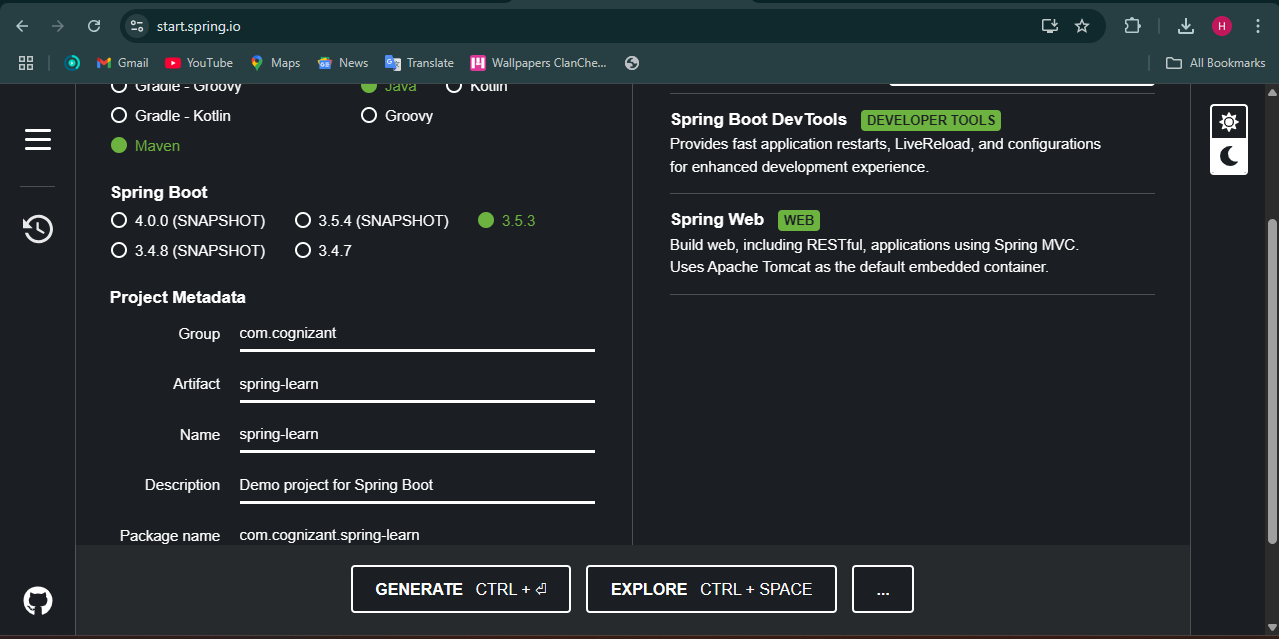
Spring Rest Handson and JWT OUTPUTS

Name : G. Sri Ratna Hemanth

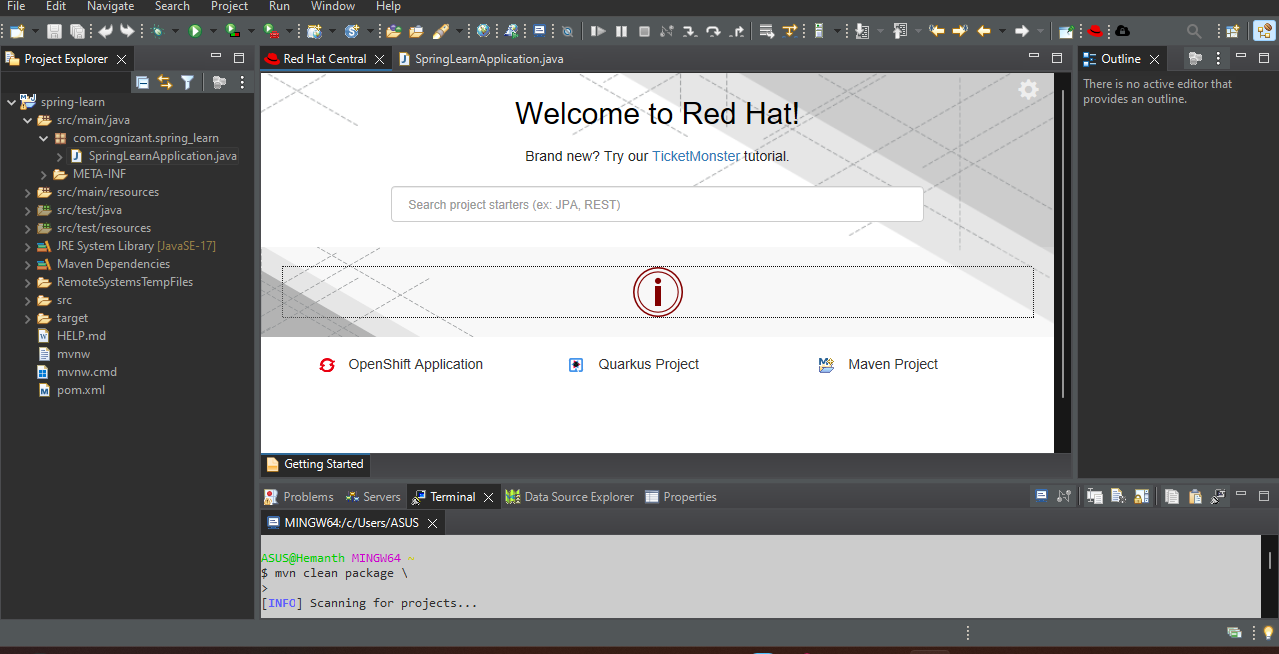
Superset Id : 6124988

1. Create a Spring Web Project using Maven

**Step – 1 :** Go to Spring Initializr and create a java project and select dependencies.



**Step – 2 :** Extract the folder and import the file in eclipse.



**Step – 3 :** Open SpringLearnApplication.java and write the code.

Code :

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class SpringLearnApplication {

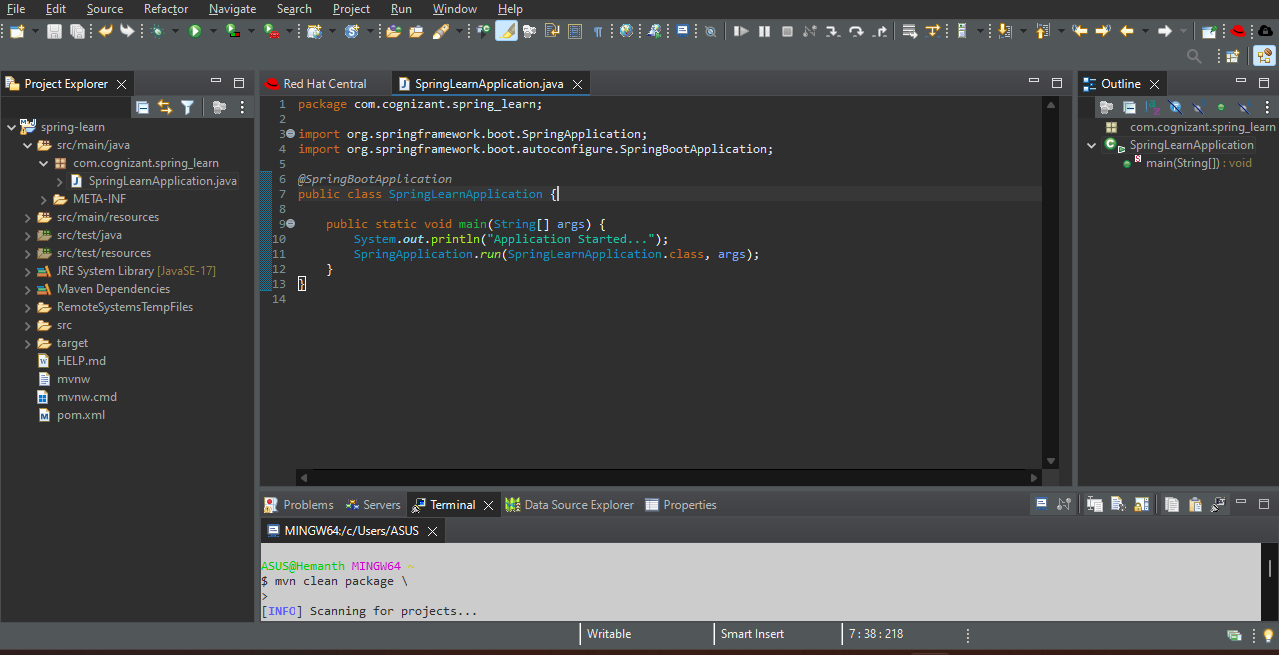
public static void main(String[] args) {

System.***out***.println("Application Started...");

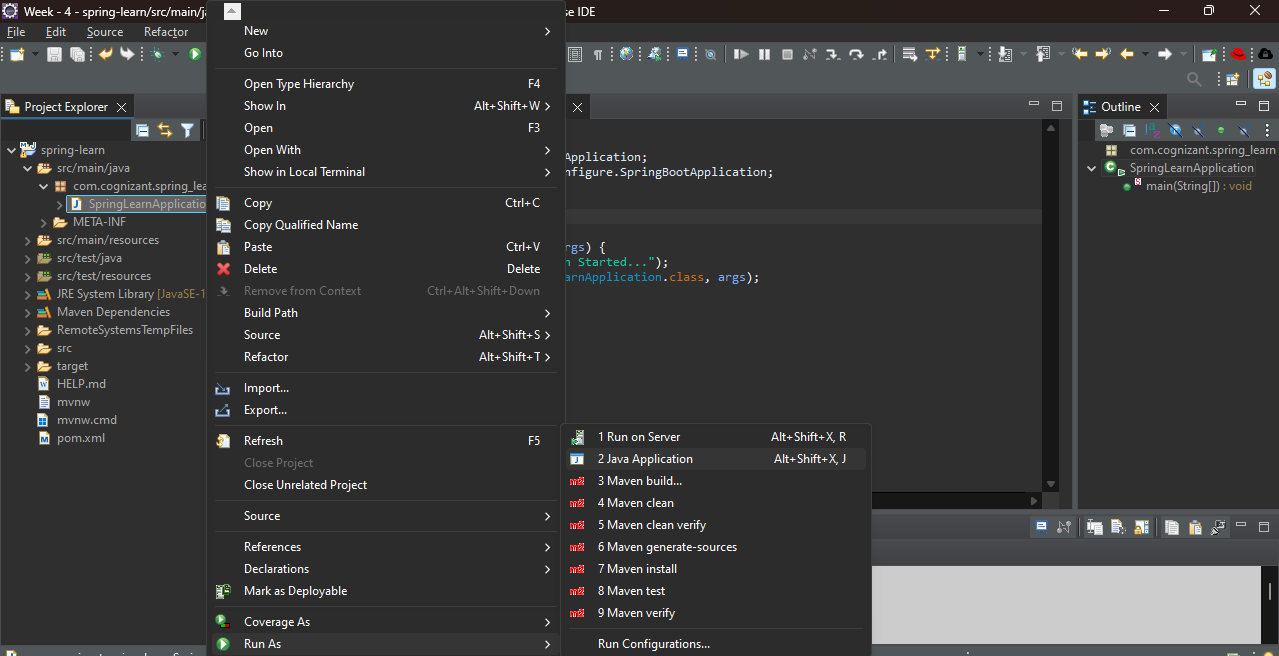
SpringApplication.*run*(SpringLearnApplication.class, args);

}

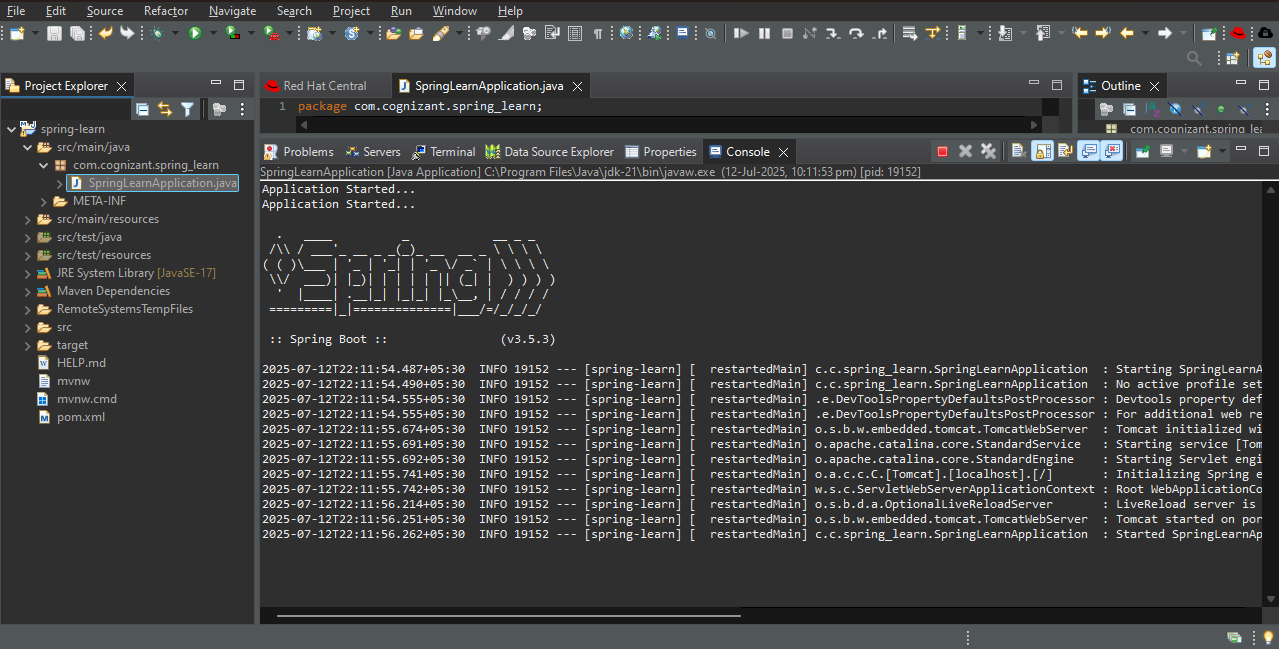
}



**Step – 4 :** Run the project as Java application.

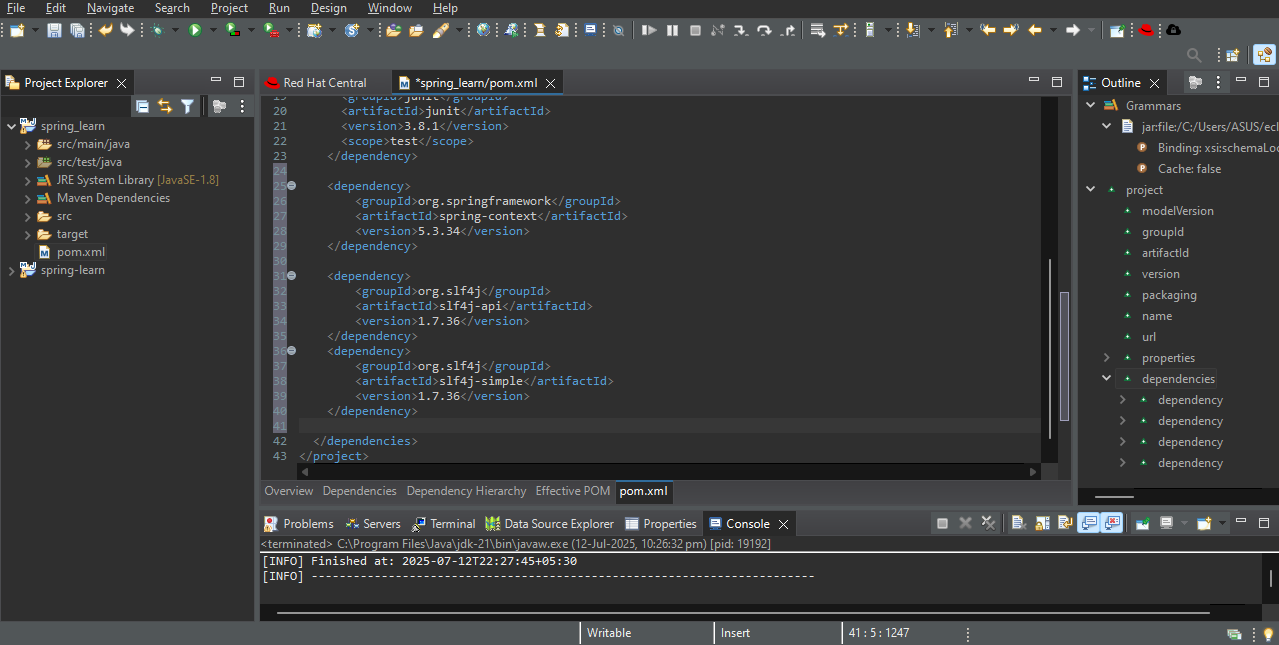


OUTPUT :



