**Digital Nurture 4.0 – Week 4**

**Spring Rest using Spring Boot 3**

**1.Create a Spring Web Project using Maven:**

**Objective:**

To Create a Spring Rest Using Spring Boot 3 and Spring Web Project using Maven.

**Step 1:**

Create a spring bot project named springlearn.Refer below details

* + **Project:** Maven
  + **Language:** Java
  + **Spring Boot:** 3.5.3
  + **Group:** com.cognizant
  + **Artifact:** springlearn
  + **Name:** springlearn
  + **Package Name:** com.cognizant.springlearn
  + **Packaging:** Jar
  + **Java version:** 24

**Dependencies:**

* + Spring Boot DevTools
  + Spring Web

**Step 2:**

In **SpringlearnApplication.java,**Add SLF4J Logger and add Add sample logging config on **application.properties**.

**SpringlearnApplication.java**:

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("END");

}

}

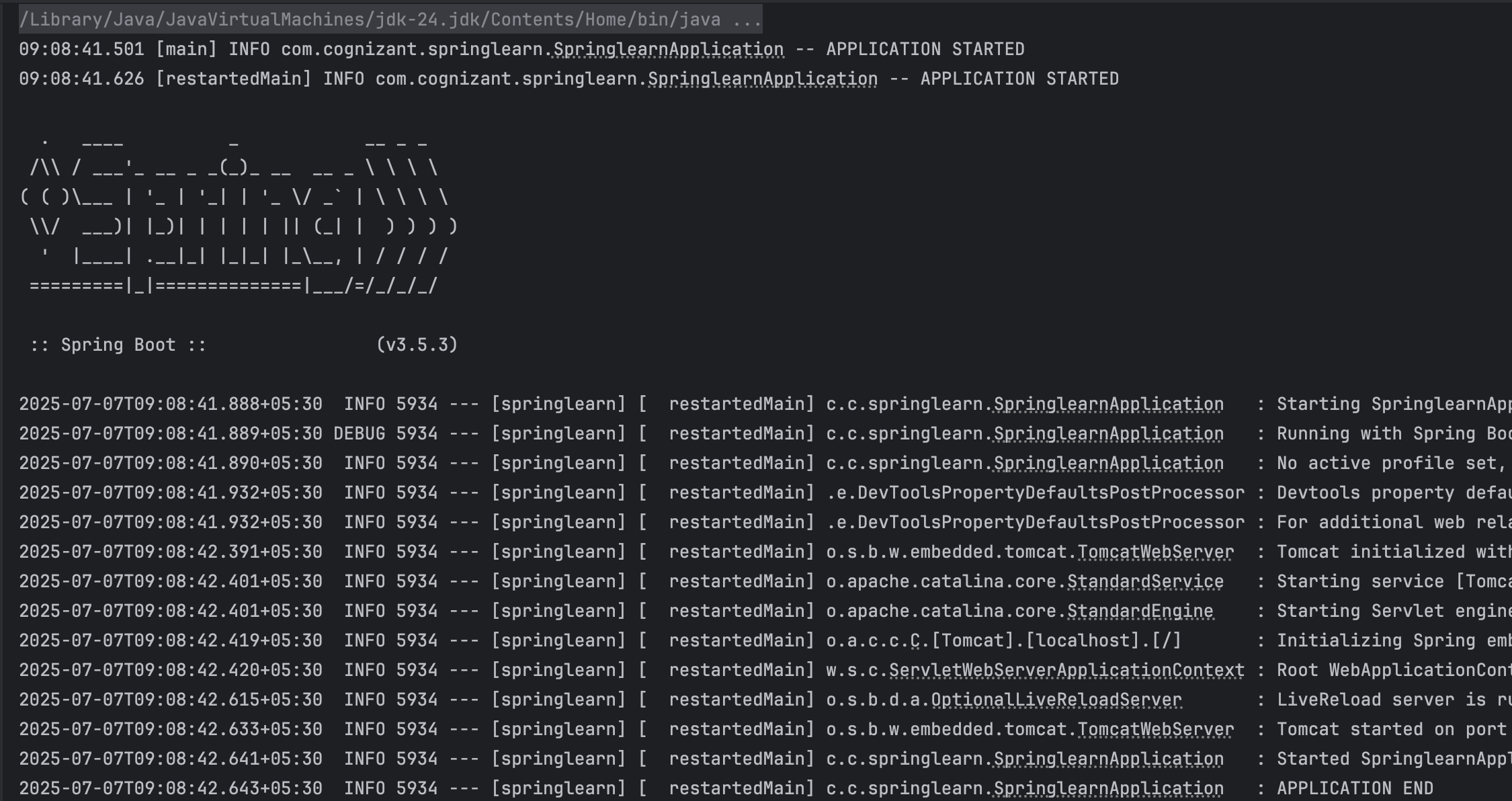
**application.properties**:

logging.level.org.springframework=INFO

logging.level.com.cognizant.springlearn=DEBUG

**Step 3:**

Run **SpringLearnApplication** to start thes server with log details.



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

**Objective:**

To Create an Load Country from Spring Configuration XML Uaing the Spring Core.

**Step 1:**

Use the same springlearn project.Configure or Add some changes.In **SpringLearnApplication.java**,

package com.cognizant.springlearn;

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;

@SpringBootApplication

public class SpringlearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringlearnApplication.class, args);

SpringlearnApplication app = new SpringlearnApplication();

app.displayCountry();

LOGGER.info("END");

}

public void displayCountry() {

LOGGER.info("START");

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

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

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

LOGGER.info("END");

}

}

**Step 2:**

Create **Country.java** in the src/main/java/com/cognizant/springlearn directory and **country.xml** in the src/main/resources directory.

