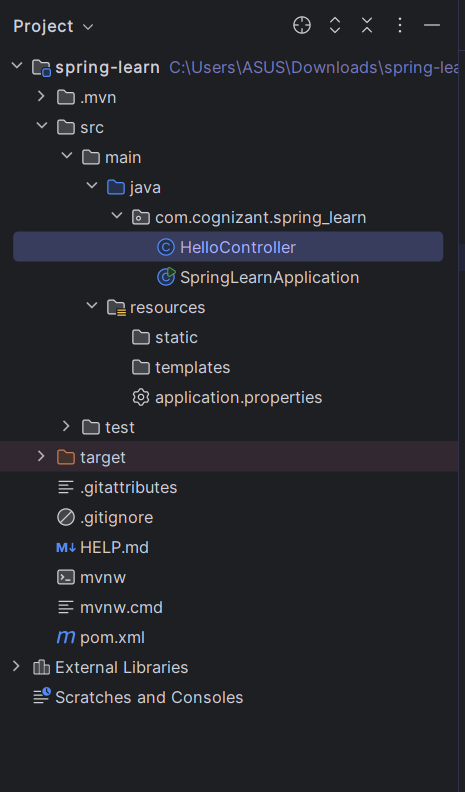
**COGNIZANT DIGI NURTURE 4.0**

**WEEK 4 MANDATORY HANDS ON**

**Spring Rest:**

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

Project Structure:



1. Generate the Project Using Spring Initializr:

* **Project:** Maven
* **Language:** Java
* **Spring Boot Version:** 3.5.3
* **Group:** com.cognizant
* **Artifact:** spring-learn
* **Name:** spring-learn
* **Java Version:** 21

1. Dependencies

* **Spring Web**
* **Spring Boot DevTools**

1. Create the Main Application Class

src/main/java/com/cognizant/springlearn/SpringLearnApplication.java

**SpringLearnApplication.java**

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) {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 }  
}

1. Create a REST Controller

src/main/java/com/cognizant/springlearn/HelloController.java

**HelloController.java**

package com.cognizant.spring\_learn;  
  
 import org.springframework.web.bind.annotation.GetMapping;  
 import org.springframework.web.bind.annotation.RestController;  
  
 @RestController  
  
 public class HelloController {  
  
 @GetMapping("/hello")  
 public String sayHello() {  
 return "Hello from Spring Boot!";  
 }  
 }

1. **Run the Application**

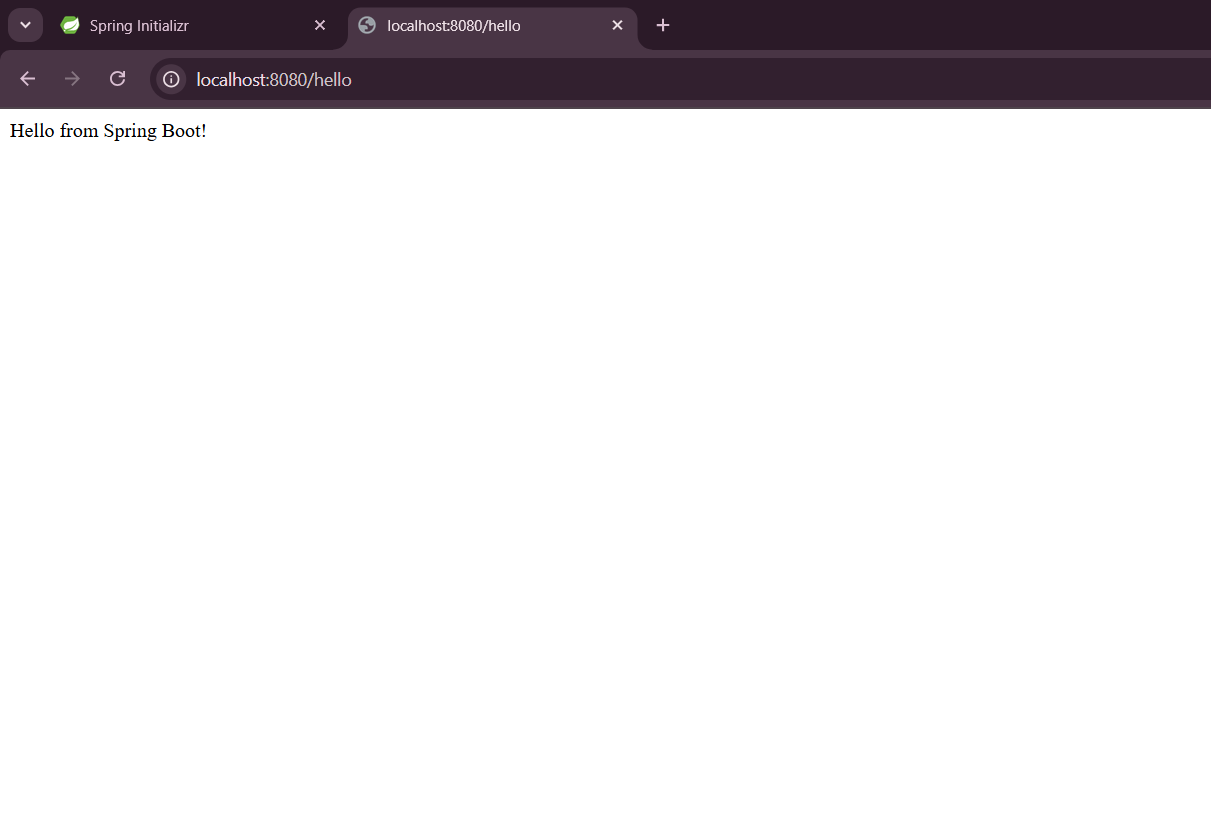
* Open the web browser and paste the url