1. Spring Core – Load Country from Spring Configuration XML

**Step – 1 :** Create a maven project and add spring dependency to it in pom.xml



**Step – 2 :** Create a new package in src/main/java and create a class Country.java and write code.

**Code :**

package com.cognizant.springlearn;

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("Inside getCode()");

return code;

}

public void setCode(String code) {

***LOGGER***.debug("Inside setCode()");

this.code = code;

}

public String getName() {

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

return name;

}

public void setName(String name) {

***LOGGER***.debug("Inside setName()");

this.name = name;

}

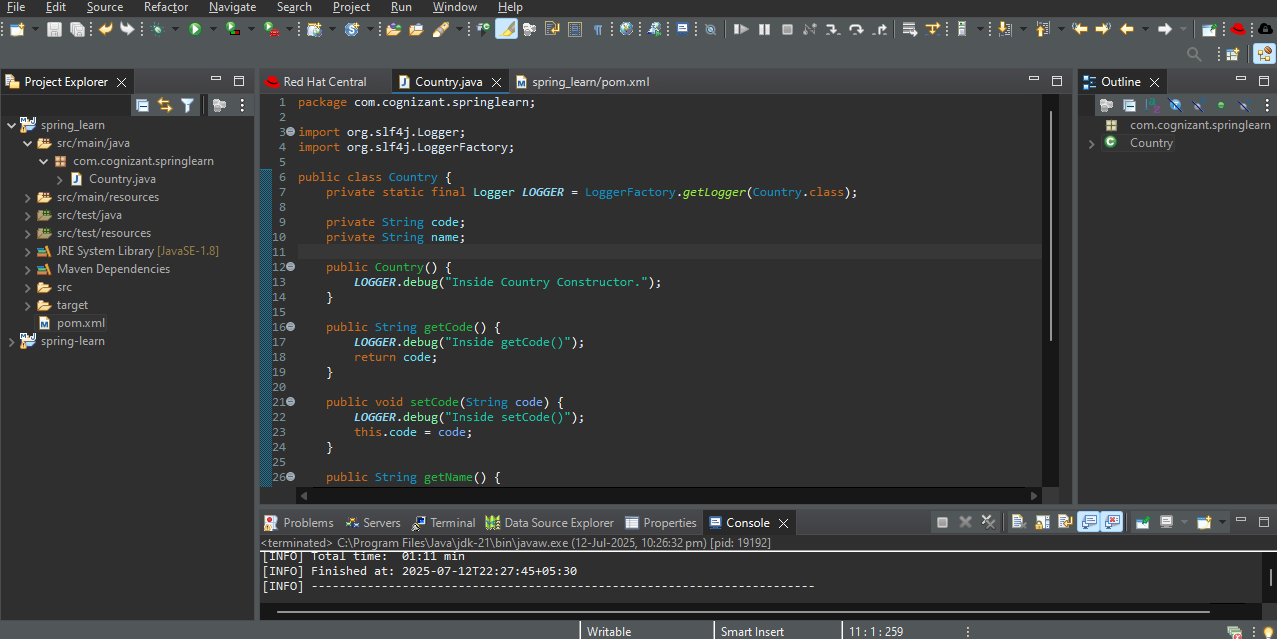
*@Override*

public String toString() {

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

}

}



**Step – 3 :** Create country.xml in src/main/resources and write the code.

**Code :**

<?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">

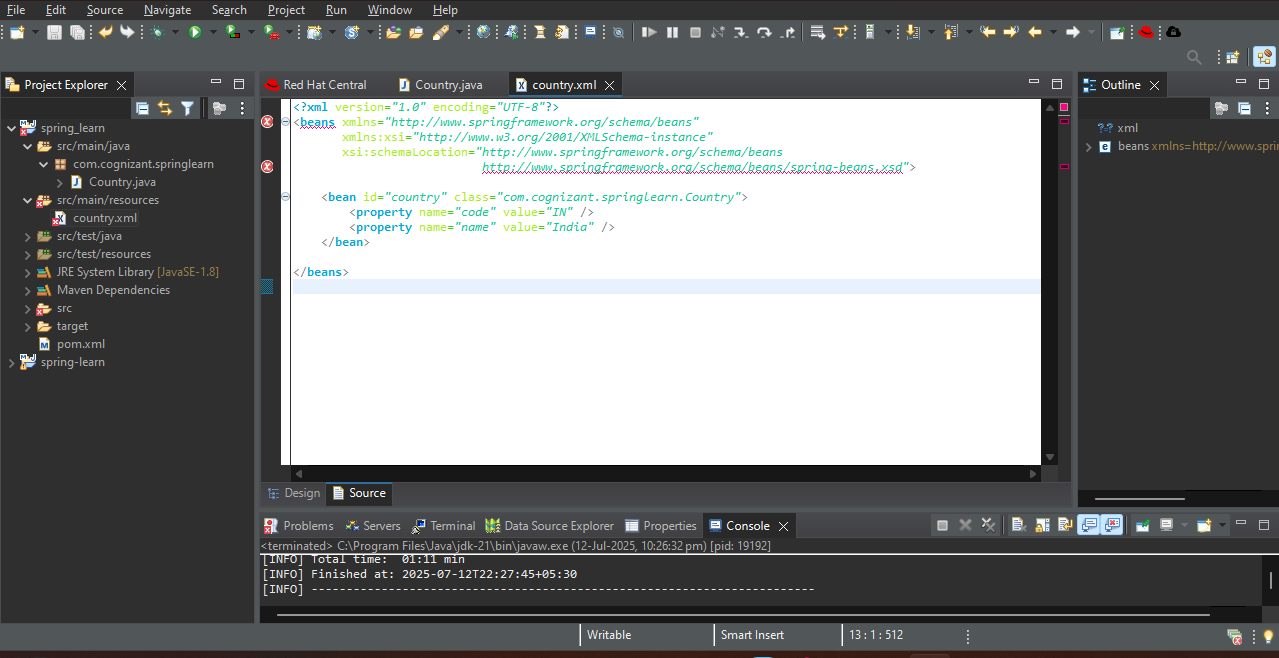
<bean id="country" class="com.cognizant.springlearn.Country">

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

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

</bean>

</beans>



**Step – 4 :** Create main file SpringLearnApplication.java and write code.

**Code :**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpringLearnApplication {

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

public static void main(String[] args) {

displayCountry();

}

public static void displayCountry() {

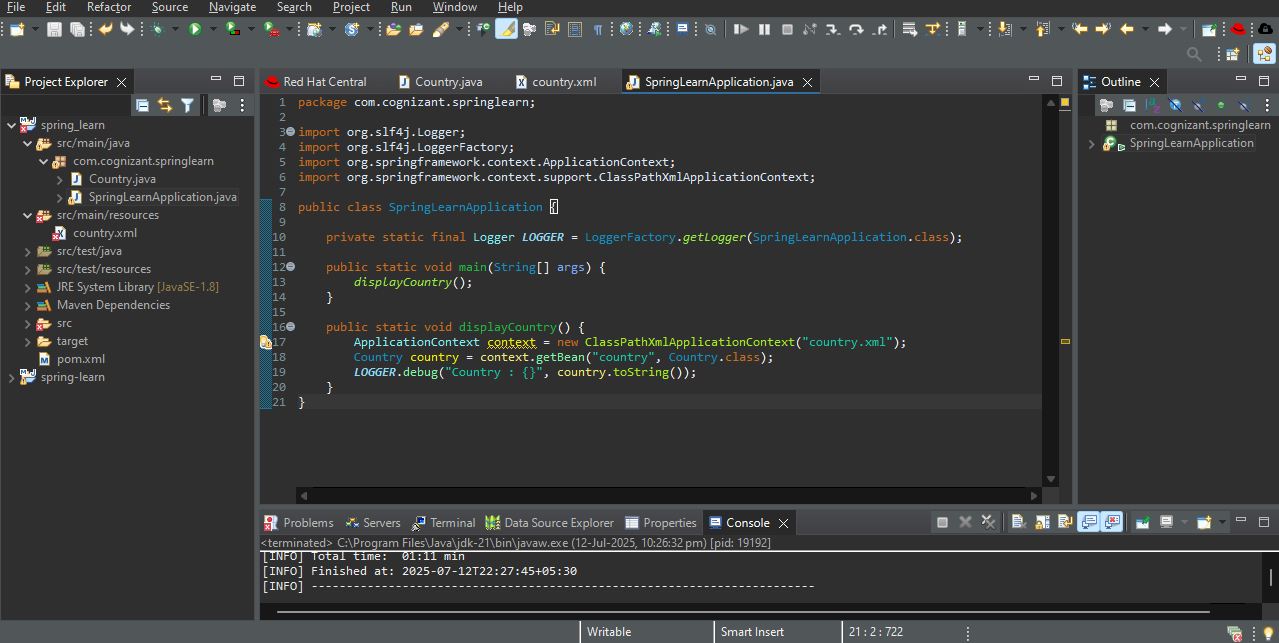
ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");

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

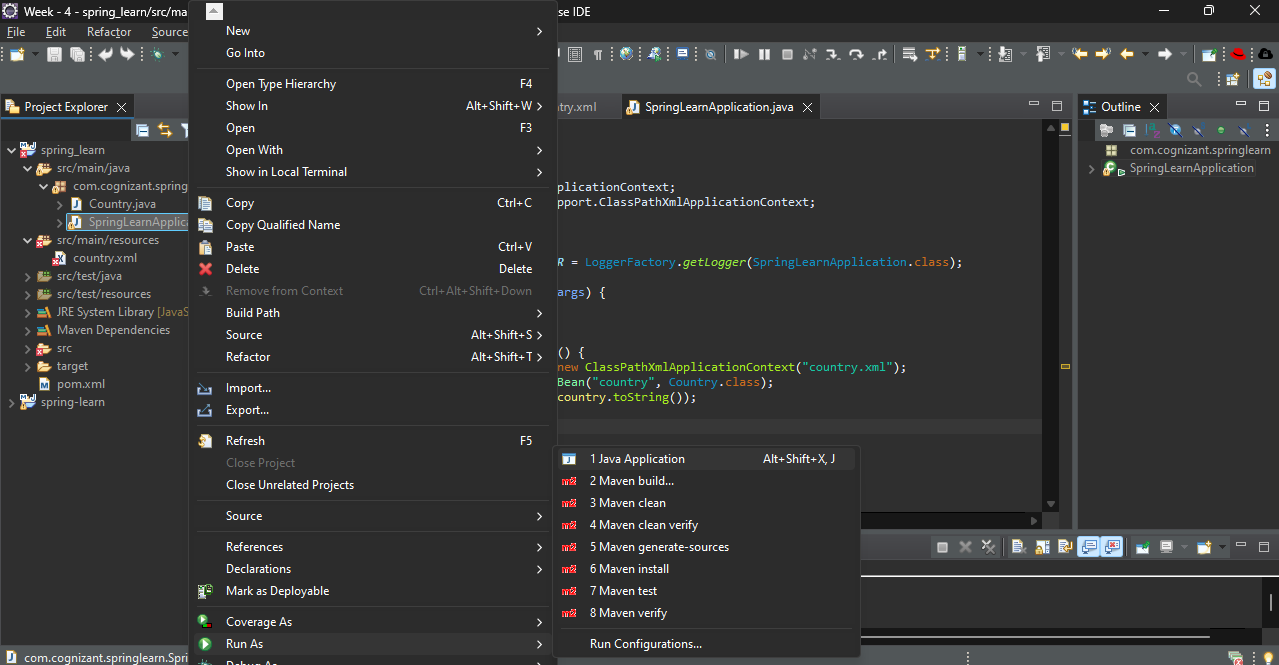
LOGGER.debug("Country : {}", country.toString());

}

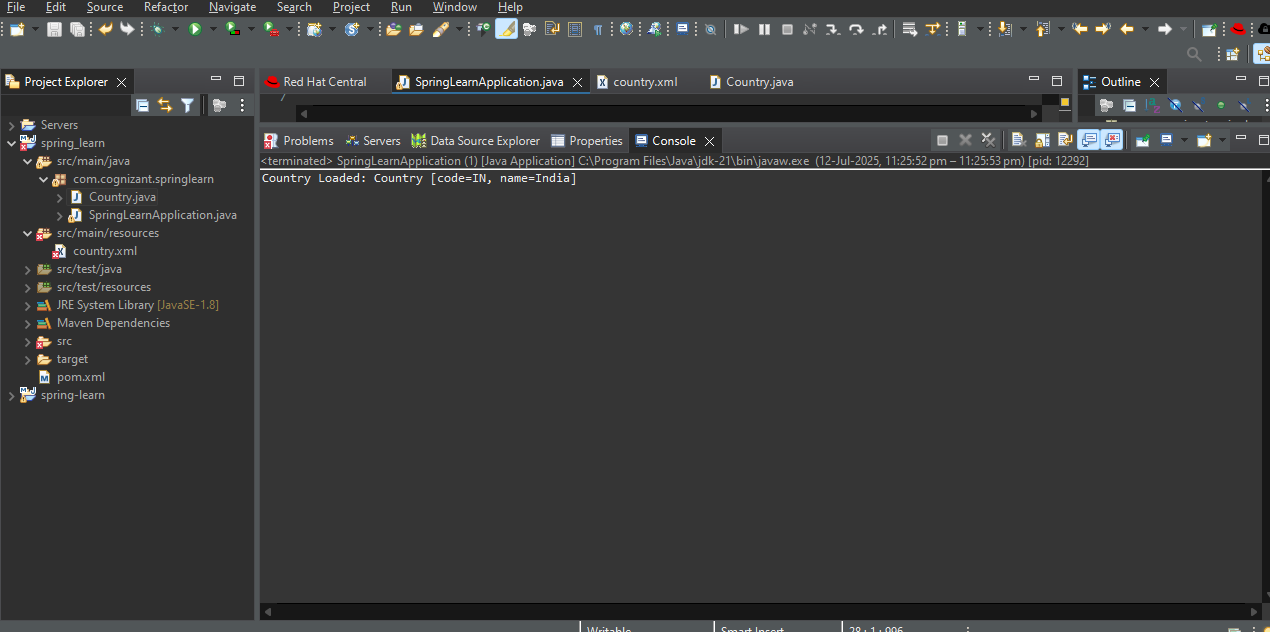
}



**Step – 5 :** Run the main class as java application.

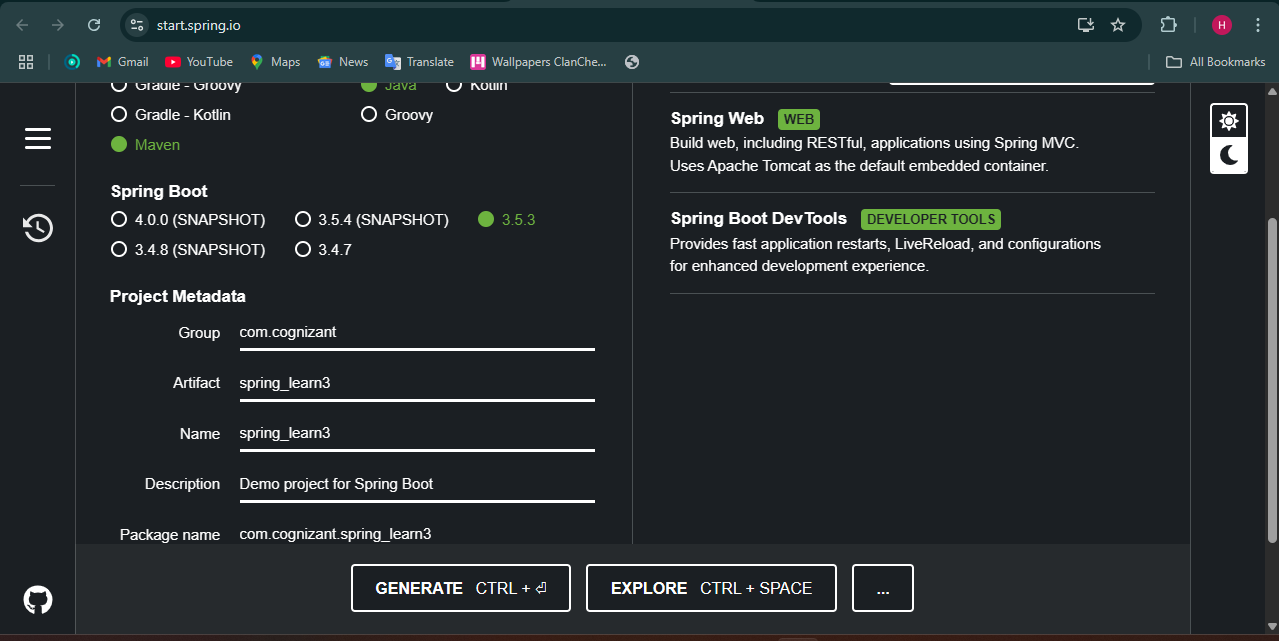


OUTPUT :