**Country.java**:

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 + '\'' +

'}';

}

}

**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

https://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

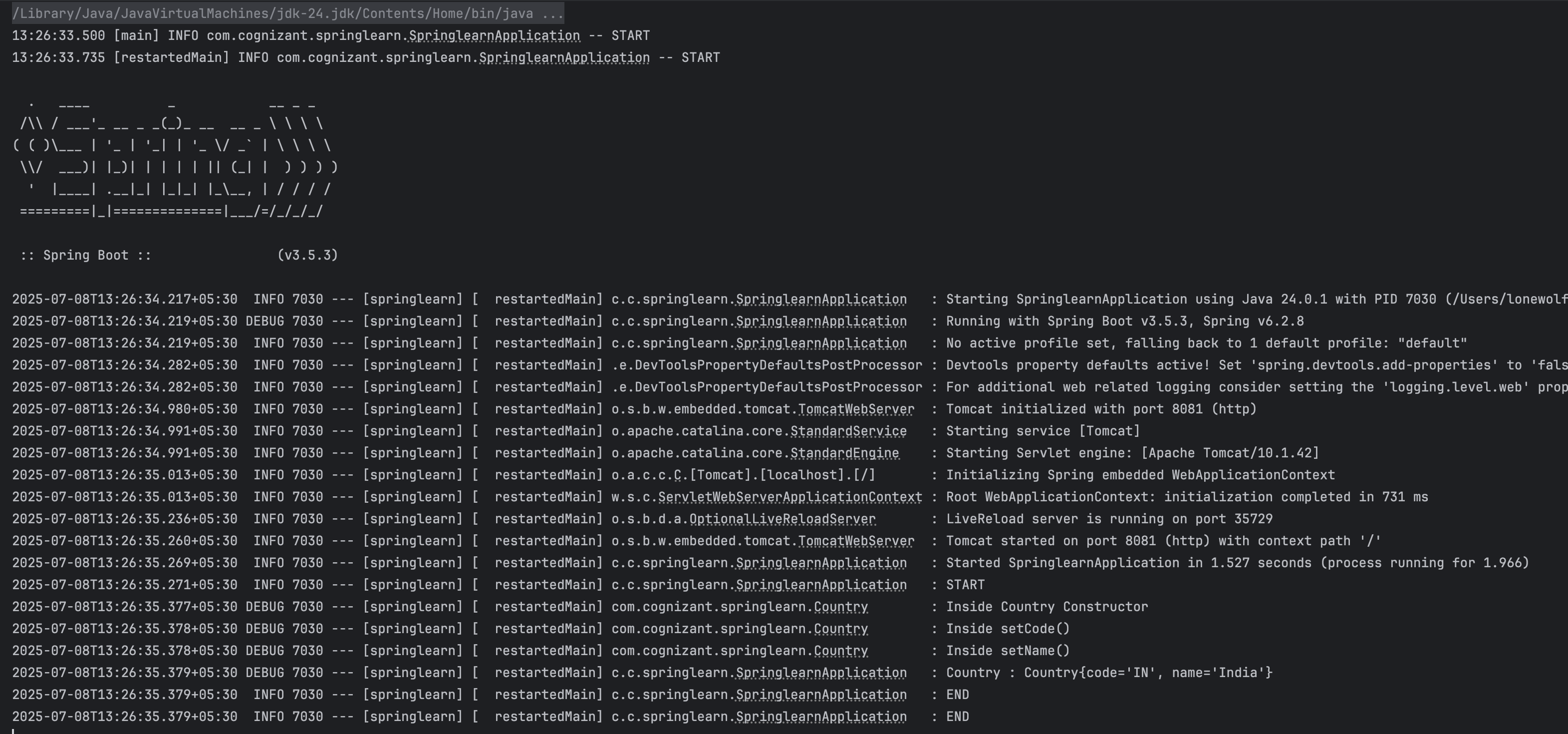
<property name="name" value="India"/>

</bean>

</beans>

**Step 3:**

Run **SpringLearnApplication** to start the server.



**2. Spring-REST**

**1.Hello World RESTful Web Service:**

**Objective:**

To Print the Hello World Using the RESTfuk Web Services.

**Step 1:**

Use the same springlearn project.Configure some changes.In **SpringLearnApplication.java**,

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("END");

}

}

**Step 2:**

Create **HelloController.java** in the com.cognizant.springlearn.controller directory and configure **application.properties.**

**HelloController.java:**

package com.cognizant.springlearn.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");

String message = "Hello World!!";

LOGGER.debug("Message: {}", message);

LOGGER.info("END");

return message;

