:**

****

**Exercise 2: Implementing Dependency Injection**

**//HelloWorld.java**

package org.example;

public class HelloWorld {

private String message;

public void setMessage(String message) {

this.message = message;

}

public void getMessage() {

System.out.println("Your Message: " + message);

}

}

**//MainApp.java**

package org.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

HelloWorld obj = (HelloWorld) context.getBean("helloWorld");

obj.getMessage();

}

}

**//Beans.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="helloWorld" class="org.example.HelloWorld">

<property name="message" value="Hello Spring Framework!" />

</bean>

</beans>

**//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>org.example</groupId>

<artifactId>SpringCoreExample</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

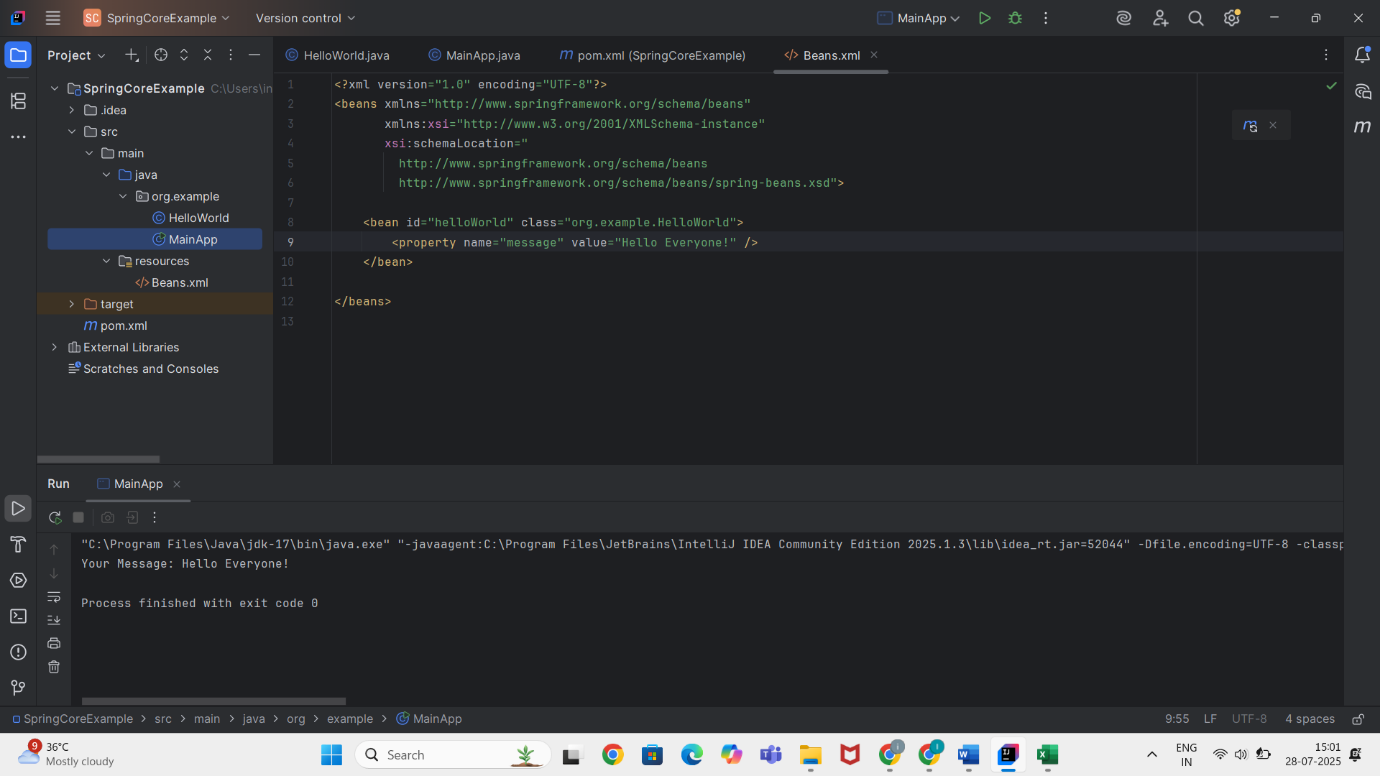
<version>5.3.33</version>

</dependency>

</dependencies>

</project>

**OUTPUT:**

****

**Exercise 4: Creating and Configuring a Maven Project**

**//HelloWorld.java**

package org.example;

public class HelloWorld {

private String message;

public void setMessage(String message) {

this.message = message;

}

public void getMessage() {

System.out.println("Your Message: " + message);

}

}

**//MainApp.java**

package org.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

HelloWorld obj = (HelloWorld) context.getBean("helloWorld");

obj.getMessage();

}

}

**//Beans.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="helloWorld" class="org.example.HelloWorld">

<property name="message" value="Hello from Spring Maven Project!" />

</bean>

</beans>

**//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>org.example</groupId>

<artifactId>SpringMavenExample</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Spring Core Dependency -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.20</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<resources>