1. Hello World RESTful Web Service

**Step – 1 :** Go to spring initializr.io create a spring boot project.



**Step – 2 :** Extract the folder and import it in Eclipse.



**Step – 3 :** Create a class HelloController.java and write the code.

**Code :**

package com.cognizant.spring\_learn3.controller;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class HelloController {

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

@GetMapping("/hello")

public String sayHello() {

LOGGER.info("START - sayHello()");

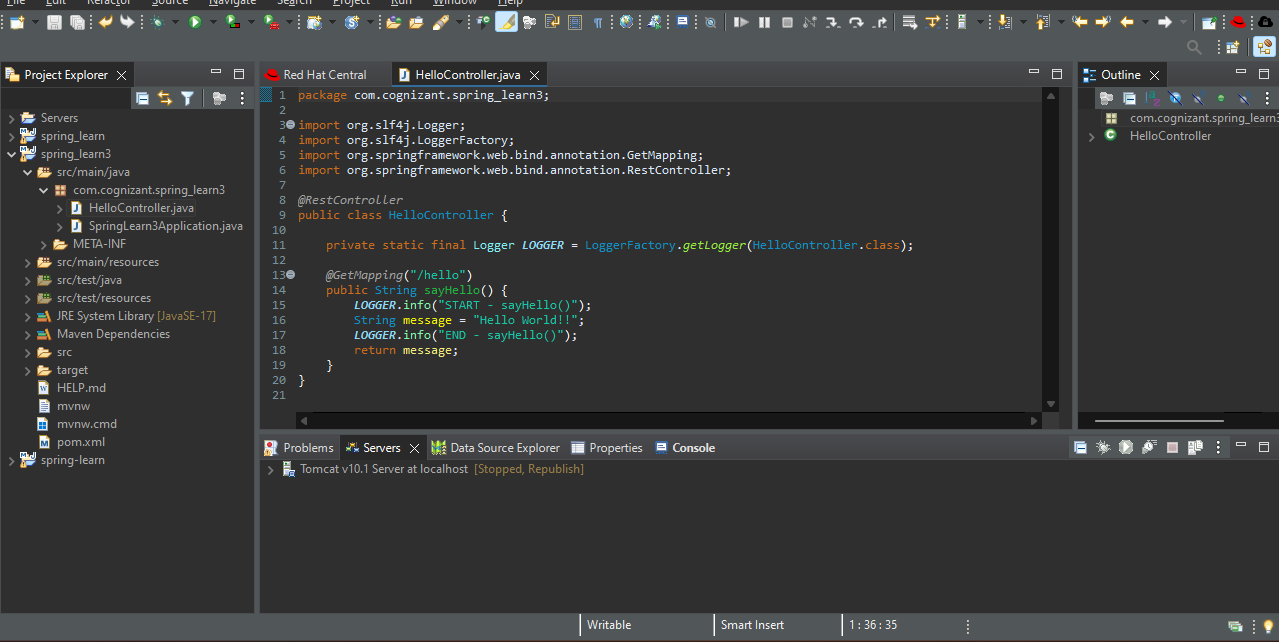
String message = "Hello World!!";

LOGGER.info("END - sayHello()");

return message;

}

}



**Step – 4 :** Write the main code in SpringLearn3Application.java.

**Code :**

package com.cognizant.spring\_learn3;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

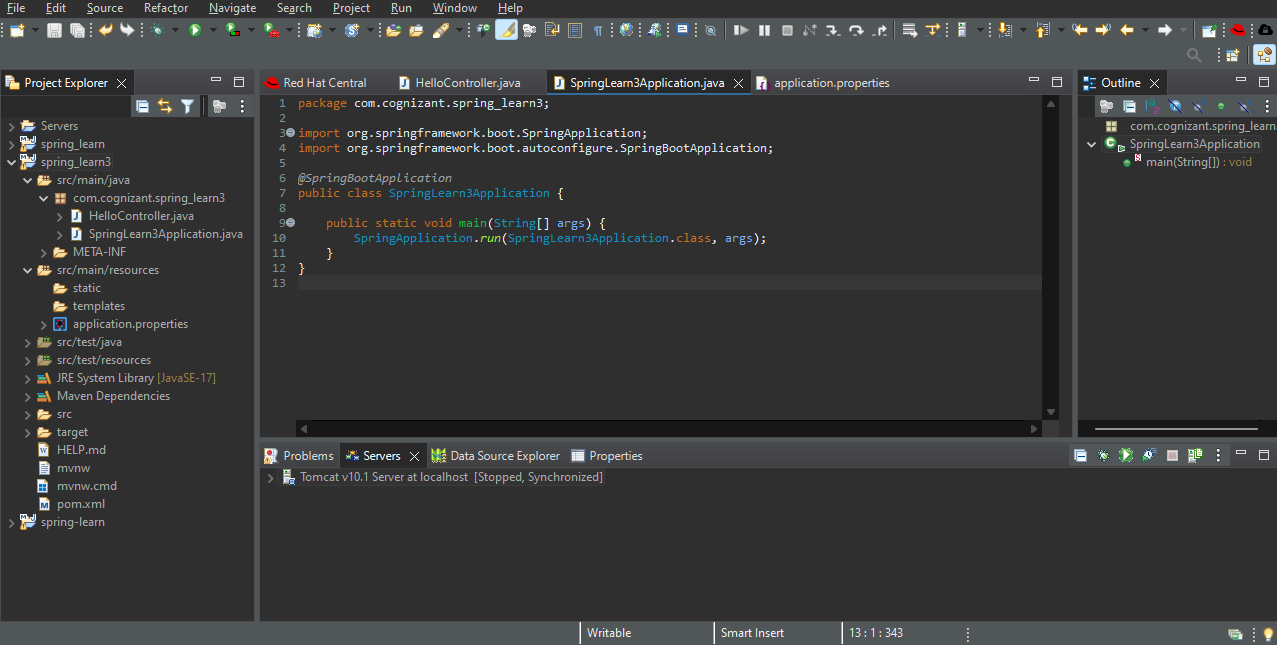
public class SpringLearn3Application {

public static void main(String[] args) {

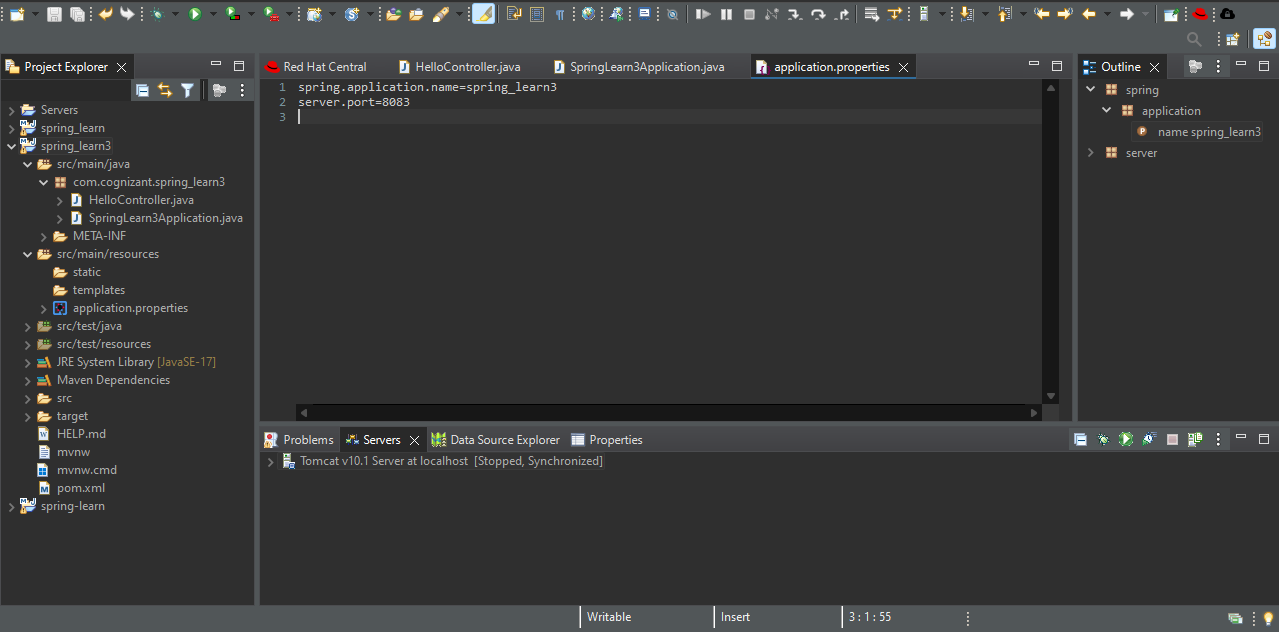
SpringApplication.run(SpringLearn3Application.class, args);