}

}

**application.properties**:

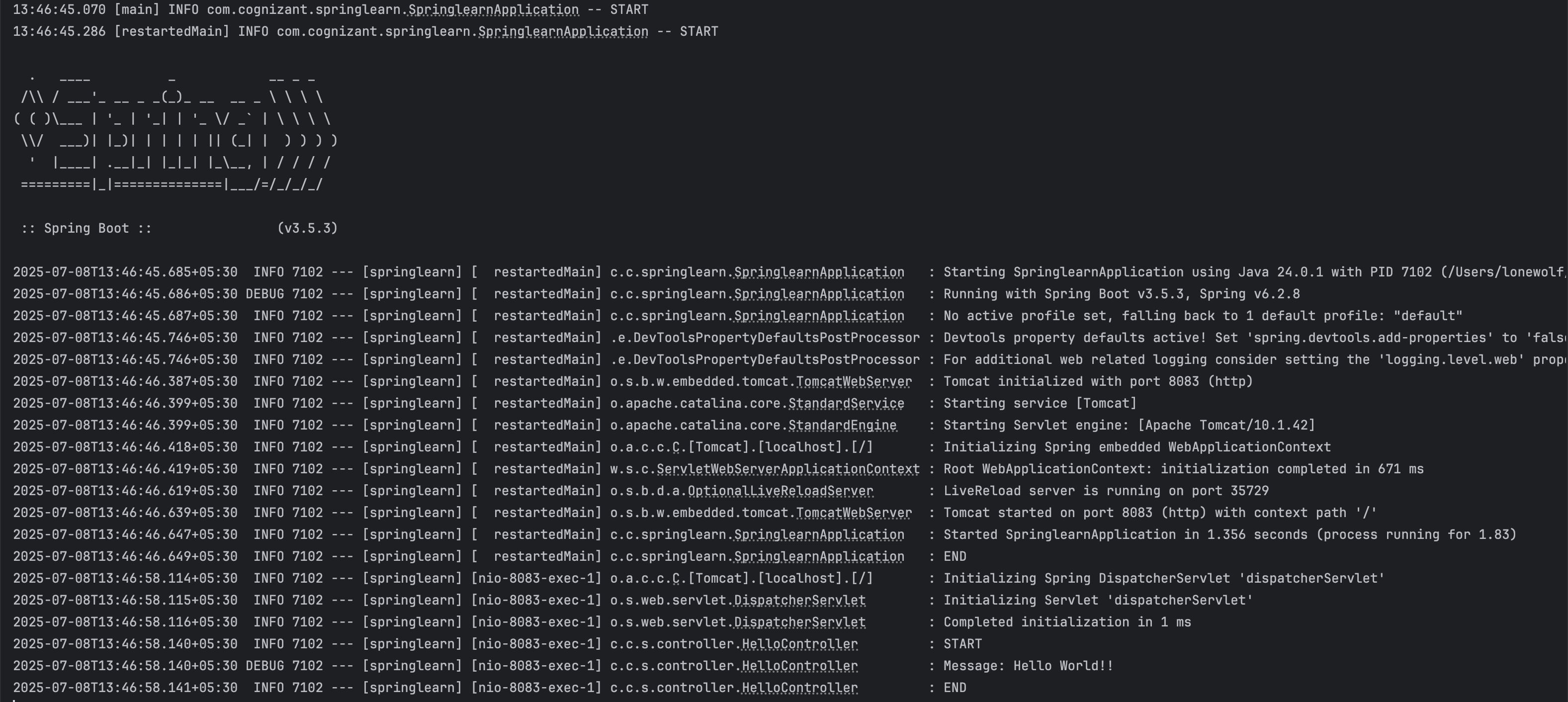
server.port=8083

logging.level.org.springframework=INFO

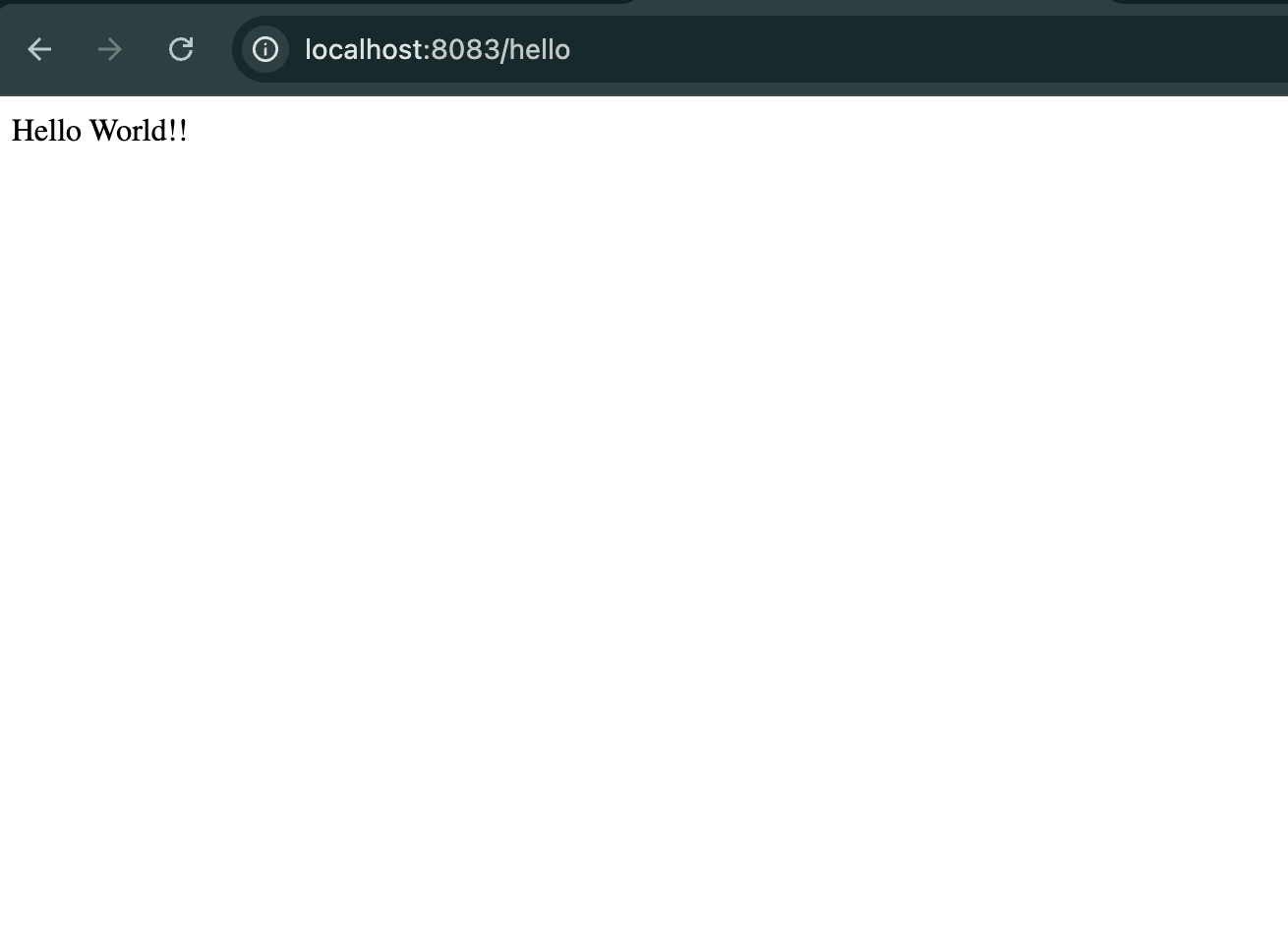
logging.level.com.cognizant.springlearn=DEBUG

**Step 3:**

Run **SpringLearnApplication** to start the server.



Open <http://localhost:8083/hello> on your browser or postman.



**2.REST - Country Web Service:**

**Objective:**

To create an Country WebService using the REST

**Step 1:**

Use the same springlearn project.Configure or add some changes.In **SpringLearnApplication.java**,

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("END");

}

}

**Step 2:**

Create **Country.java** in the src/main/java/com/cognizant/springlearn directory,**country.xml** in the src/main/resources directory,**CountryController.java** in the src/main/java/com/cognizant/springlearn/controller directory and **application.properties** remains same.

