1. **spring-rest-handson**

**Hands on 1**

**Create a Spring Web Project using Maven**

**Code:**

SpringLearnApplication.java

**package** com.cognizant.springlearn;

**import** org.springframework.boot.SpringApplication;

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

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

@SpringBootApplication

**public** **class** SpringLearnApplication {

**private** **static** **final** Logger ***logger*** = LoggerFactory.*getLogger*(SpringLearnApplication.**class**);

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

***logger***.info("Starting Spring Learn Application...");

SpringApplication.*run*(SpringLearnApplication.**class**, args);

***logger***.info("Spring Learn Application started successfully!");

}

}

HelloController.java

**package** com.cognizant.springlearn.controller;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

@RestController

**public** **class** HelloController {

**private** **static** **final** Logger ***logger*** = LoggerFactory.*getLogger*(HelloController.**class**);

@GetMapping("/")

**public** String home() {

***logger***.info("Home endpoint accessed");

**return** "Welcome to Spring Learn Application!";

}

@GetMapping("/hello")

**public** String hello() {

***logger***.info("Hello endpoint accessed");

**return** "Hello from Spring Boot!";

}

}

Application.properties

# Application Configuration

spring.application.name=spring-learn

server.port=8080

# Logging Configuration

logging.level.com.cognizant.springlearn=INFO

logging.level.org.springframework.web=DEBUG

# DevTools Configuration

spring.devtools.restart.enabled=true

spring.devtools.livereload.enabled=true

pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<!-- Spring Boot Parent -->

<parent>

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

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

<version>3.2.0</version>

<relativePath/>

</parent>

<!-- Project Information -->

<groupId>com.cognizant</groupId>

<artifactId>spring-learn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-learn</name>

<description>Spring Learn Application</description>

<!-- Java Version -->

<properties>

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

</properties>

<!-- Dependencies -->

<dependencies>

<!-- Spring Boot Web Starter -->

<dependency>

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

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

</dependency>

<!-- Spring Boot DevTools -->

<dependency>

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

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

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Spring Boot Test Starter -->

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<!-- Build Configuration -->