}

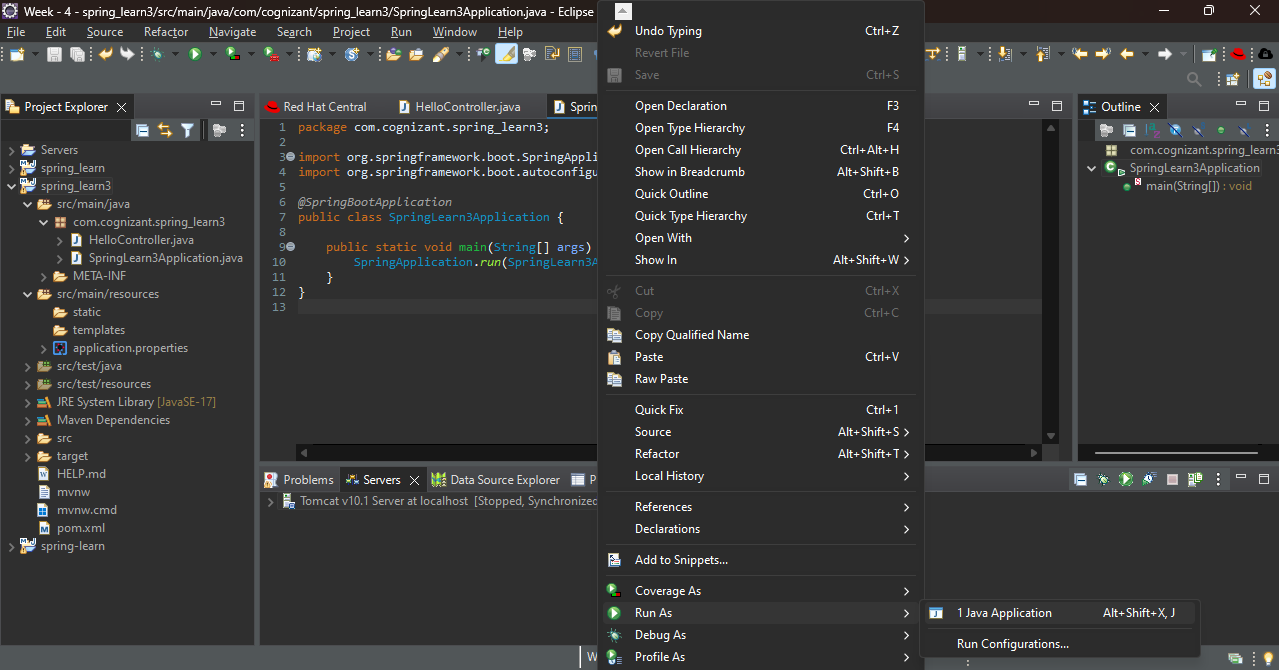
}

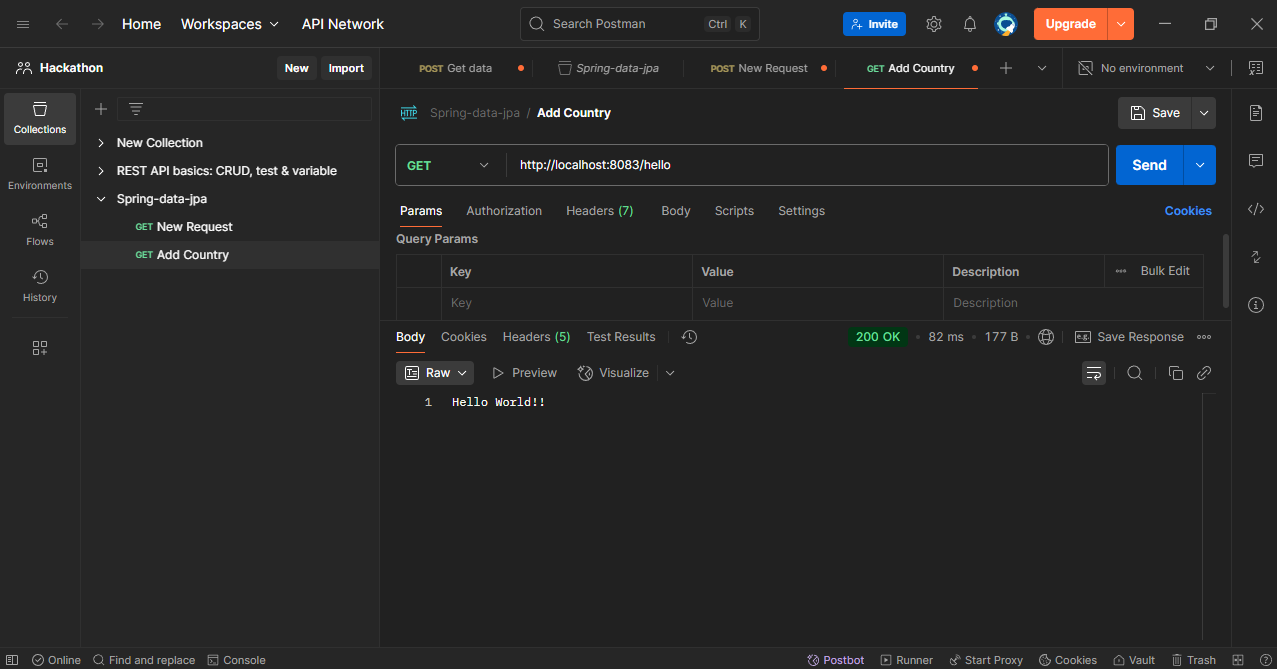


**Step – 5 :** Go to rec/main/resources 🡪 application.properties, Enter your tomcat port number.

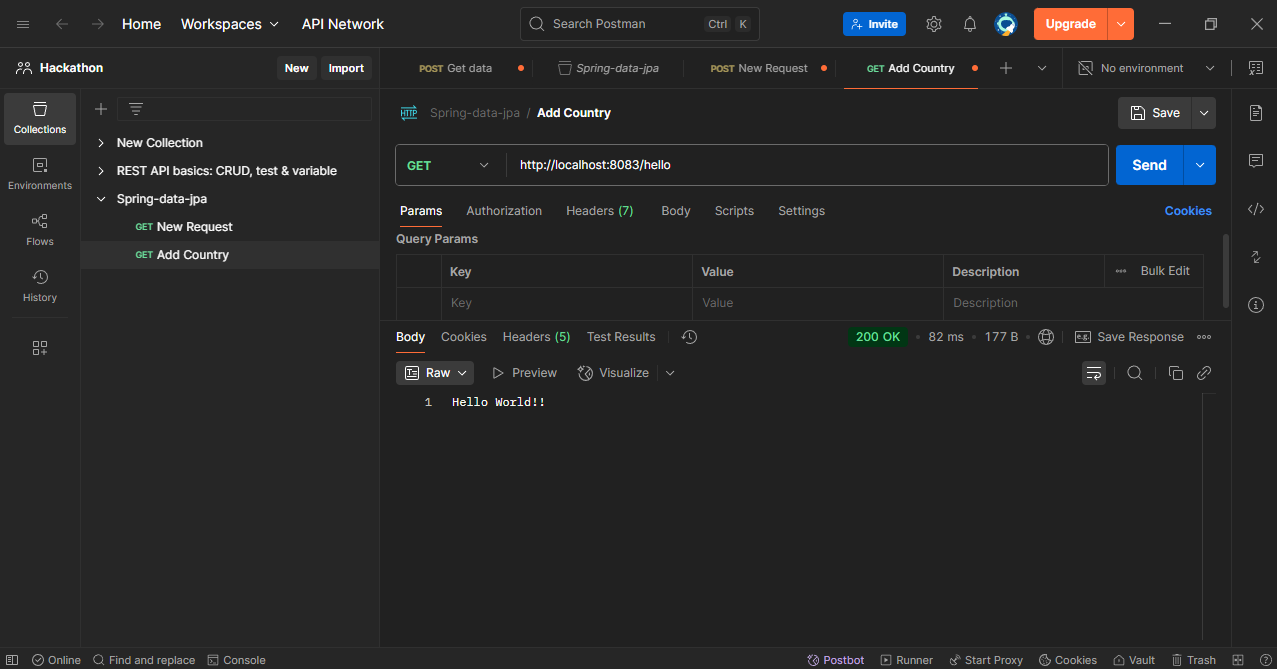


**Step – 6 :** Run the main application as java application.

**Step – 7 :** Go to Postman and enter <http://localhost:8083/hello>.

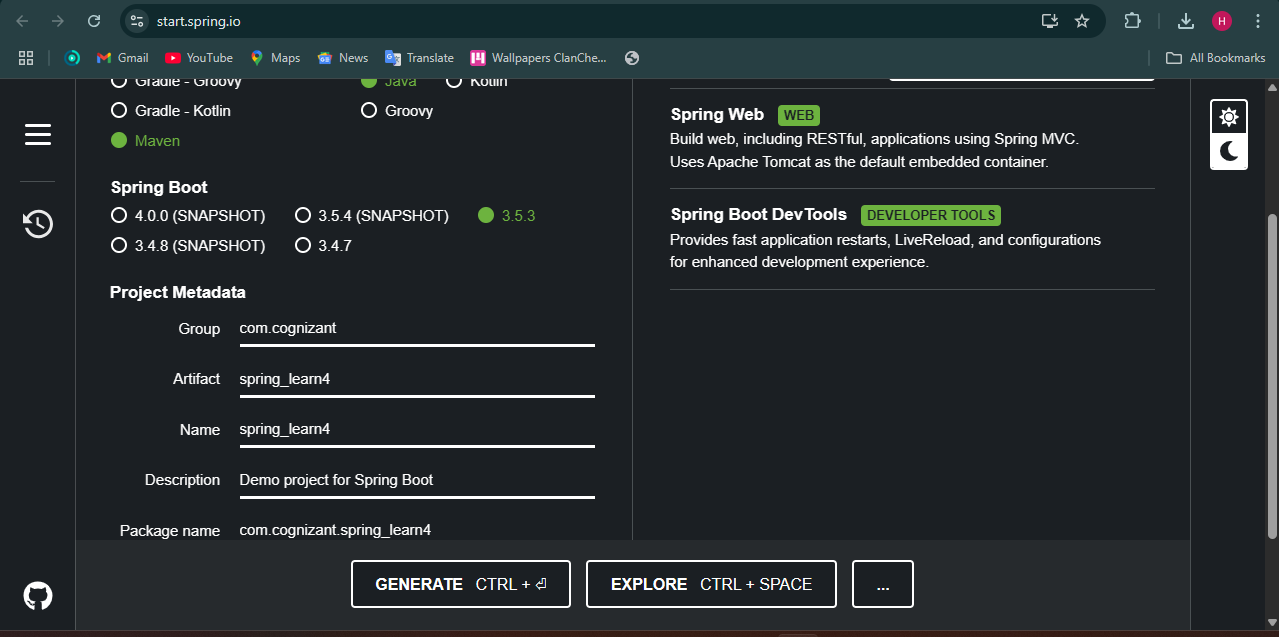
****

OUTPUT :

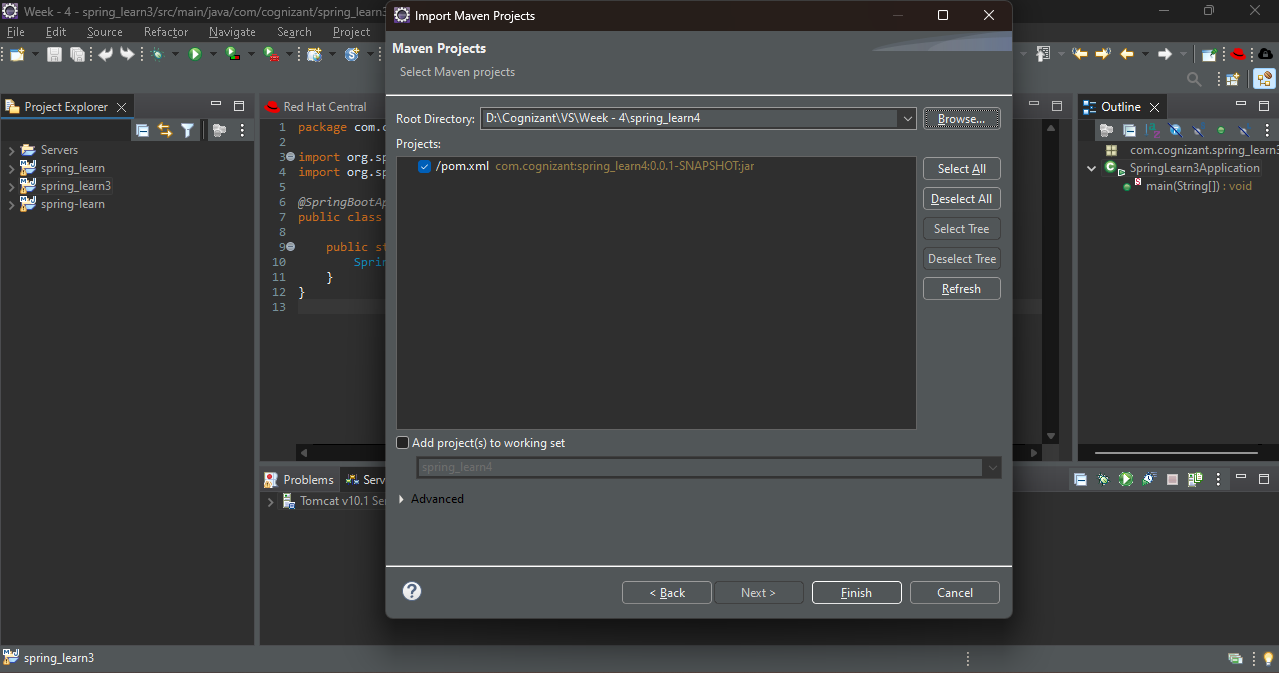


1. REST - Country Web Service

**Step – 1 :** Go to spring initializr.io create a spring boot project.



**Step – 2 :** Extract the folder and import it in Eclipse.



**Step – 3 :** Create a new package model and create a class Country.java.

**Code :**

package com.cognizant.spring\_learn4.model;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor");

}

public String getCode() {

System.out.println("Inside getCode()");

return code;

}

public void setCode(String code) {

System.out.println("Inside setCode()");

this.code = code;

}

public String getName() {

System.out.println("Inside getName()");

return name;

}

public void setName(String name) {

System.out.println("Inside setName()");

this.name = name;

}

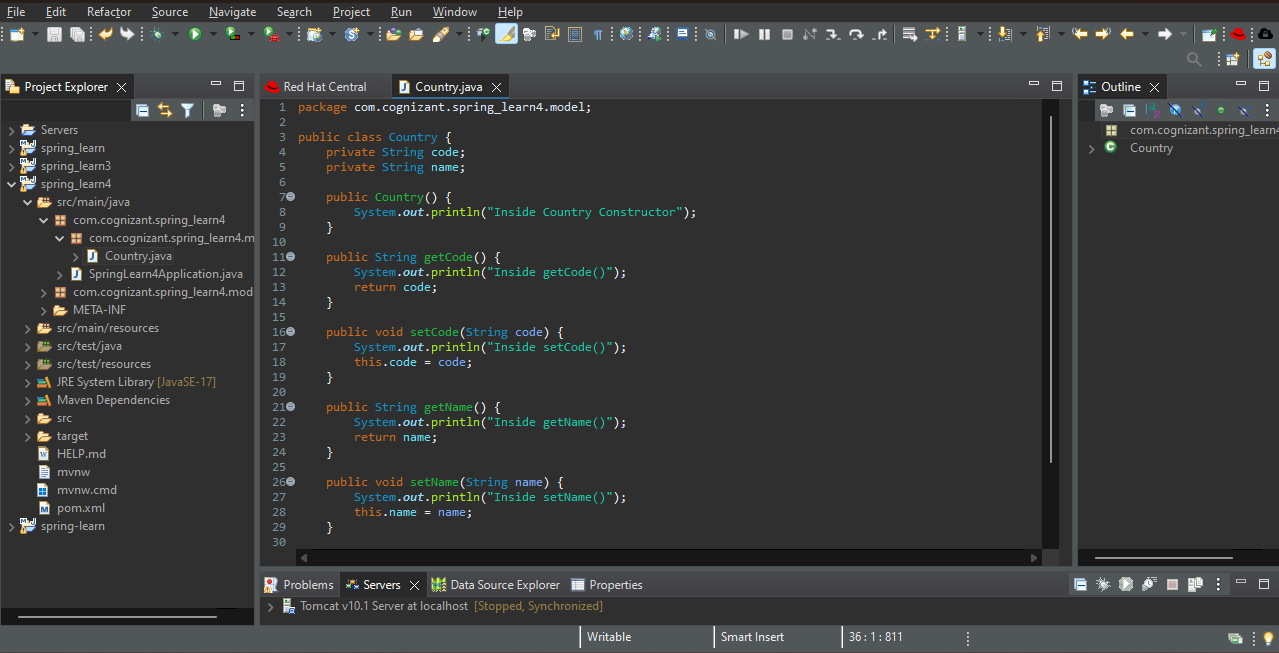
@Override

public String toString() {

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

}

}



**Step – 4 :** In src/main/resources file add new file country.xml.

**Code :**

<?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">

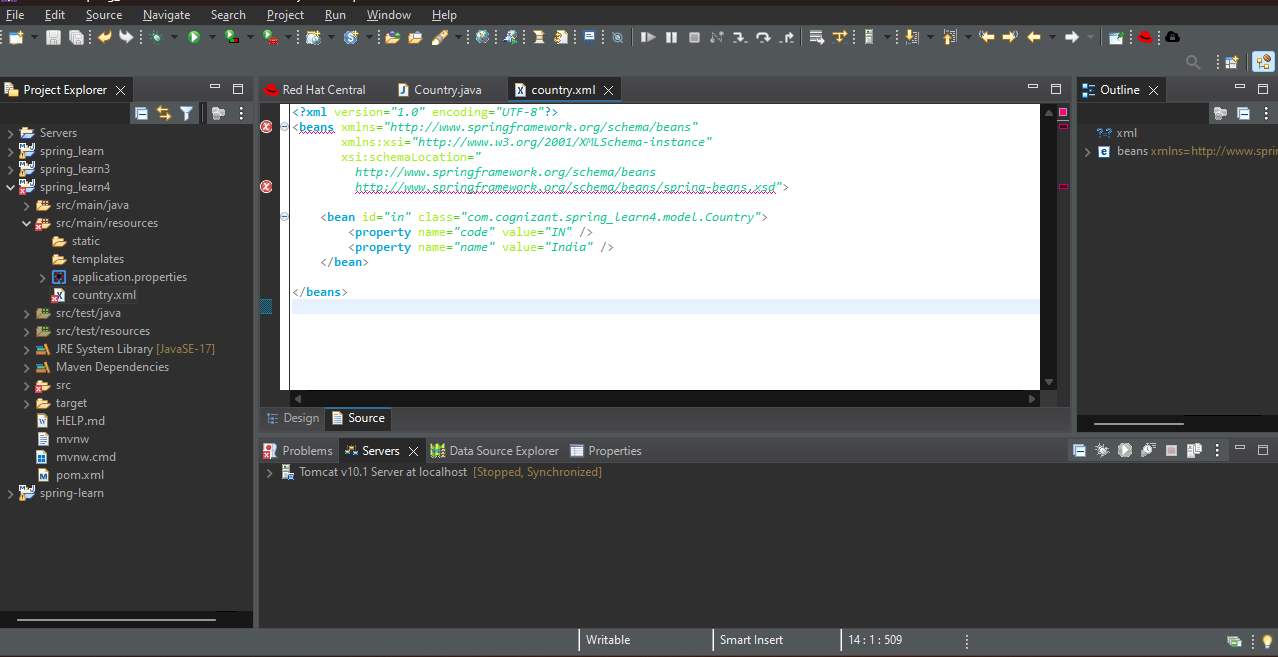
<bean id="in" class="com.cognizant.spring\_learn4.model.Country">

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

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

</bean>

</beans>



**Step – 5 :** Create a package Controller and create a class CountryController.java.

**Code :**

package com.cognizant.spring\_learn4.controller;

import com.cognizant.spring\_learn4.model.Country;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

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

@RestController

public class CountryController {

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

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START getCountryIndia()");