**Country.java**:

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 + '\'' +

'}';

}

}

**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

https://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>

**CountryController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.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");

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

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

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

LOGGER.info("END");

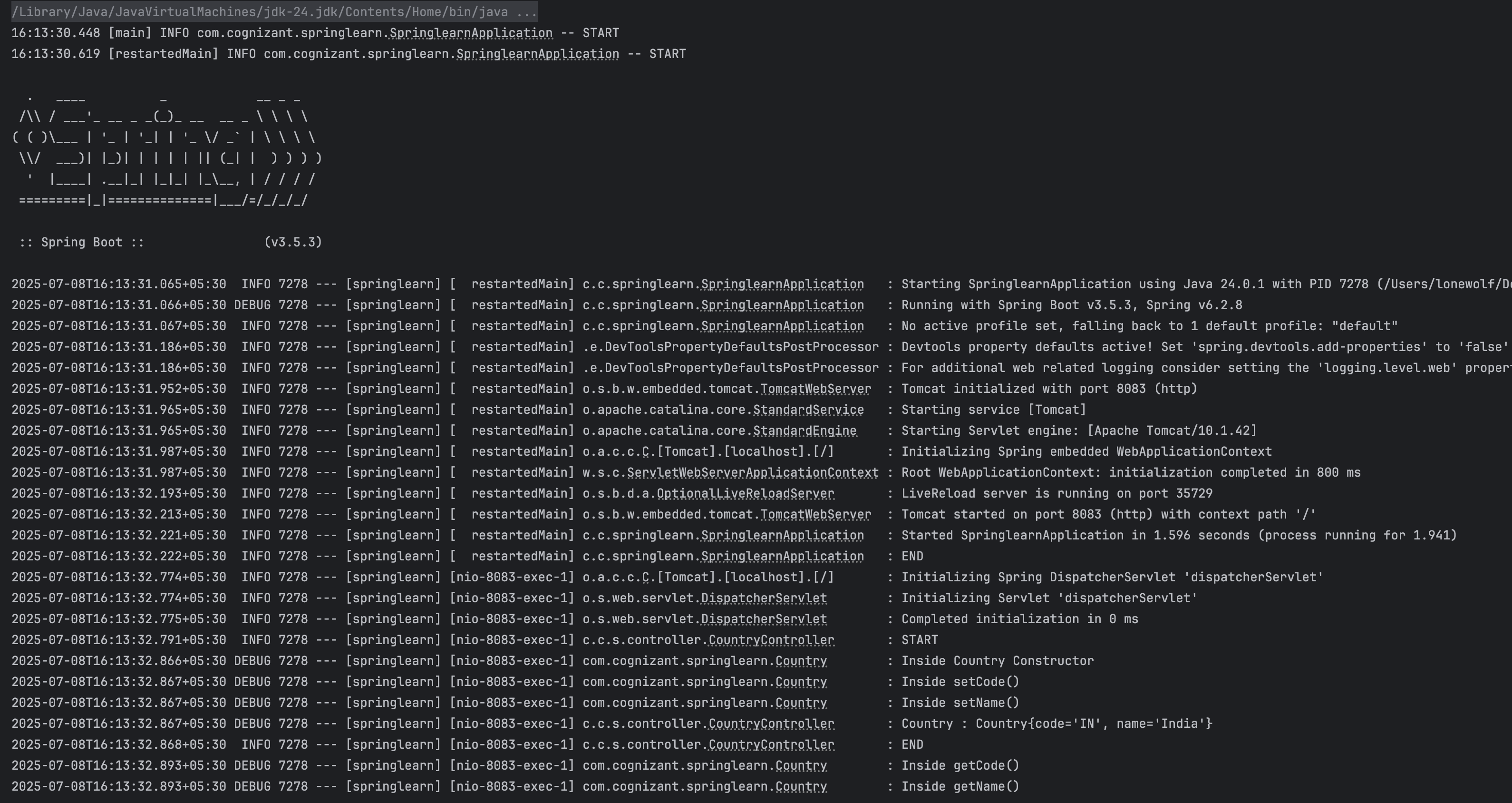
return country;

