**Hands on 1**

**Create a Spring Web Project using Maven**   
  
SOLUTION:

package com.example.springlearn.model;

public class Country {

private String code;

private String name;

public Country() {

}

public Country(String code, String name) {

this.code = code;

this.name = 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;

}

}

COUNTRYCONTROLLER CLASS:

package com.example.springlearn.controller;

import com.example.springlearn.model.Country;

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

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

@RestController

public class CountryController {

@GetMapping("/country")

public Country getCountry() {

Country country = new Country();

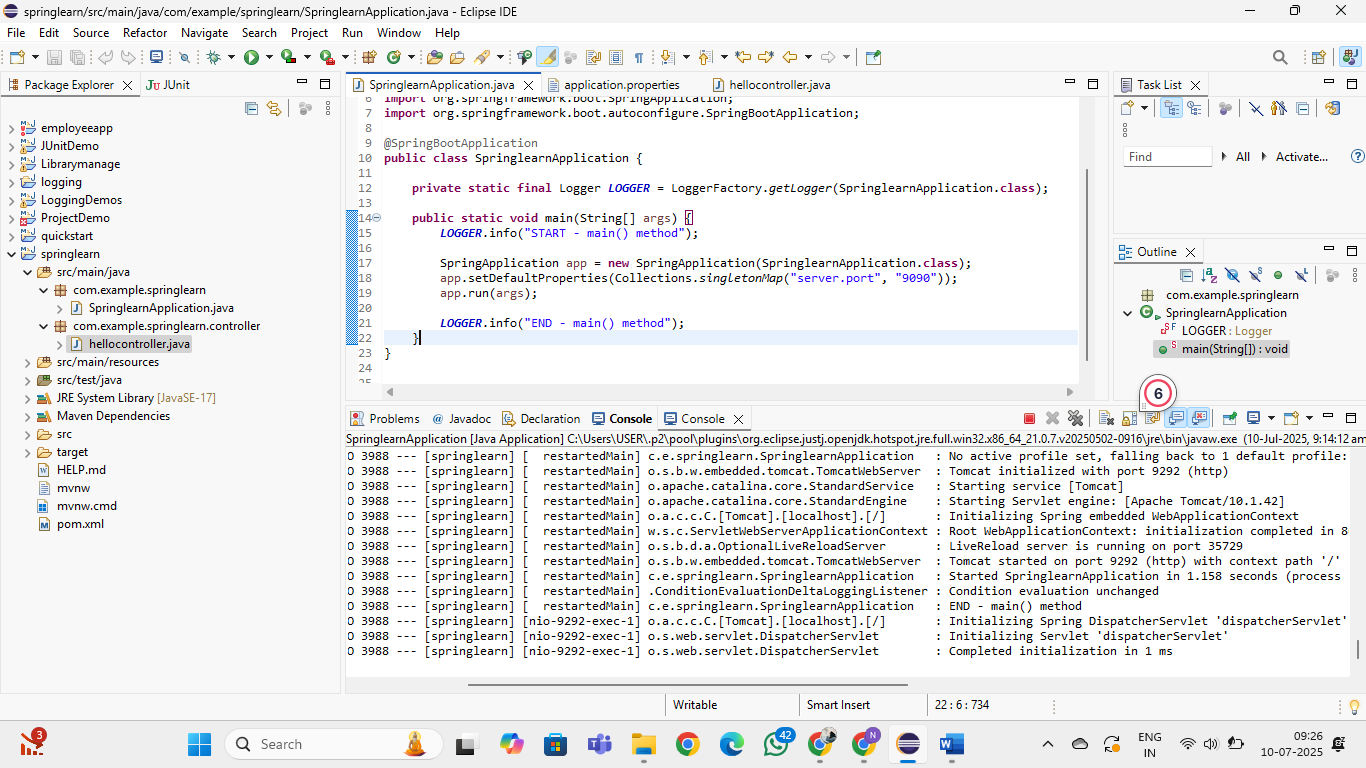
country.setCode("IN");