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

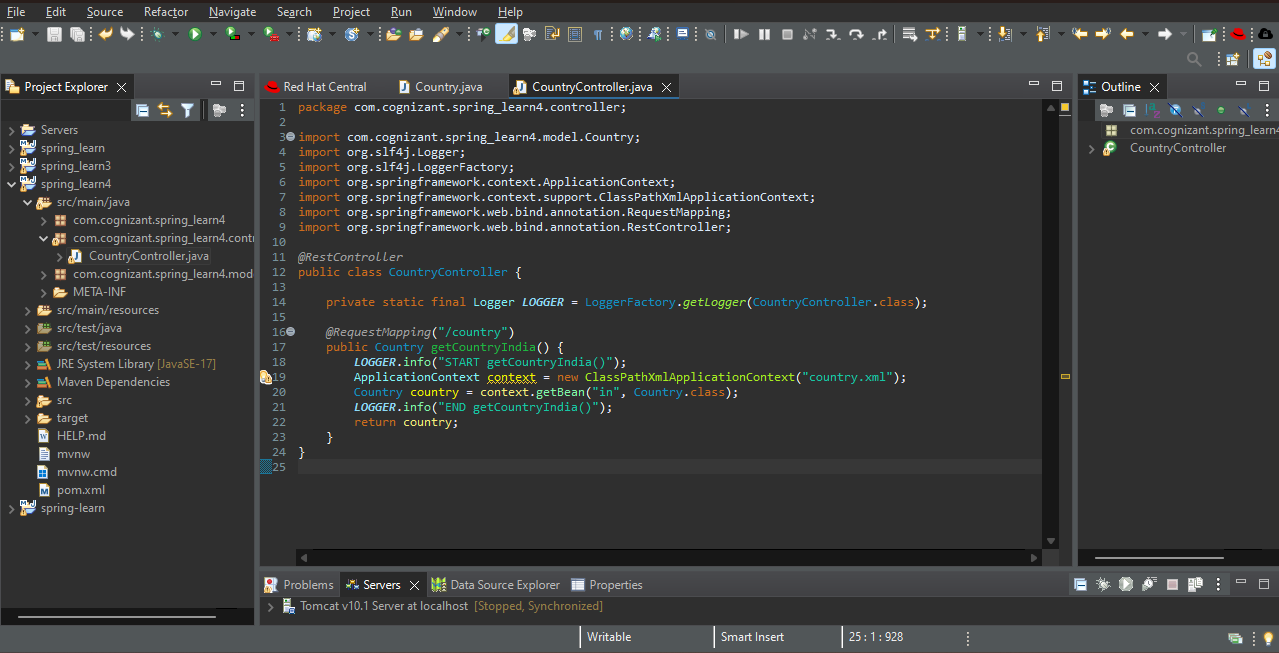
Country country = context.getBean("in", Country.class);

LOGGER.info("END getCountryIndia()");

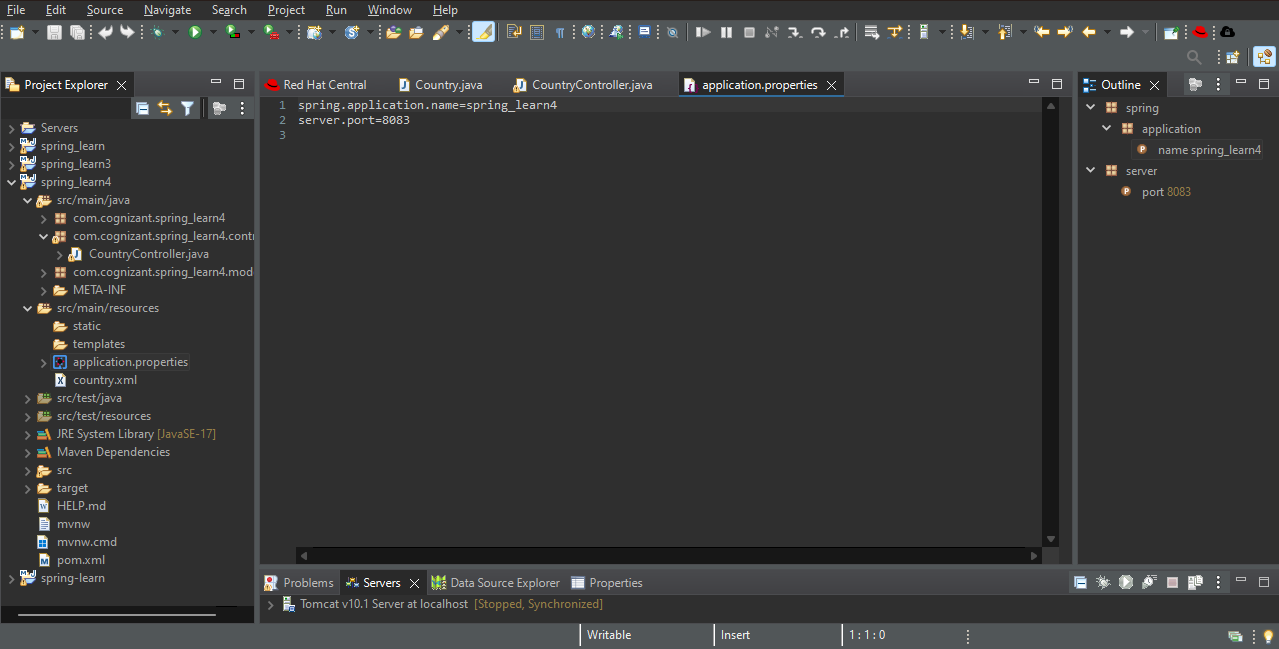
return country;

}

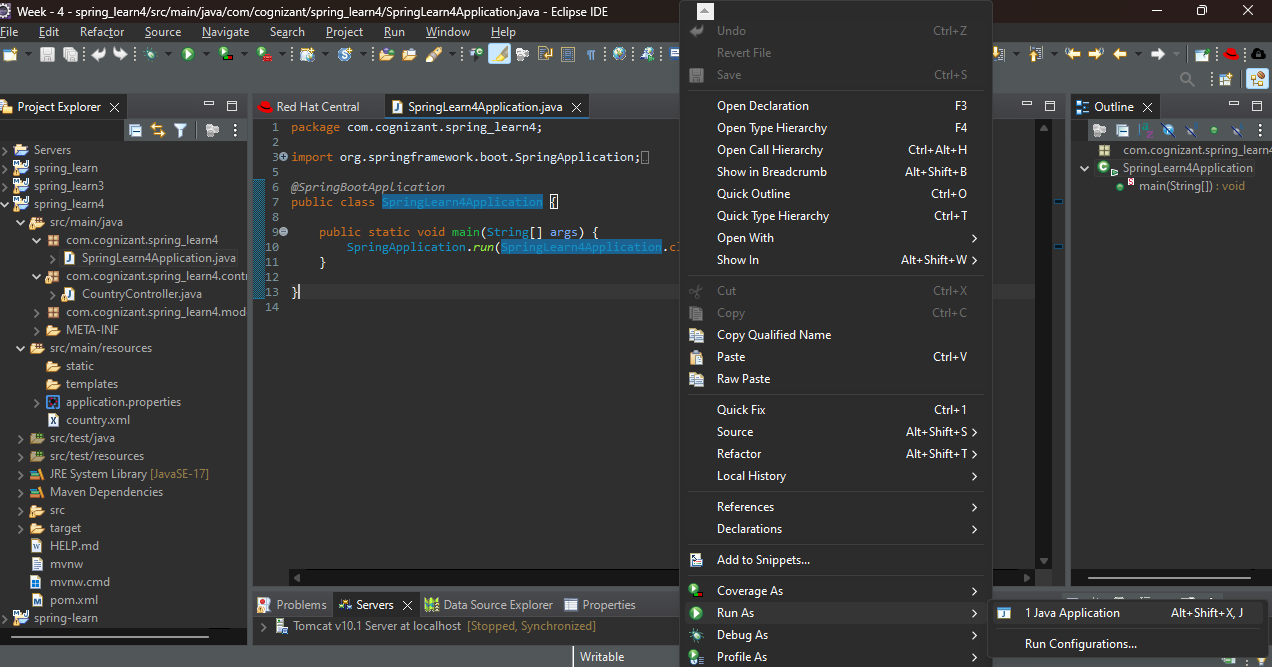
}



**Step – 6 :** Go to application.properties and enter port number.

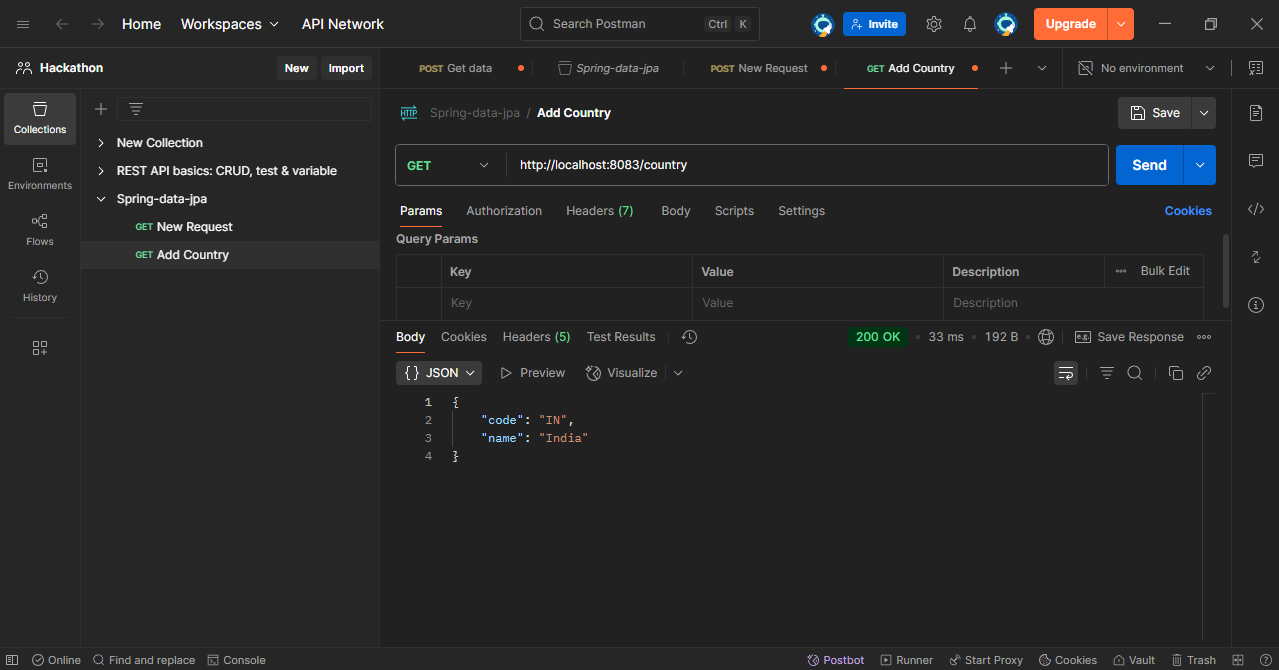


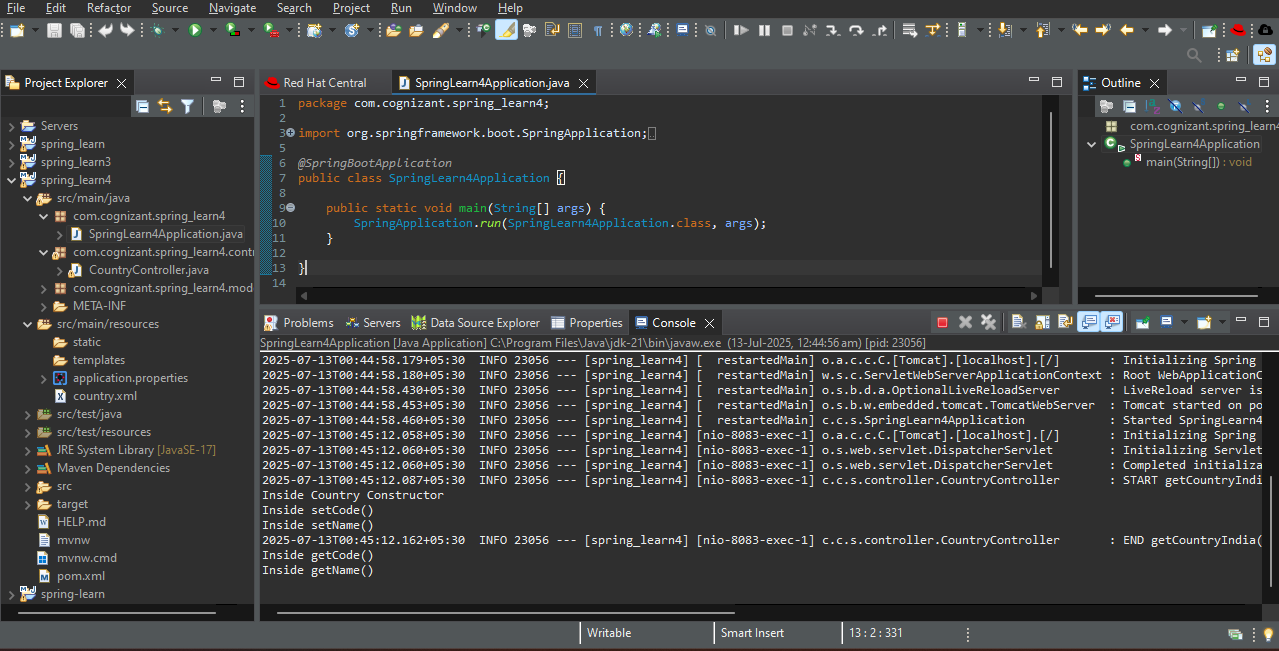
**Step – 7 :** Write the main code SpringLearn4Application.java and run as java application.



OUTPUT :

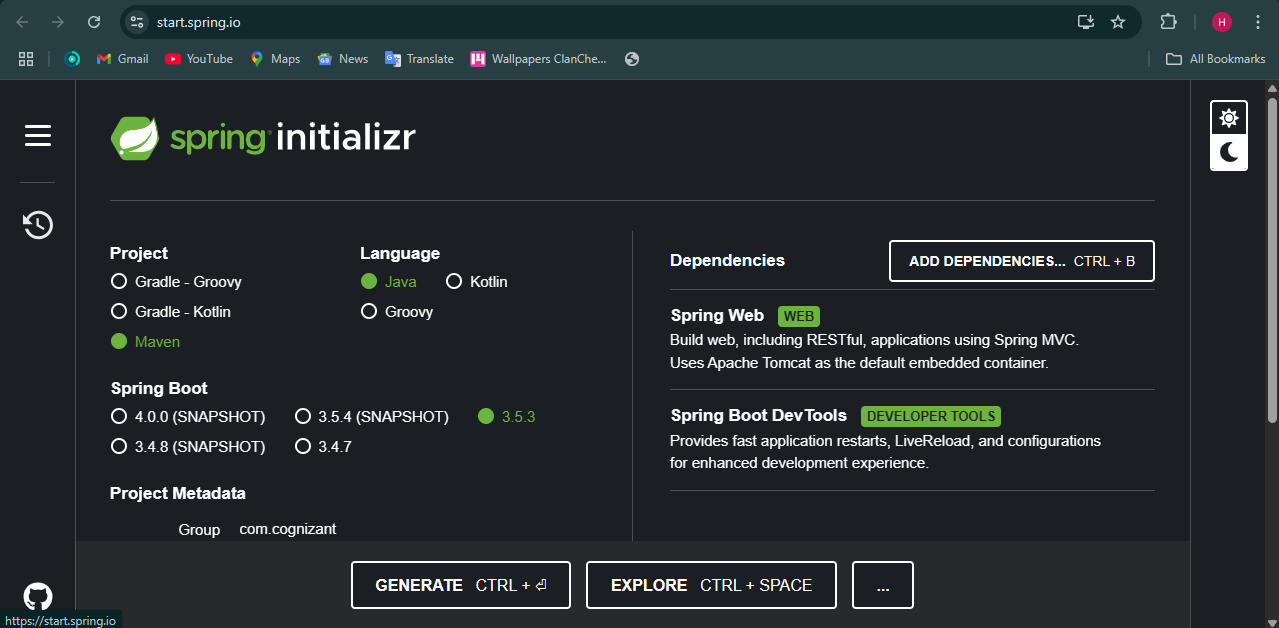
Enter the link in Postman <http://localhost:8083/country>