}

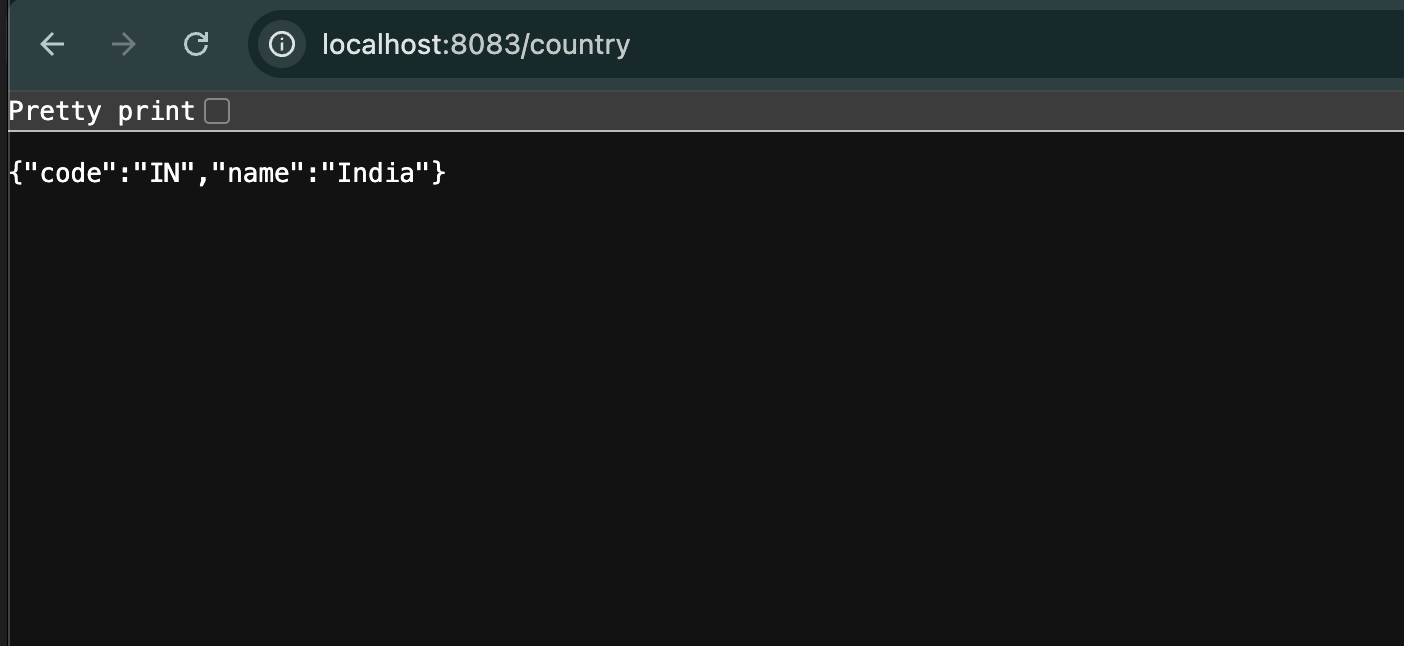
}

**Step 3:**

Run **SpringLearnApplication** to start the server.



Run <http://localhost:8083/country> on brower.



**3.REST - Get country based on country code:**

**Objective:**

To create an get Country Based on country code Using The spring Rest Concept.

**Step 1:**

Use the same springlearn project.Configure or add some changes.In **SpringLearnApplication.java**,

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("END");

}

}

**Step 2:**

* Create Country.java in the src/main/java/com/cognizant/springlearn directory,country.xml in the src/main/resources directory,CountryService.java in the src/main/java/com/cognizant/springlearn/service directory,CountryNotFoundException.java in the src/main/java/com/cognizant/springlearn/service/exception directory,GlobalExceptionHandler.java in the src/main/java/com/cognizant/springlearn/service/exception,CountryController.java in the src/main/java/com/cognizant/springlearn/controller directory, and no need to chnages in the application.properties.

**Country.java**:

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 + '\'' +

'}';

}

}

**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

https://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

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

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

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

</bean>

<bean id="countryList" class="java.util.ArrayList">

<constructor-arg>

<list>

<ref bean="in"/>

<ref bean="us"/>

<ref bean="de"/>

<ref bean="jp"/>

</list>

</constructor-arg>

</bean>

</beans>

**CountryService.java**:

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.Country;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class CountryService {

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

public Country getCountry(String code) throws CountryNotFoundException {

LOGGER.info("START");

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

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

for (Country country : countryList) {

if (country.getCode().equalsIgnoreCase(code)) {

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

LOGGER.info("END");

return country;

}

}

LOGGER.error("Country with code {} not found", code);

throw new CountryNotFoundException("Country not found");

}

}

**CountryNotFoundException.java:**

package com.cognizant.springlearn.service.exception;

public class CountryNotFoundException extends Exception {

public CountryNotFoundException(String message) {

super(message);

}

}

**GlobalExceptionHandler.java**:

package com.cognizant.springlearn.exception;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

import java.util.HashMap;

import java.util.Map;

@ControllerAdvice

public class GlobalExceptionHandler {

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

@ExceptionHandler(CountryNotFoundException.class)

public ResponseEntity<Object> handleCountryNotFoundException(CountryNotFoundException ex) {

LOGGER.error("Handling CountryNotFoundException: {}", ex.getMessage());

Map<String, Object> body = new HashMap<>();

body.put("error", "Country not found");

body.put("message", ex.getMessage());

body.put("status", HttpStatus.NOT\_FOUND.value());

return new ResponseEntity<>(body, HttpStatus.NOT\_FOUND);

}

}

**CountryController.java**:

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.Country;

import com.cognizant.springlearn.service.CountryService;

import com.cognizant.springlearn.service.exception.CountryNotFoundException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

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

@RestController

public class CountryController {

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

@Autowired

private CountryService countryService;

@RequestMapping("/country")

public Country getCountryIndia() {

LOGGER.info("START");

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

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

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

LOGGER.info("END");

return country;

}

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

public Country getCountry(@PathVariable String code) throws CountryNotFoundException {

LOGGER.info("START");

Country country = countryService.getCountry(code);

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

LOGGER.info("END");

return country;