<resource>

<directory>src/main/resources</directory>

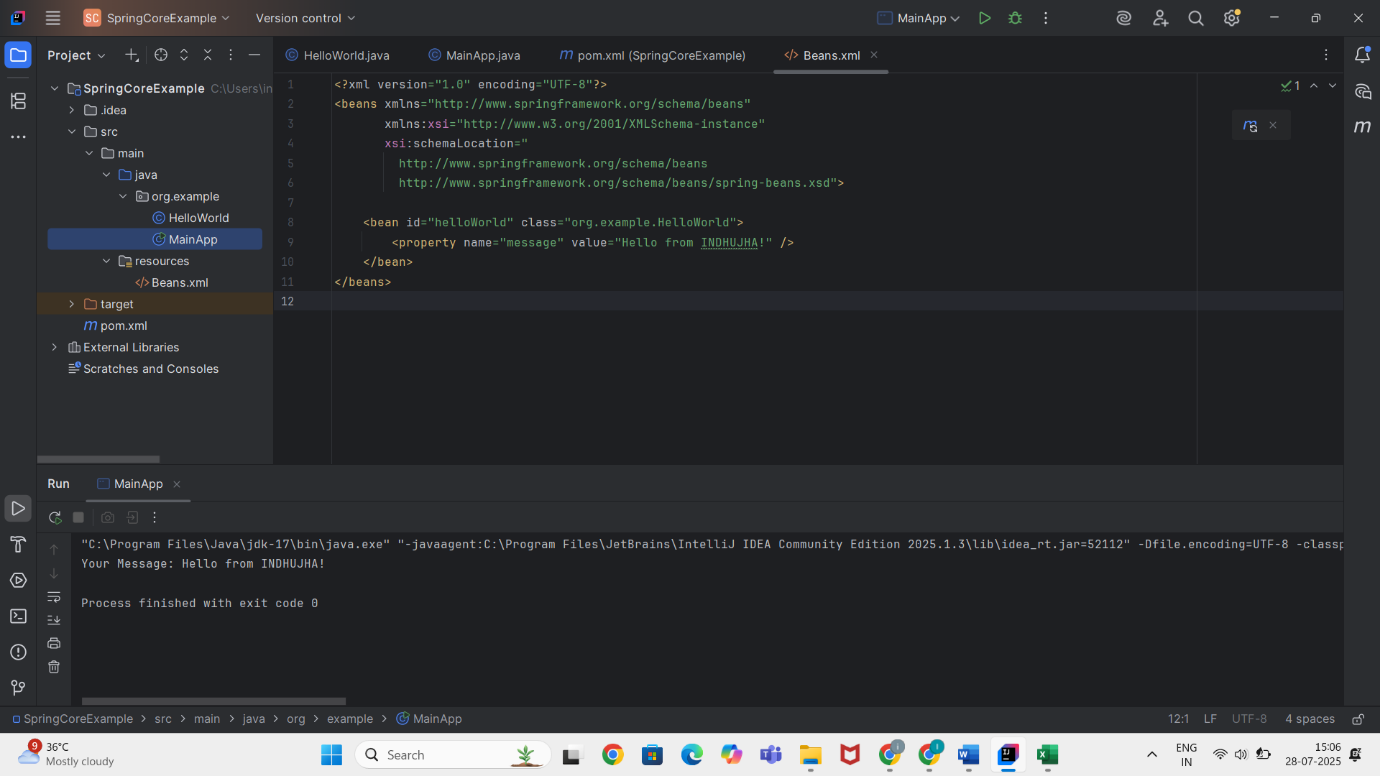
</resource>

</resources>

</build>

</project>

**OUTPUT:**

****

**Spring Data JPA - Quick Example**

**//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.example</groupId>

<artifactId>spring-data-jpa-example</artifactId>

<version>1.0.0</version>

<packaging>jar</packaging>

<name>Spring Data JPA Example</name>

<description>Spring Boot with Spring Data JPA and H2</description>

<parent>

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

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

<version>3.1.2</version>

<relativePath/>

</parent>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<!-- Spring Boot Starter Web -->

<dependency>

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

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

</dependency>

<!-- Spring Data JPA -->

<dependency>

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

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

</dependency>

<!-- H2 Database -->

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Spring Boot Maven Plugin -->

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**//User.java**

package com.example.springdatajpaexample;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.Id;

@Entity

public class User {

@Id

@GeneratedValue

private Long id;

private String name;

private String email;

public User() {}

public User(String name, String email) {

this.name = name;

this.email = email;

}

// Getters & Setters

public Long getId() { return id; }

public String getName() { return name; }

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

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

}

**//UserRepository**

package com.example.springdatajpaexample;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.Id;

@Entity

public class User {

@Id

@GeneratedValue

private Long id;

private String name;

private String email;

public User() {}

public User(String name, String email) {

this.name = name;

this.email = email;

}

// Getters & Setters

public Long getId() { return id; }

public String getName() { return name; }

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

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

}

**//DemoApplivation.java**

package com.example.springdatajpaexample;

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 DemoApplication implements CommandLineRunner {

@Autowired

private UserRepository repository;

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

@Override

public void run(String... args) {

repository.save(new User("Alice", "alice@example.com"));

repository.save(new User("Bob", "bob@example.com"));

System.out.println("All users:");

repository.findAll().forEach(user -> {

System.out.println(user.getId() + ": " + user.getName() + " - " + user.getEmail());

});

}

}