<build>

<plugins>

<plugin>

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

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

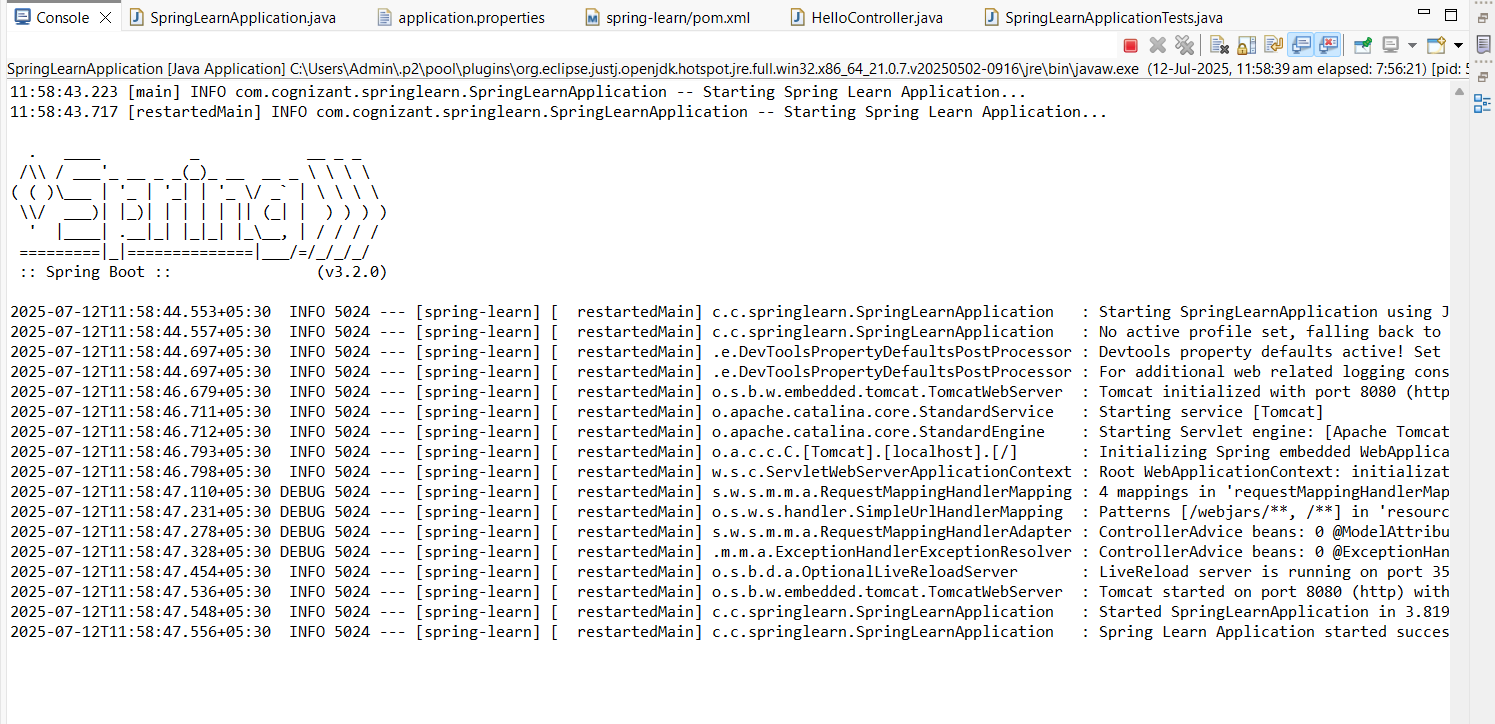
</plugin>

</plugins>

</build>

</project>

**Output:**



**Hands On 4**

**Spring Core – Load Country from Spring Configuration XML**

SpringCoreTutorialApplication.java

**package** com.cognizant.springcore;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**import** org.springframework.context.ApplicationContext;

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

**public** **class** SpringCoreTutorialApplication {

**private** **static** **final** Logger ***LOGGER*** = LoggerFactory.*getLogger*(SpringCoreTutorialApplication.**class**);

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

***LOGGER***.debug("Starting SpringCoreTutorialApplication");

// Invoke displayCountry method

*displayCountry*();

***LOGGER***.debug("SpringCoreTutorialApplication completed");

}

/\*\*

\* Method to display country details by reading from Spring configuration

\*/

**public** **static** **void** displayCountry() {

***LOGGER***.debug("Inside displayCountry() method");

**try** {

// Create ApplicationContext by loading country.xml from classpath

ApplicationContext context = **new** ClassPathXmlApplicationContext("country.xml");

// Get the country bean from the context

Country country = (Country) context.getBean("country", Country.**class**);

// Display country details

***LOGGER***.debug("Country : {}", country.toString());

} **catch** (Exception e) {

***LOGGER***.error("Error while loading country configuration: {}", e.getMessage(), e);

}

}

Country.java

**package** com.cognizant.springcore;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**public** **class** Country {

**private** **static** **final** Logger ***LOGGER*** = LoggerFactory.*getLogger*(Country.**class**);

**private** String code;

**private** String name;

/\*\*

\* Empty parameter constructor

\*/

**public** Country() {

***LOGGER***.debug("Inside Country Constructor.");

}

/\*\*

\* Getter for code

\* **@return** String code

\*/

**public** String getCode() {

***LOGGER***.debug("Inside getCode() method");

**return** code;

}

/\*\*

\* Setter for code

\* **@param** code String

\*/

**public** **void** setCode(String code) {

***LOGGER***.debug("Inside setCode() method with value: {}", code);

**this**.code = code;

}

/\*\*

\* Getter for name

\* **@return** String name

\*/

**public** String getName() {

***LOGGER***.debug("Inside getName() method");

**return** name;

}

/\*\*

\* Setter for name

\* **@param** name String

\*/

**public** **void** setName(String name) {

***LOGGER***.debug("Inside setName() method with value: {}", name);

**this**.name = name;

}

/\*\*

\* toString method

\* **@return** String representation of Country object

\*/

@Override

**public** String toString() {

**return** "Country{" +

"code='" + code + '\'' +

", name='" + name + '\'' +

'}';

}

}

Country.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">

<!-- Country bean configuration -->

<bean id="country" class="com.cognizant.springcore.Country">

<property name="code" value="IN" />

<property name="name" value="India" />

</bean>

</beans>

**Output:**