}

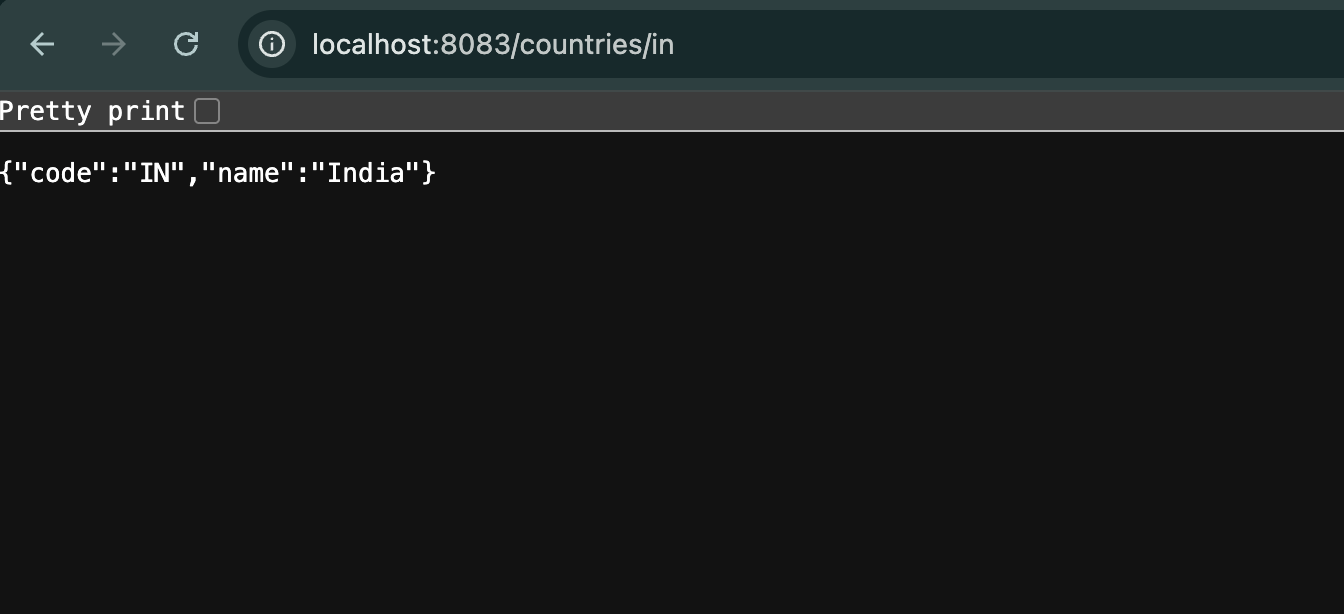
}

**Step 3:**

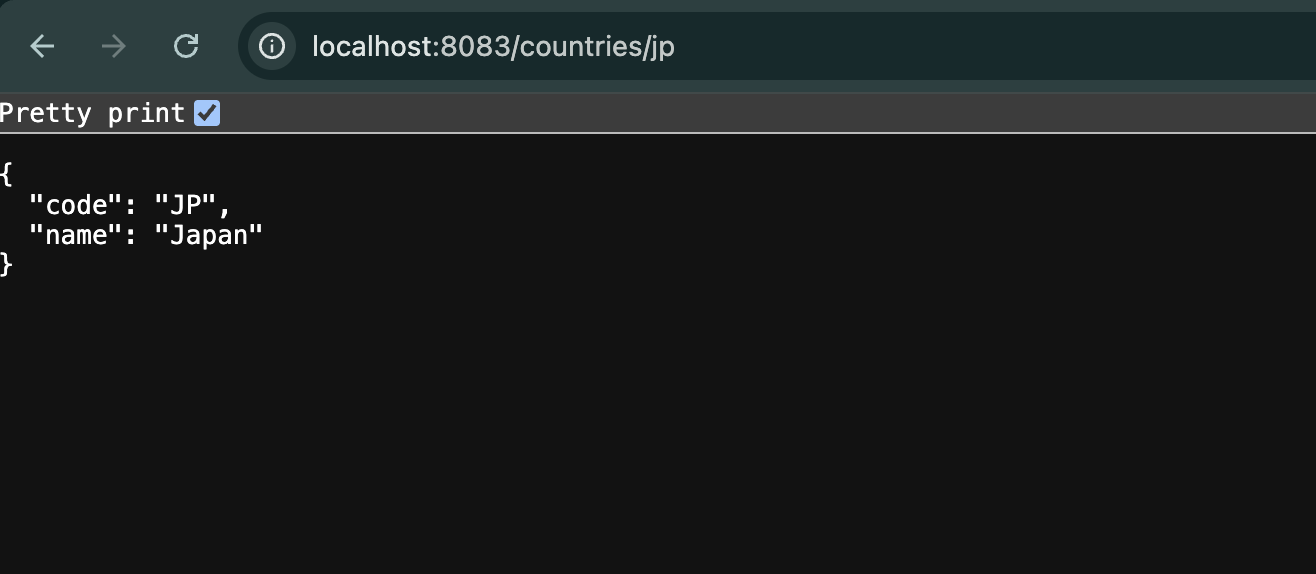
Run **SpringLearnApplication** to start the server.And Run below urls on browser.



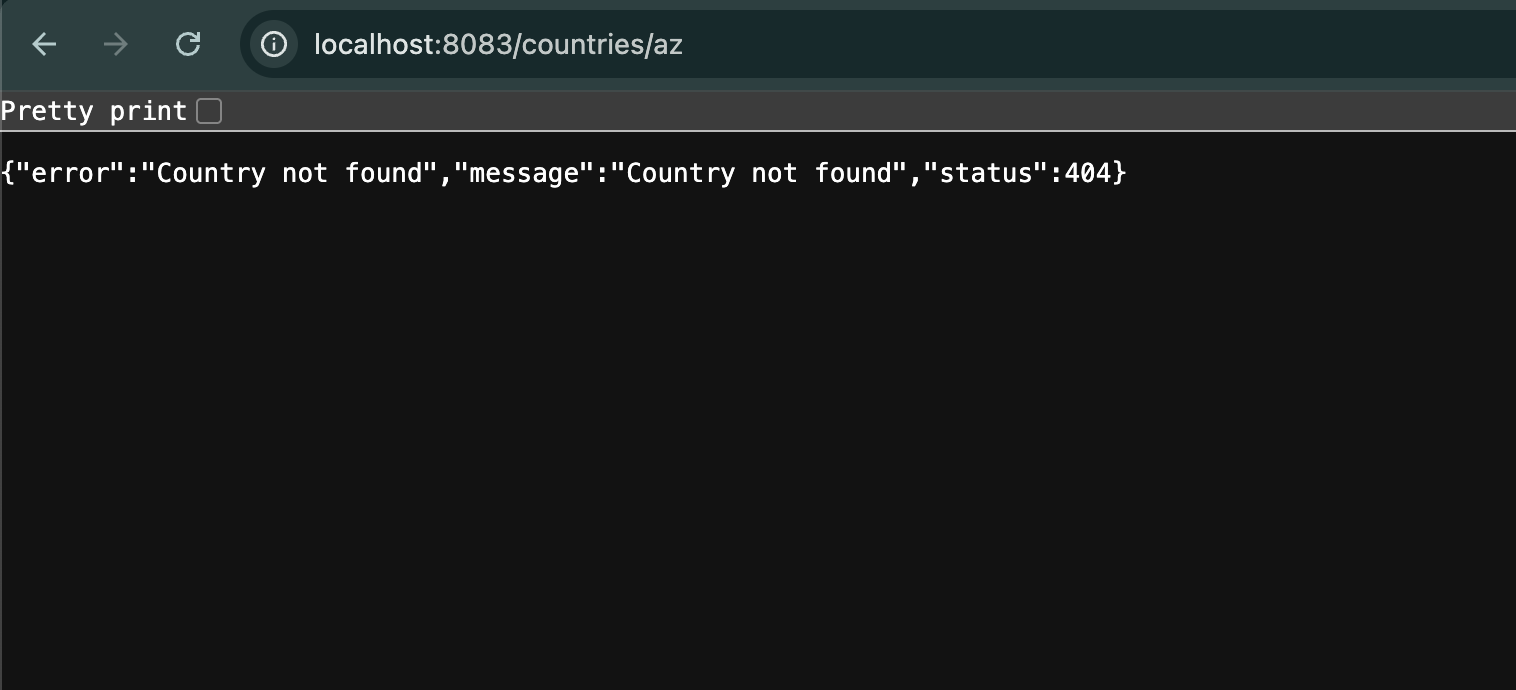
[http://localhost:8083/countries/in**,**](http://localhost:8083/countries/in,)