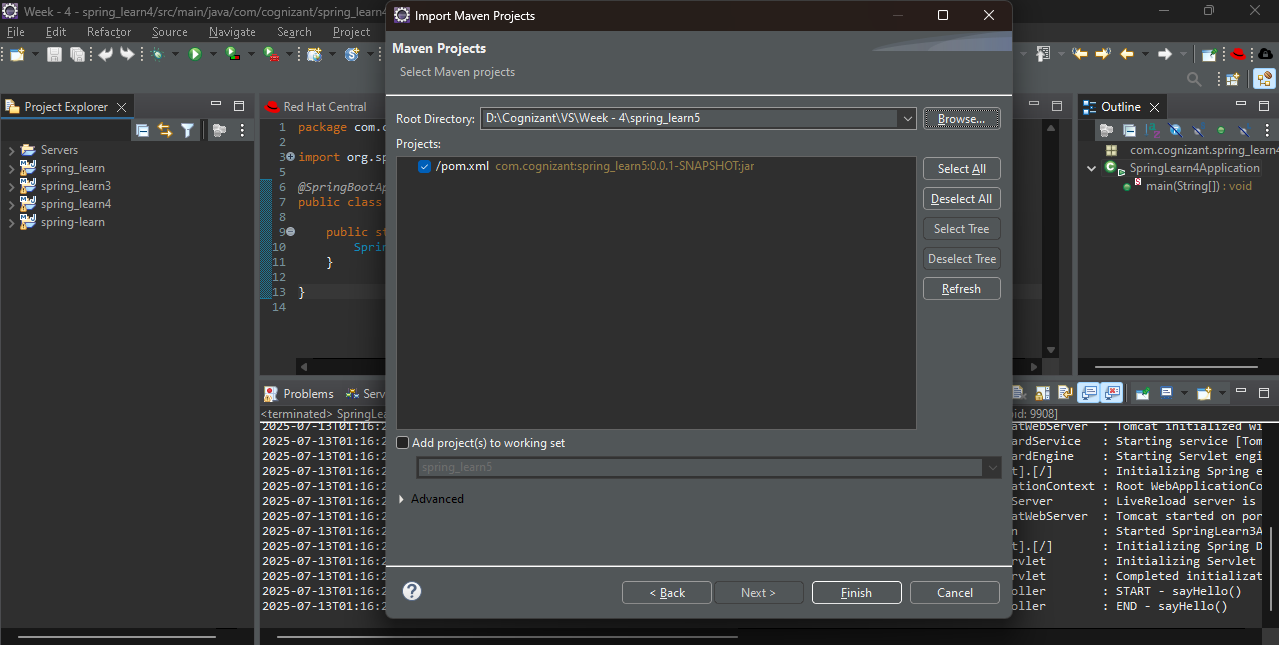


1. REST - Get country based on country code

**Step – 1 :** Go to spring initializr.io create a spring boot project.



**Step – 2 :** Extract the folder and import it in Eclipse.



**Step – 3 :** Create a package model and in it create Country.java class.

**Code :**

package com.cognizant.spring\_learn5.model;

public class Country {

private String code;

private String name;

public Country() {

System.out.println("Inside Country Constructor");

}

public String getCode() {

return code;

}

public void setCode(String code) {

System.out.println("Inside setCode()");

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

System.out.println("Inside setName()");

this.name = name;

}

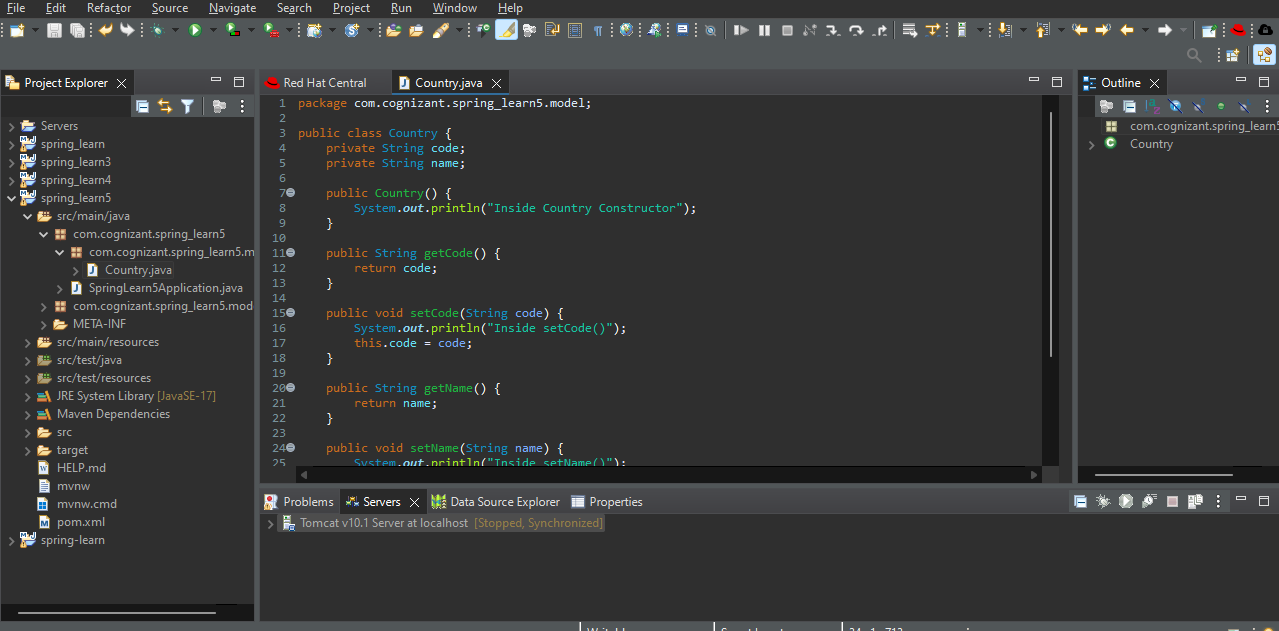
@Override

public String toString() {

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

}

}



**Step – 4 :** Go to src/main/resources, create a file country.xml.

**Code :**

<?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="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<bean class="com.cognizant.spring\_learn5.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn5.model.Country">

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

<property name="name" value="United States"/>

</bean>

<bean class="com.cognizant.spring\_learn5.model.Country">

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

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

</bean>

<bean class="com.cognizant.spring\_learn5.model.Country">

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

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

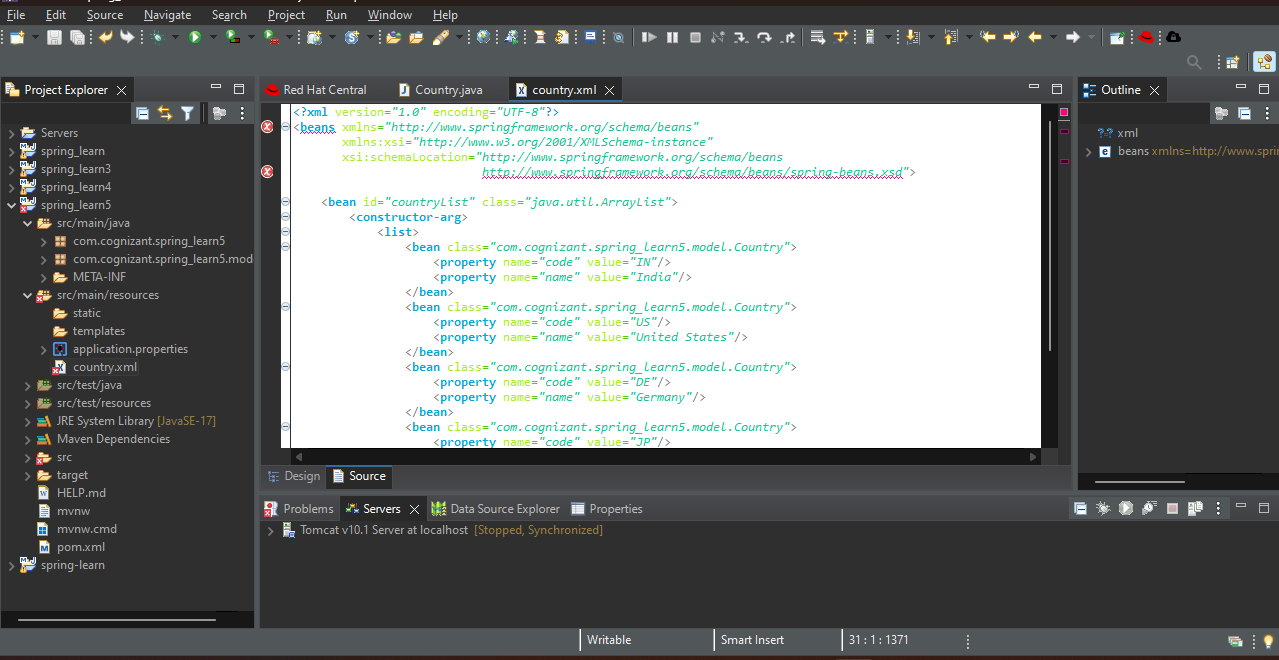
</bean>

</list>

</constructor-arg>

</bean>

</beans>



**Step – 5 :** Create a package service and in it create a class CountryService.java.

**Code :**

package com.cognizant.spring\_learn5.service;

import com.cognizant.spring\_learn5.model.Country;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

public Country getCountry(String code) {

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

List<Country> countries = (List<Country>) context.getBean("countryList");

return countries.stream()

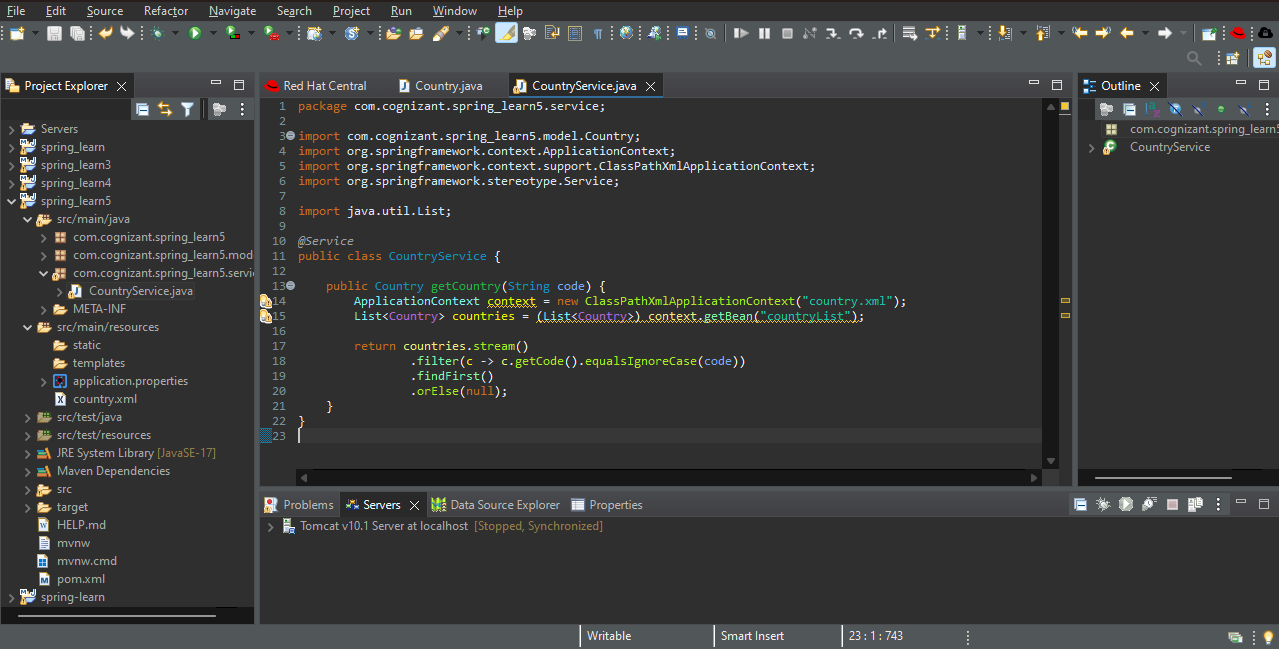
.filter(c -> c.getCode().equalsIgnoreCase(code))

.findFirst()

.orElse(null);

}

}



**Step – 6 :** Create a package controller and in it create a class CountryController.java.

**Code :**

package com.cognizant.spring\_learn5.controller;

import com.cognizant.spring\_learn5.model.Country;

import com.cognizant.spring\_learn5.service.CountryService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

@GetMapping("/country/{code}")

public Country getCountry(@PathVariable String code) {

LOGGER.info("START getCountry()");

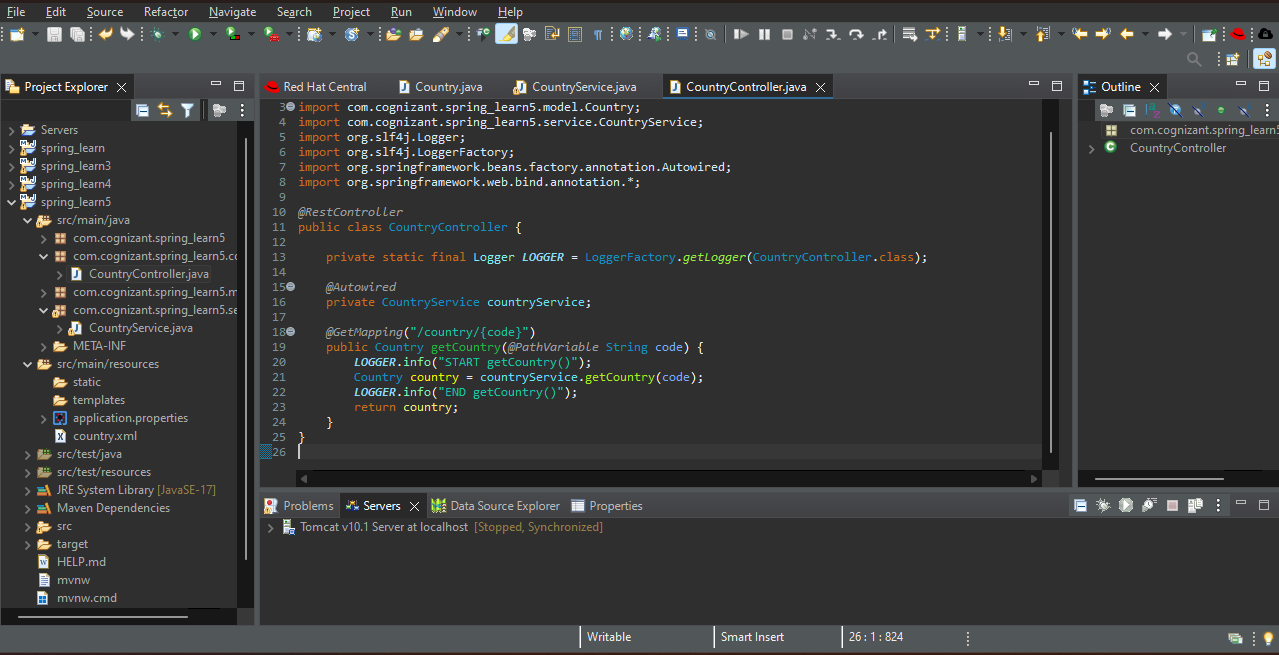
Country country = countryService.getCountry(code);

LOGGER.info("END getCountry()");

return country;

}

}



**Step – 7 :** Go to SpringLearn5Application.java and write code.

**Code :**

package com.cognizant.spring\_learn5;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