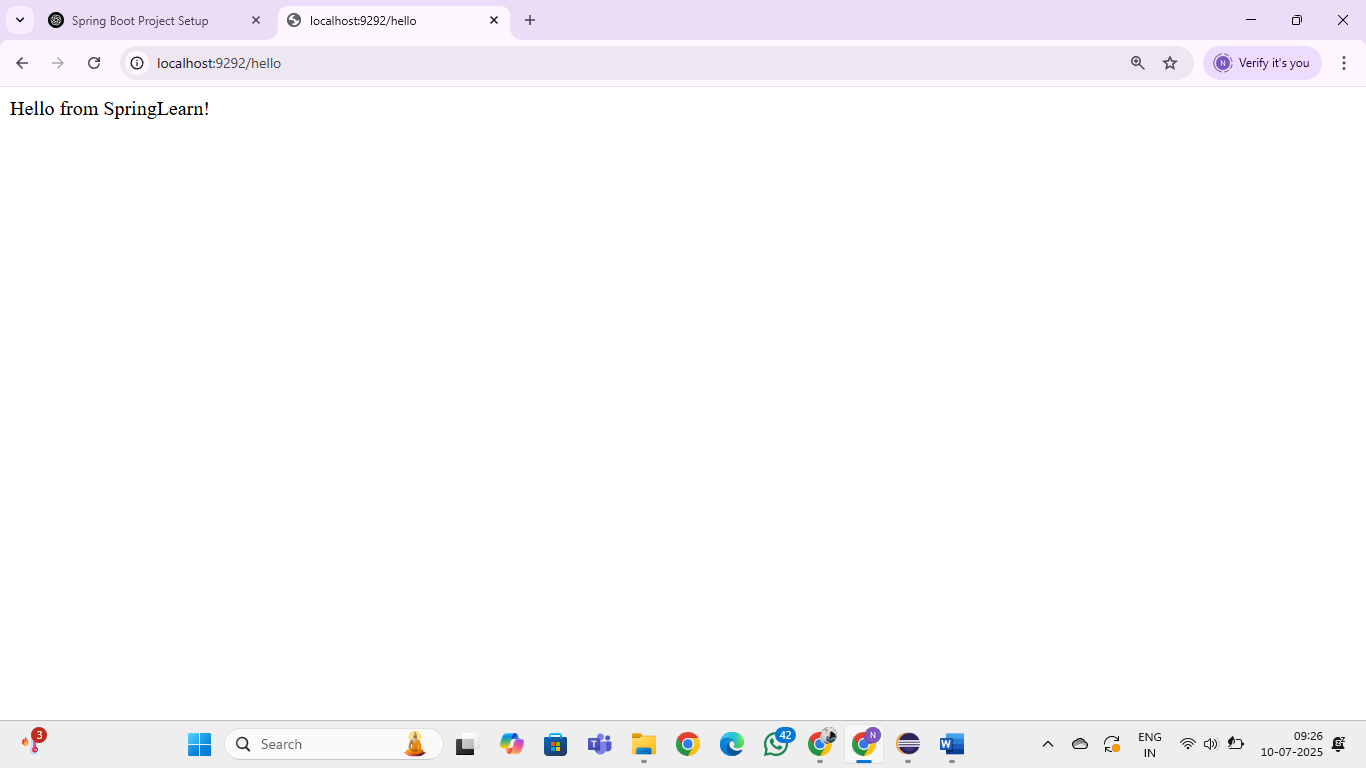
country.setName("India");

return country;

}

}





**Hands on 4**

**Spring Core – Load Country from Spring Configuration XML**   
  
An airlines website is going to support booking on four countries. There will be a drop down on the home page of this website to select the respective country. It is also important to store the two-character ISO code of each country. 

|  |  |
| --- | --- |
| **Code** | **Name** |
| US | United States |
| DE | Germany |
| IN | India |
| JP | Japan |

Above data has to be stored in spring configuration file. Write a program to read this configuration file and display the details.  
  
SOLUTION:

SpringLearnApplication.java:

**package** com.example.springlearn;

**import** java.util.Collections;

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

**import** com.example.springlearn.model.country;

@SpringBootApplication

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

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

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

***LOGGER***.info("START - main() method");

SpringApplication app = **new** SpringApplication(SpringLearnApplication.**class**);

app.setDefaultProperties(Collections.*singletonMap*("server.port", "9090"));

app.run(args);

*displayCountry*();

***LOGGER***.info("END - main() method");

}

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

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

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

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

}

}

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">

<bean id="country" class="com.example.springlearn.model.country">

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

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

</bean>

</beans>

**Country.java:**

**package** com.example.springlearn.model;

**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;

**public** country() {

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

}

**public** String getCode() {

***LOGGER***.debug("Getter for code called");

**return** code;

}

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

***LOGGER***.debug("Setter for code called");

**this**.code = code;

}

**public** String getName() {

***LOGGER***.debug("Getter for name called");

**return** name;

}

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

***LOGGER***.debug("Setter for name called");

**this**.name = name;

}

@Override

**public** String toString() {

**return** "Country [code=" + code + ", name=" + name + "]";

}

}

OUTPUT:  
  