<http://localhost:8083/countries/jp,>



<http://localhost:8083/countries/az,>



**JWT-handson**

**1.Create authentication service that returns JWT:**

**Objective:**

To Create the authentication Service that returns JWT.

**Step 1:**

Use the same springlearn project.Configure or add some changes.In **pom.xml**,add addtional dependencies,

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-api</artifactId>

<version>0.11.5</version>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-impl</artifactId>

<version>0.11.5</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>io.jsonwebtoken</groupId>

<artifactId>jjwt-jackson</artifactId>

<version>0.11.5</version>

<scope>runtime</scope>

</dependency>

**SpringLearnApplication.java:**

package com.cognizant.springlearn;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringLearnApplication {

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

public static void main(String[] args) {

LOGGER.info("START");

SpringApplication.run(SpringLearnApplication.class, args);

LOGGER.info("END");

}

}

**Step 2:**

* Create SecurityConfig.java in the src/main/java/com/cognizant/springlearn/security directory,AuthenticationController.java in the src/main/java/com/cognizant/springlearn/controller directory, and alter application.properties.

**SecurityConfig.java**:

package com.cognizant.springlearn.security;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.Customizer;

import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

@EnableWebSecurity

@EnableMethodSecurity

public class SecurityConfig {

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

@Bean

public PasswordEncoder passwordEncoder() {

LOGGER.info("Creating password encoder");

return new BCryptPasswordEncoder();

}

@Bean

public UserDetailsService userDetailsService(PasswordEncoder passwordEncoder) {

LOGGER.info("Setting up in-memory users");

InMemoryUserDetailsManager manager = new InMemoryUserDetailsManager();

manager.createUser(User.withUsername("admin")

.password(passwordEncoder.encode("pwd"))

.roles("ADMIN")

.build());

manager.createUser(User.withUsername("user")

.password(passwordEncoder.encode("pwd"))

.roles("USER")

.build());

return manager;

}

@Bean

public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {

LOGGER.info("Configuring HTTP security");

return http

.csrf(csrf -> csrf.disable())

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").hasAnyRole("ADMIN", "USER")

.anyRequest().authenticated()

)

.httpBasic(Customizer.withDefaults())

.build();

}

}

**AuthenticationController.java**:

package com.cognizant.springlearn.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.security.Keys;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

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

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

import java.security.Key;

import java.util.Base64;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

@RestController

public class AuthenticationController {

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

private static final String SECRET\_KEY = "12345678901234567890123456789012";

@GetMapping("/authenticate")

public Map<String, String> authenticate(@RequestHeader("Authorization") String authHeader) {

LOGGER.info("START");

LOGGER.debug("Authorization Header: {}", authHeader);

String user = extractUser(authHeader);

String token = generateJwt(user);

Map<String, String> response = new HashMap<>();

response.put("token", token);

LOGGER.info("END");

return response;

}

private String extractUser(String authHeader) {

LOGGER.info("Start extractUser()");

String encoded = authHeader.substring(6);

byte[] decodedBytes = Base64.getDecoder().decode(encoded);

String decodedString = new String(decodedBytes);

String user = decodedString.split(":")[0];

LOGGER.debug("Decoded User: {}", user);

return user;

}

private String generateJwt(String user) {

LOGGER.info("Start generateJwt()");

Key key = Keys.hmacShaKeyFor(SECRET\_KEY.getBytes());

String token = Jwts.builder()

.setSubject(user)

.setIssuedAt(new Date())

.setExpiration(new Date(System.currentTimeMillis() + 20 \* 60 \* 1000))

.signWith(key)

.compact();

LOGGER.debug("Generated Token: {}", token);

return token;