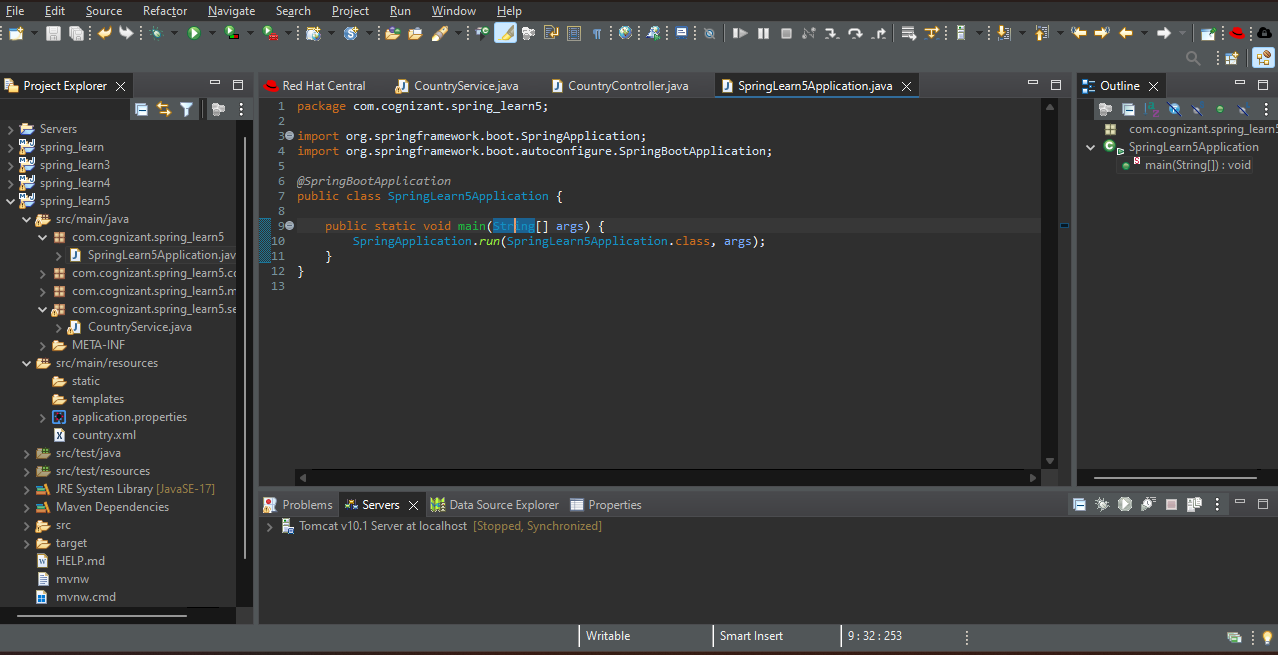
public class SpringLearn5Application {

public static void main(String[] args) {

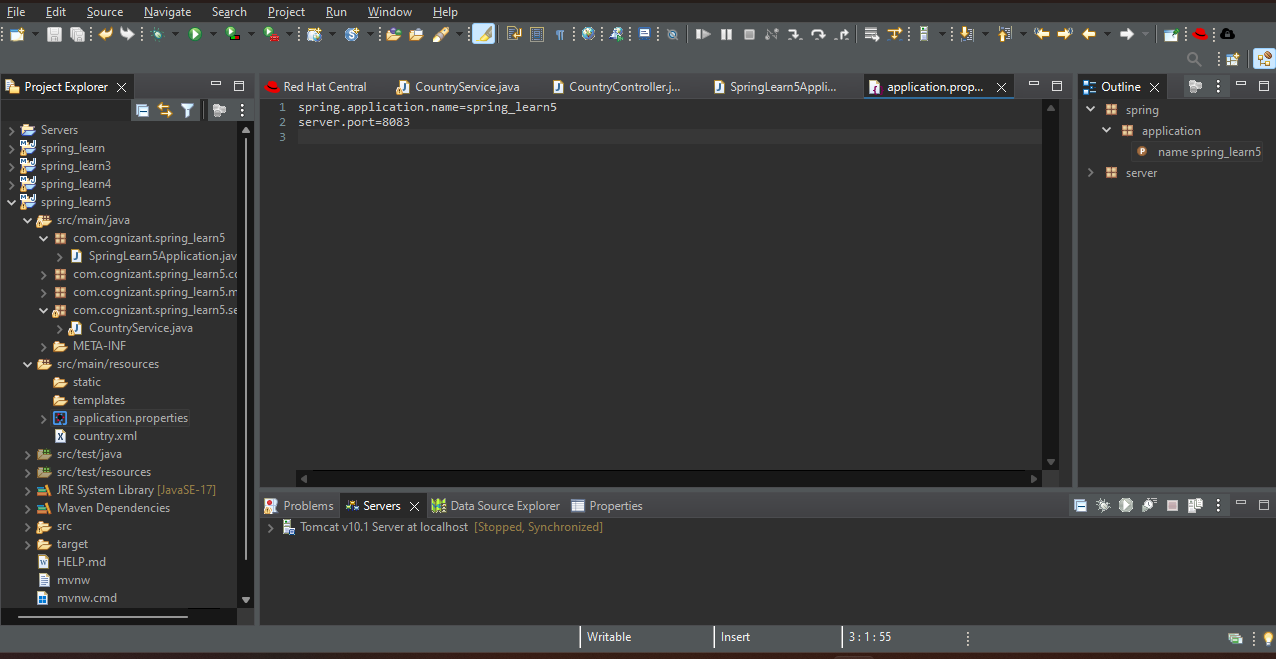
SpringApplication.run(SpringLearn5Application.class, args);

}

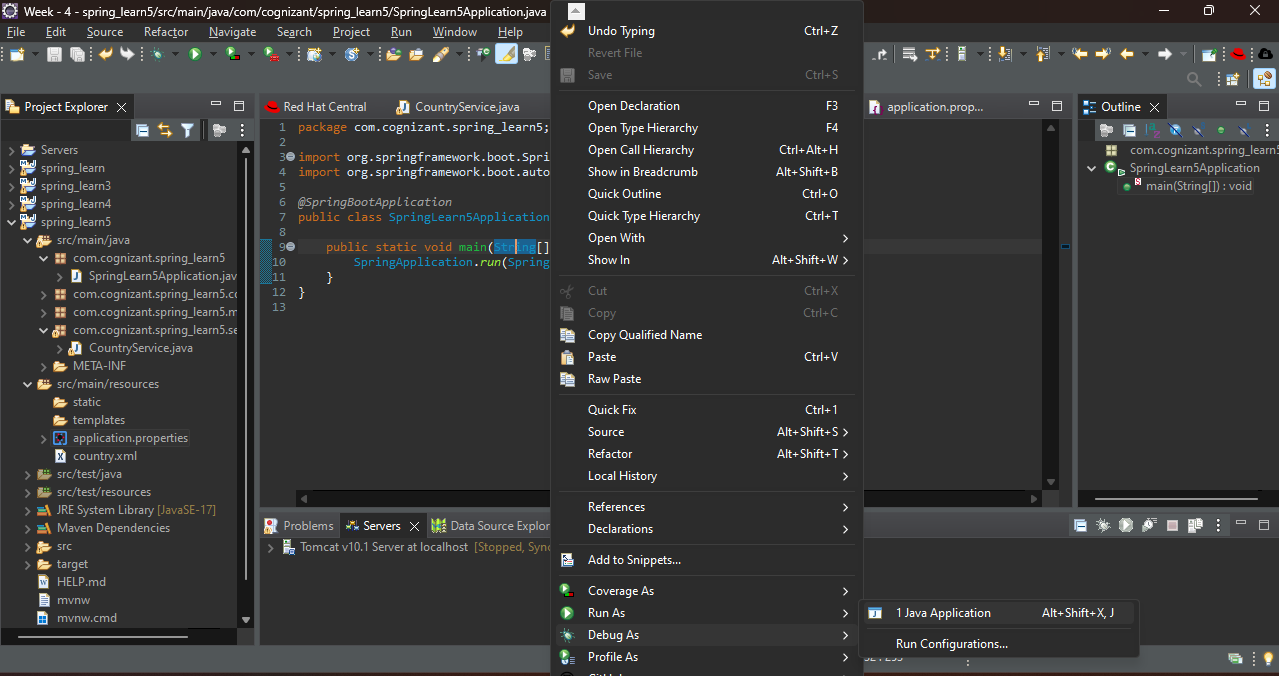
}



**Step – 8 :** Go to src/main/resources 🡪 application.properties and enter your port number.

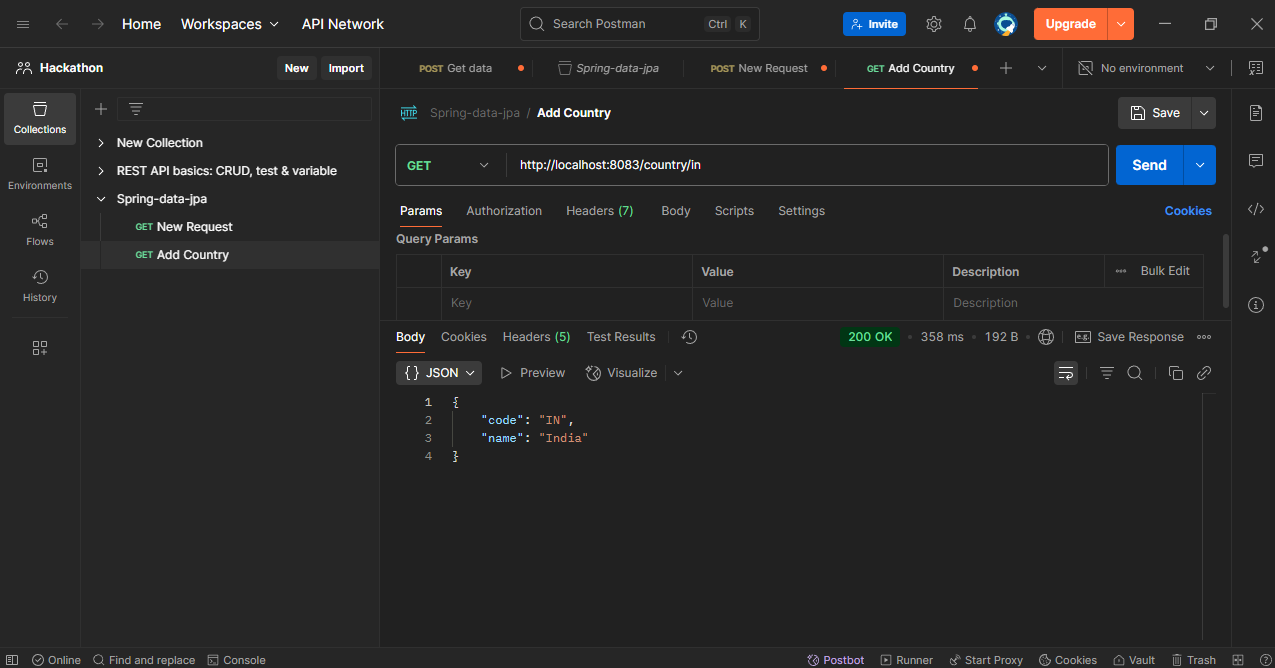


**Step – 9 :** Run the SpringLearn5Application.java as java application.



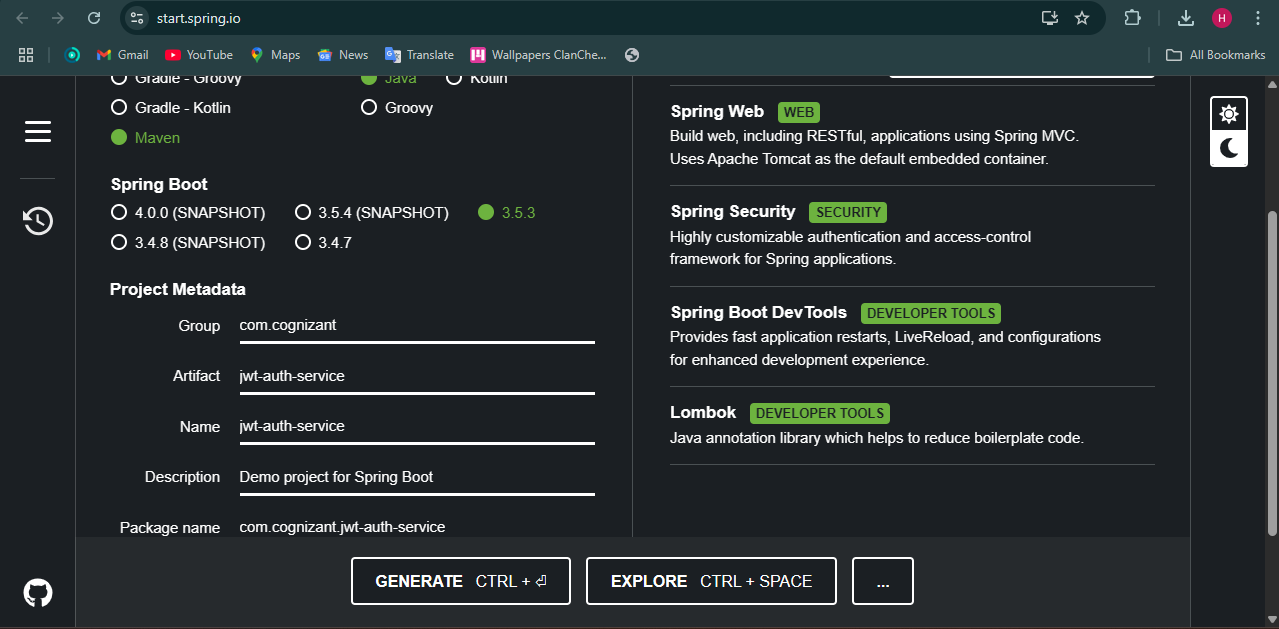
OUTPUT :

Enter <http://localhost:8083/country/in> in postman.