}

}

**application.properties**:

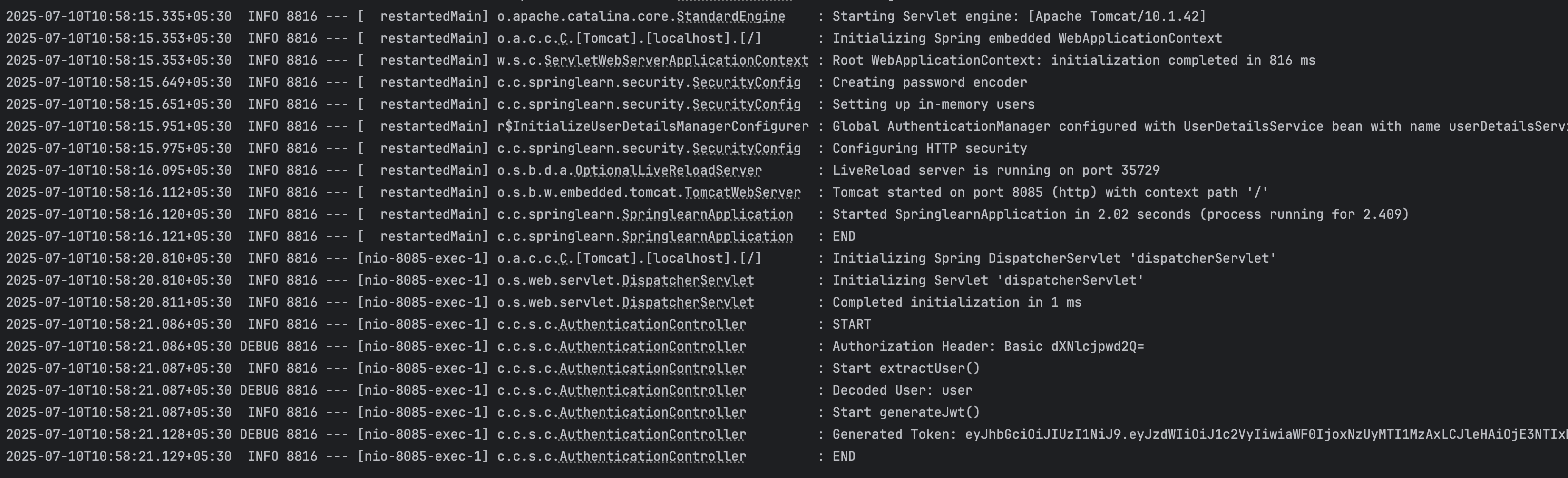
server.port=8085

logging.level.org.springframework=INFO

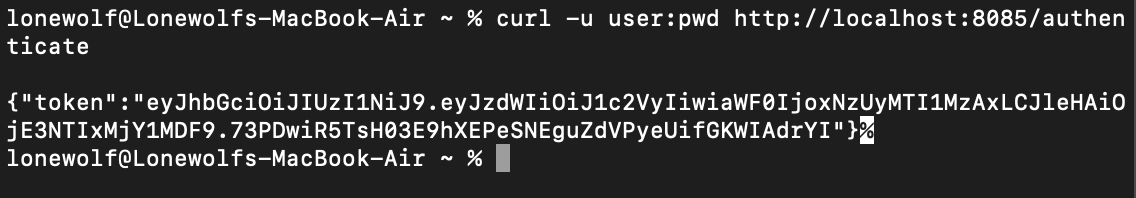
logging.level.com.cognizant.springlearn=DEBUG

**Step 3:**

Run **SpringLearnApplication** to start the server.



Run curl -u user:pwd <http://localhost:8085/authenticate> in command prompt/terminal.



**Additional important hands-on**

**Spring-REST-handson**

**Step 1:**

Configure spring.io based on beloe details,

|  |  |
| --- | --- |
| **Field** | **Value** |
| **Project** | Maven |
| **Language** | Java |
| **Spring Boot** | 3.5.3 |
| **Group** | com.example |
| **Artifact** | springresthandson |
| **Name** | springresthandson |
| **Description** | Spring Boot REST Handson Project |
| **Package Name** | com.example.springresthandson |
| **Packaging** | Jar |
| **Java Version** | 24 |
| **Dependencies** | - Spring Web - |

**Step 2:**

Configure **SpringRestHandsonApplication.java** and **pom.xml:**

**SpringRestHandsonApplication.java:**

package com.example.springresthandson;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringRestHandsonApplication {

public static void main(String[] args) {

SpringApplication.run(SpringRestHandsonApplication.class, args);

}

}

**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>

<parent>

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

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

<version>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>spring-rest-handson</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>spring-rest-handson</name>

<description>Spring Boot REST Handson Project</description>

<properties>

<java.version>24</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>

**Step 2:**

Create **EmployeeController.java** in the **controller** package,**EmployeeDao.java** in the **dao** package,**EmployeeService.java** in the **service** package,**employee.xml** in the **resource** directory,**Skill.java** and **Employee.java** in the **model** package.

**EmployeeController.java**:

package com.example.springresthandson.controller;

import com.example.springresthandson.model.Employee;

import com.example.springresthandson.service.EmployeeService;

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

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

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

@RestController

@RequestMapping("/employee")

public class EmployeeController {

private final EmployeeService employeeService;

public EmployeeController(EmployeeService employeeService) {

this.employeeService = employeeService;

}

@GetMapping

public Employee getEmployee() {

return employeeService.getEmployee();

}

}

**EmployeeDao.java**:

package com.example.springresthandson.dao;

import com.example.springresthandson.model.Employee;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.stereotype.Repository;

@Repository

public class EmployeeDao {

private final Employee employee;

public EmployeeDao() {

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

this.employee = (Employee) context.getBean("employee");

}

public Employee getEmployee() {

return employee;

}

}

**employee.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

https://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="skillJava" class="com.example.springresthandson.model.Skill">

<property name="id" value="1"/>

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

</bean>

<bean id="skillSpring" class="com.example.springresthandson.model.Skill">

<property name="id" value="2"/>

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

</bean>

<bean id="employee" class="com.example.springresthandson.model.Employee">

<property name="id" value="101"/>

<property name="name" value="John Doe"/>

<property name="salary" value="50000"/>

<property name="skills">

<list>

<ref bean="skillJava"/>

<ref bean="skillSpring"/>

</list>

</property>

</bean>

</beans>

**EmployeeService.java**:

package com.example.springresthandson.service;

import com.example.springresthandson.model.Employee;

import com.example.springresthandson.dao.EmployeeDao;

import org.springframework.stereotype.Service;

@Service

public class EmployeeService {

private final EmployeeDao employeeDao;

public EmployeeService(EmployeeDao employeeDao) {

this.employeeDao = employeeDao;

}

public Employee getEmployee() {

return employeeDao.getEmployee();

}

}

**Skill.java:**

package com.example.springresthandson.model;

import java.io.Serializable;

public class Skill implements Serializable {

private String id;

private String name;

public Skill() {

}

public Skill(String id, String name) {

this.id = id;

this.name = name;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Skill{" +

"id='" + id + '\'' +

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

'}';

}

}

**Employee.java:**

package com.example.springresthandson.model;

import java.io.Serializable;

import java.util.List;

public class Employee implements Serializable {

private String id;

private String name;

private double salary;

private List<Skill> skills;

public Employee() {

}

public Employee(String id, String name, double salary, List<Skill> skills) {

this.id = id;

this.name = name;

this.salary = salary;

this.skills = skills;

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public List<Skill> getSkills() {

return skills;

}

public void setSkills(List<Skill> skills) {

this.skills = skills;

}

@Override

public String toString() {

return "Employee{" +

"id='" + id + '\'' +

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

", salary=" + salary +

", skills=" + skills +

'}';

}

}

**Step 3:**

Run **SpringRestHandsonApllication.java** to start the server.And run <http://localhost:8080/employee> on browser.