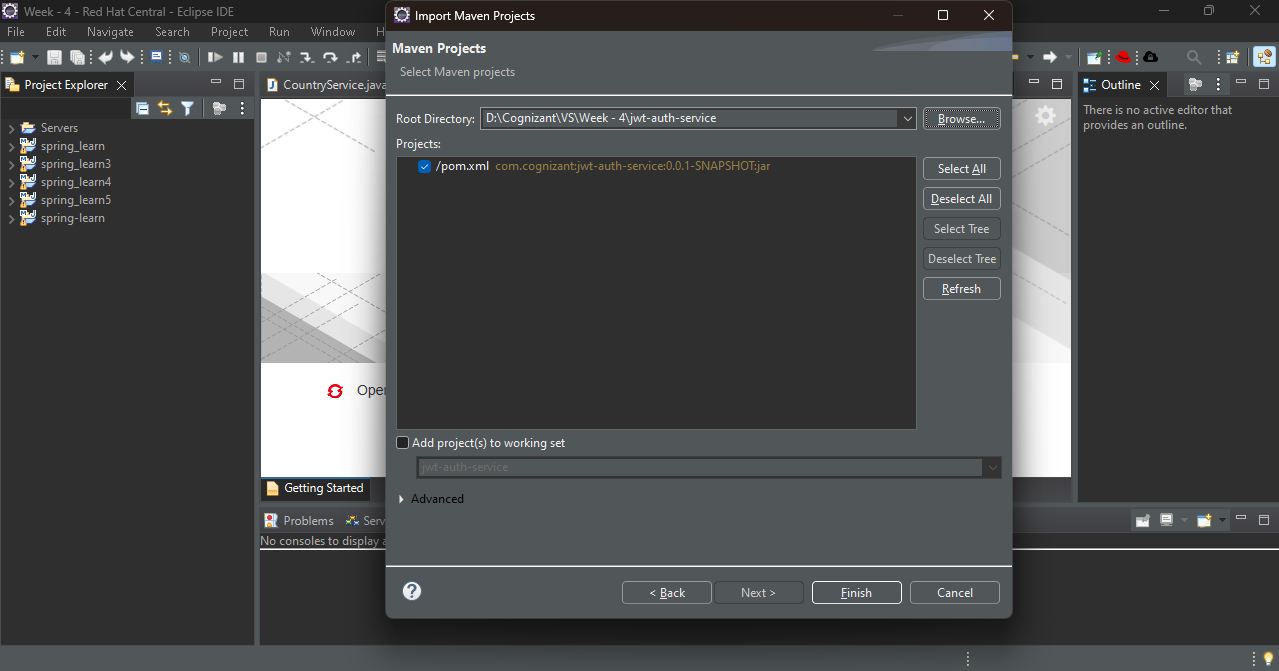


1. Create authentication service that returns JWT.

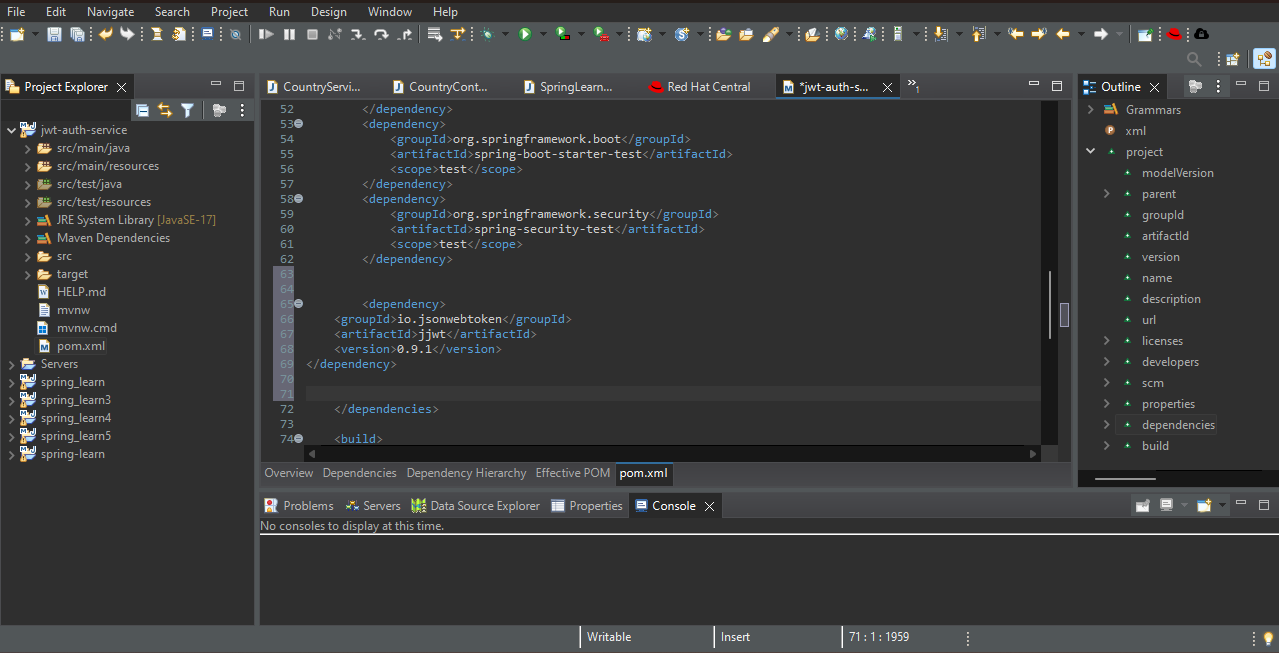
**Step – 1 :** Go to spring initializr.io create a spring boot project.



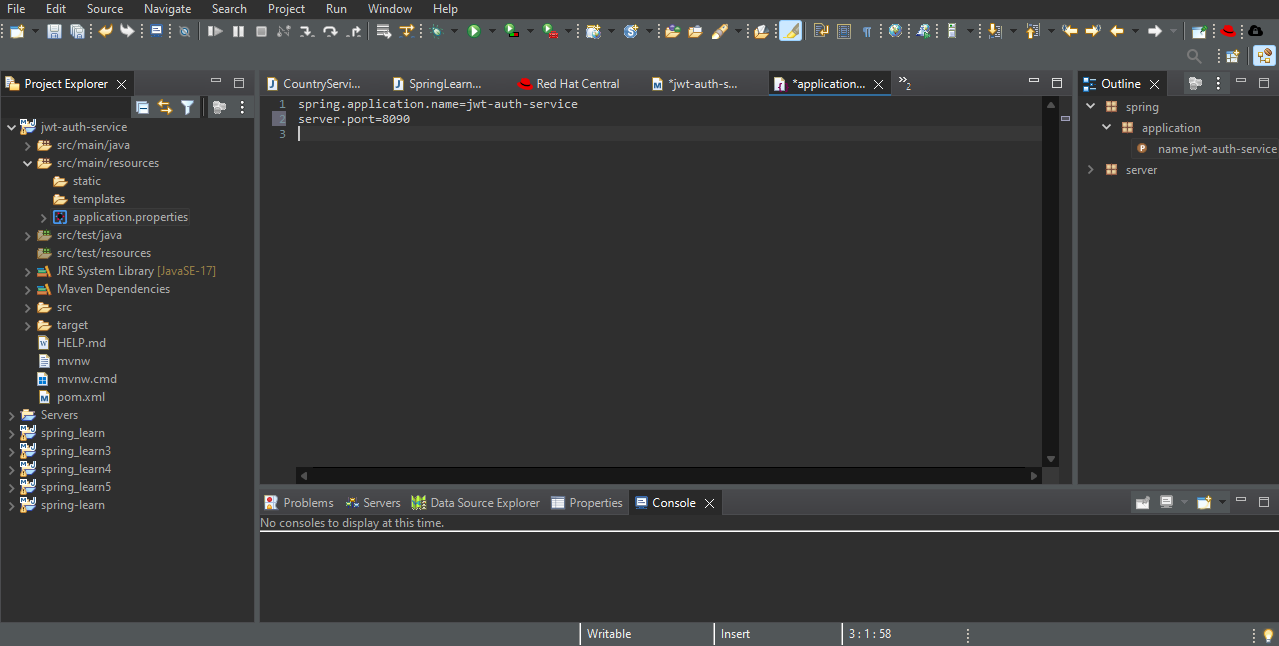
**Step – 2 :** Extract the folder and import it in Eclipse.



**Step – 3 :** Add jwt token dependency in pom.xml.



**Step – 4 :** Go to src/main/resources 🡪application.properties and enter port number.



**Step – 5 :** Create a package com.cognizant.jwtauthservice.util and in it create a class JwtUtil.java.

**Code :**

package com.cognizant.jwtauthservice.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import org.springframework.stereotype.Component;

import java.util.Date;

@Component

public class JwtUtil {

private String secretKey = "cognizant123";

public String generateToken(String username) {

long now = System.currentTimeMillis();

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(now))

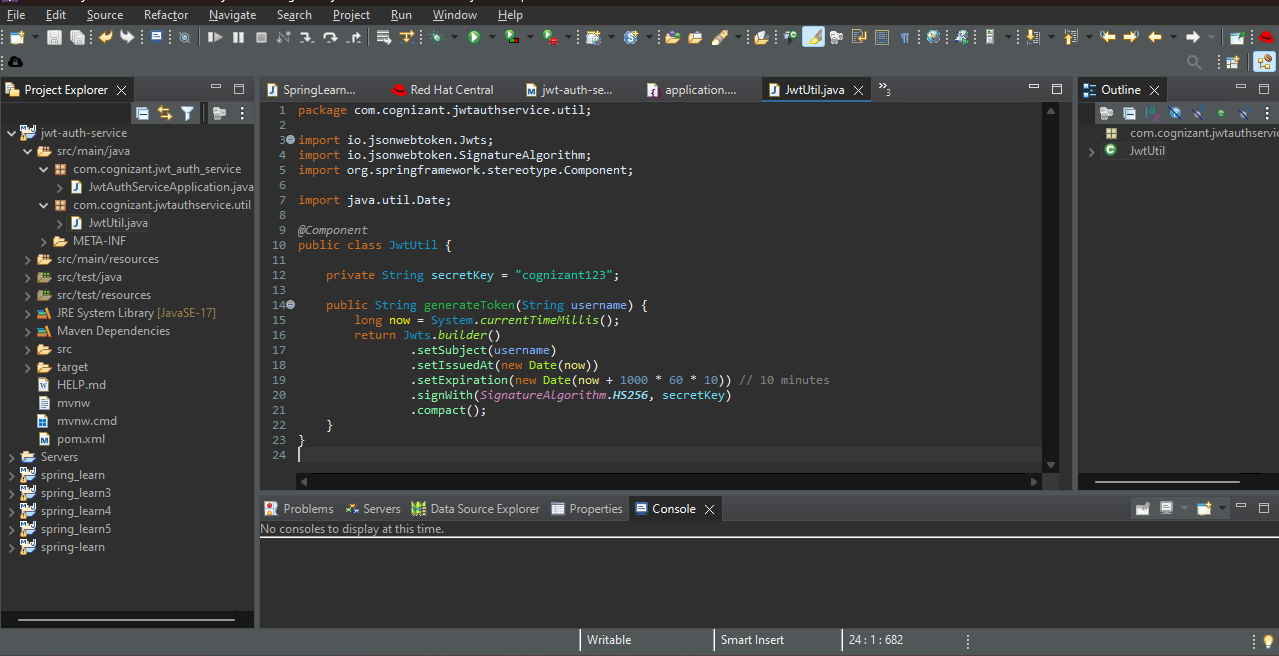
.setExpiration(new Date(now + 1000 \* 60 \* 10)) // 10 minutes

.signWith(SignatureAlgorithm.HS256, secretKey)

.compact();

}

}



**Step – 6 :** Create a package com.cognizant.jwtauthservice.Controllerand in it create a class AuthController.java.

**Code :**

package com.cognizant.jwtauthservice.controller;

import com.cognizant.jwtauthservice.util.JwtUtil;

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

import org.springframework.http.\*;

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

import java.nio.charset.StandardCharsets;

import java.util.Base64;

@RestController

public class AuthController {

@Autowired

private JwtUtil jwtUtil;

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {

if (authHeader == null || !authHeader.startsWith("Basic ")) {

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Missing or invalid Authorization header");

}

// Decode base64 credentials

String base64Credentials = authHeader.substring("Basic ".length());

byte[] decoded = Base64.getDecoder().decode(base64Credentials);

String credentials = new String(decoded, StandardCharsets.UTF\_8);

String[] values = credentials.split(":", 2);

String username = values[0];

String password = values[1];

// Validate (replace with actual check in real case)

if ("user".equals(username) && "pwd".equals(password)) {

String token = jwtUtil.generateToken(username);

return ResponseEntity.ok().body("{\"token\":\"" + token + "\"}");

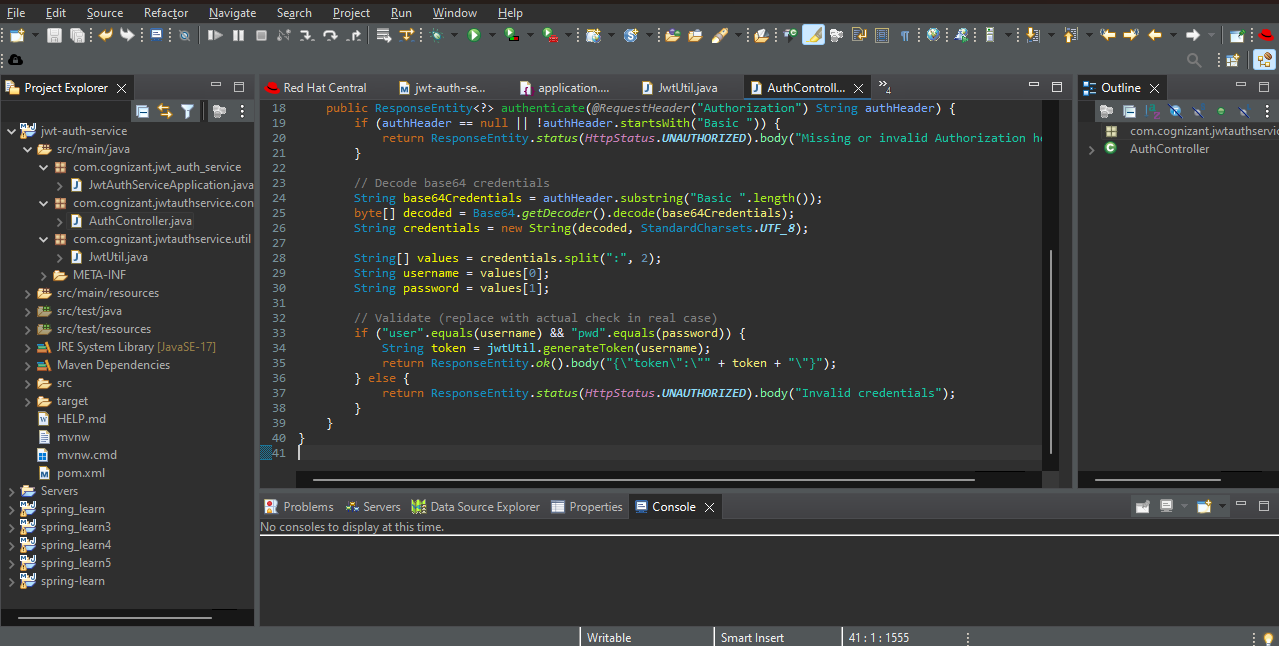
} else {

return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("Invalid credentials");

}

}

}



**Step – 7 :** Create a package com.cognizant.jwtauthservice. config and in it create a class SecurityConfig.java.

**Code :**

package com.cognizant.jwtauthservice.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.\*;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.csrf().disable()

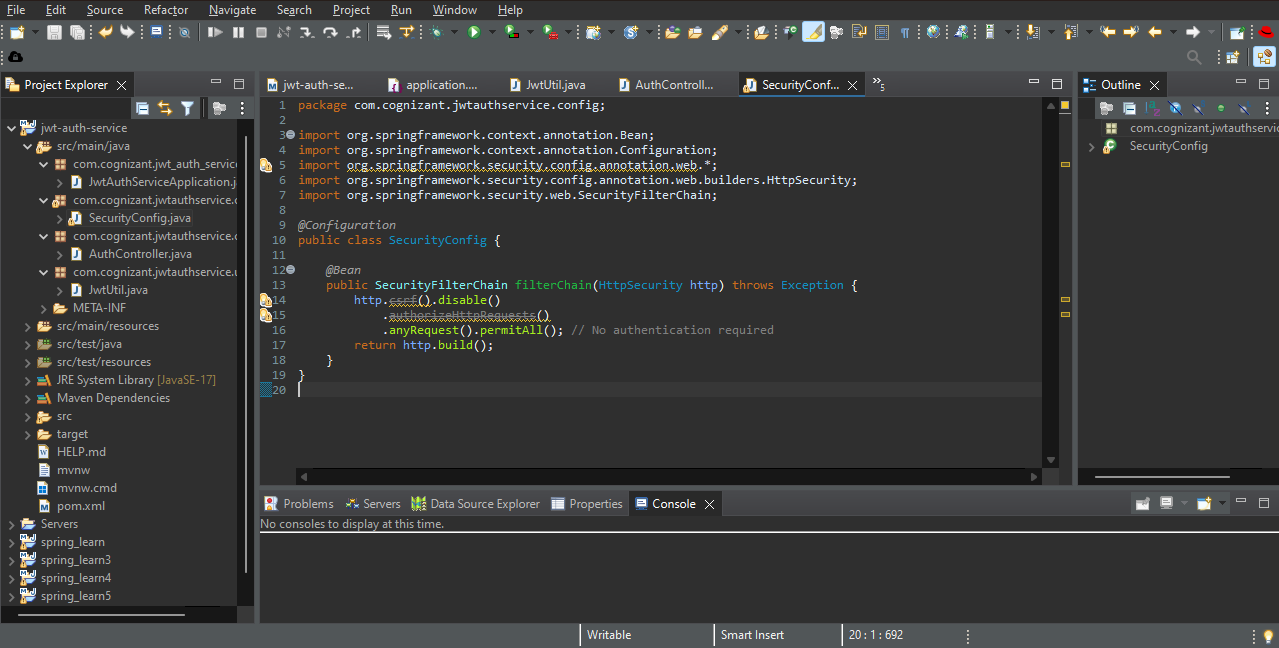
.authorizeHttpRequests()

.anyRequest().permitAll(); // No authentication required

return http.build();

}

}



**Step – 8 :** Run the application