http://localhost:8080/hello

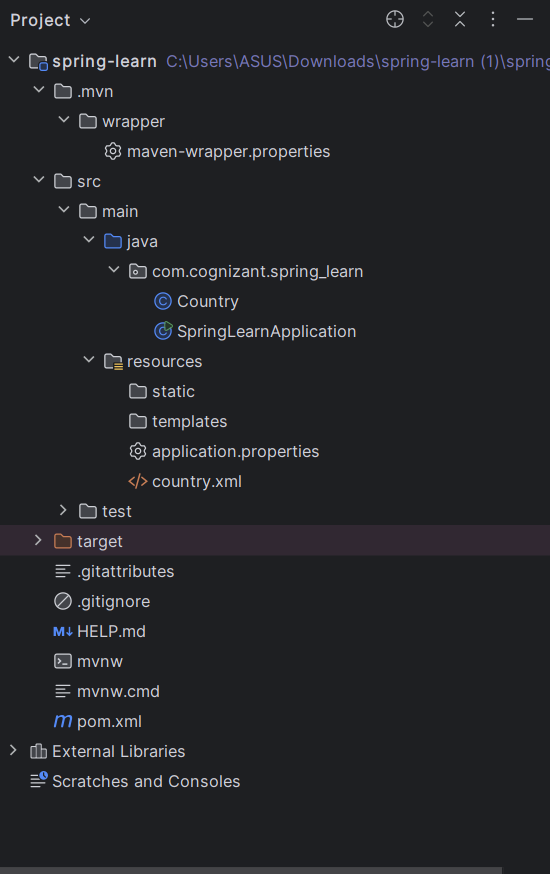
**Pom.xml:**

<?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/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Demo project for Spring Boot</description>  
  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
  
 <dependencies>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </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>  
 <version>3.5.3</version>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

**OUTPUT:**



Exersice 2:**Spring Core – Load Country from Spring Configuration XML**

**Project Structure:**  
 

**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.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Demo project for Spring Boot</description>  
  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
  
 <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>  
  
 <!-- SLF4J Logging -->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>2.0.13</version>  
 </dependency>  
  
 <!-- Testing -->  
 <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>

**Java Class:**

**Country.java:**

package com.cognizant.spring\_learn;  
  
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 + "]";  
 }  
}

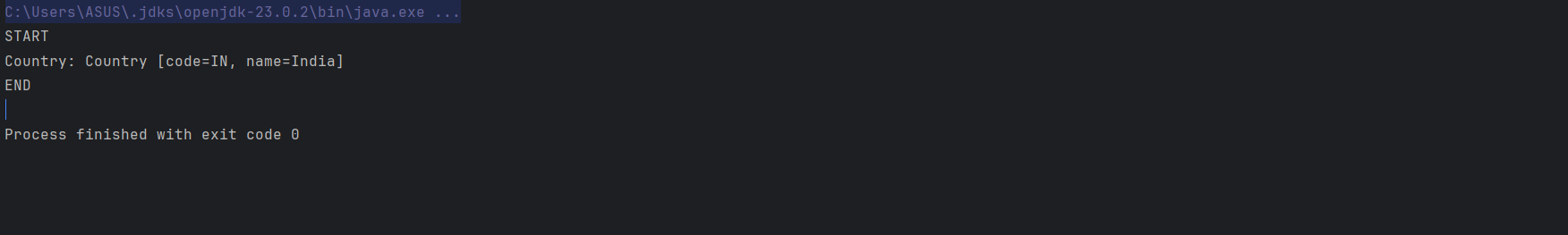
**SpringLearnApplication:**

package com.cognizant.spring\_learn;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 System.out.println("START");  
 displayCountry();  
 System.out.println("END");  
 }  
  
 public static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 System.out.println("Country: " + country.toString());  
 }  
}

Country.xml:

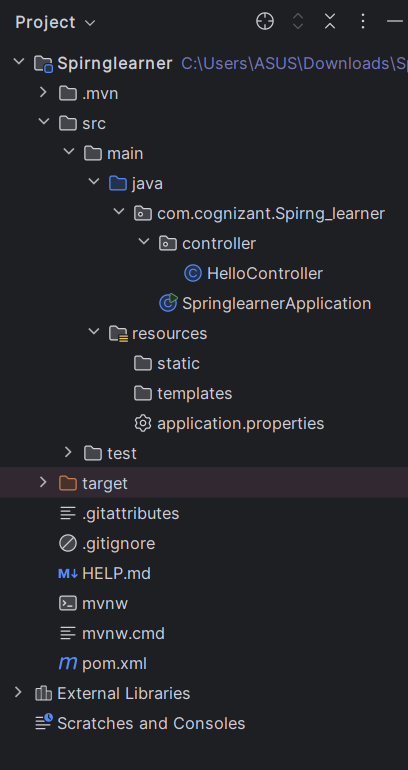
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="  
 http://www.springframework.org/schema/beans  
 https://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="country" class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
</beans>

**OUTPUT:**



Exersice 3: **Helloworld RESTFUL webservice**

**Project Structure:**



**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.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Hello World RESTful Web Service Project</description>  
  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web Starter -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Spring Boot DevTools (for hot reload during development) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <!-- Testing (optional) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- Spring Boot Maven Plugin -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <version>3.5.3</version>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Java Class:**

**HelloController.java**

package com.cognizant.Spirng\_learner.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.debug("START");  
 LOGGER.debug("END");  
 return "Hello World!!";  
 }  
}

**SpringlearnerApplication.java**

package com.cognizant.Spirng\_learner;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringlearnerApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringlearnerApplication.class, args);  
 }  
}

**Application.properties:**

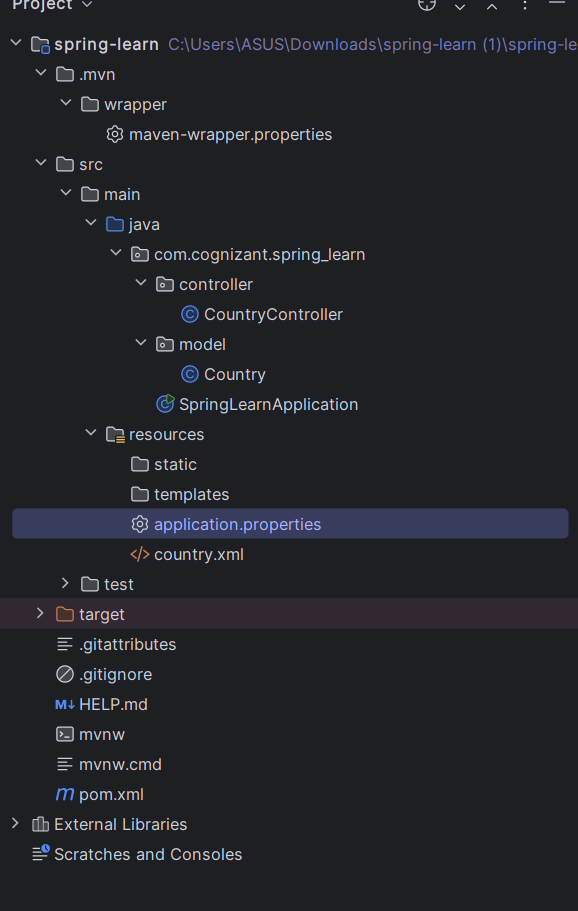
spring.application.name=Springlearner  
 server.port=8083

**OUTPUT:**



Exersice 4 :

**PROJECT STRUCTURE**



**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.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>spring-learn</name>  
 <description>Demo project for Spring Boot</description>  
  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
  
 <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>  
  
 <!-- SLF4J Logging -->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>2.0.13</version>  
 </dependency>  
  
 <!-- Testing -->  
 <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>

**Java class:**

Package : controller

Class :CountryController

**CountryController.java**

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.model.Country;  
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 {  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("in");  
 return country;  
 }  
}

Package : model

Class : Country

**Country.java:**

package com.cognizant.spring\_learn.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("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 + "]";  
 }  
}

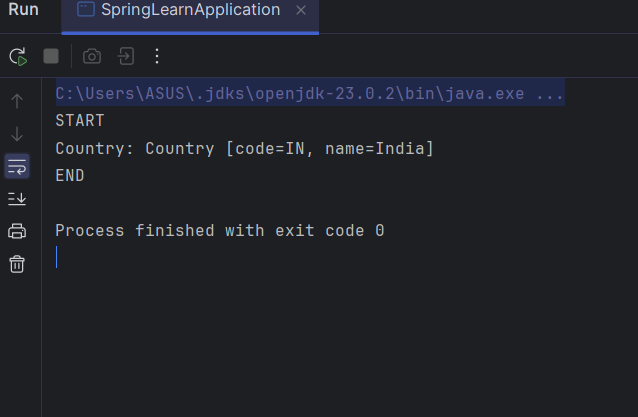
SpringLearnApplication.java

package com.cognizant.spring\_learn;  
  
import com.cognizant.spring\_learn.model.Country;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 System.out.println("START");  
 displayCountry();  
 System.out.println("END");  
 }  
  
 public static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("in");  
  
 System.out.println("Country: " + country.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="in" class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
</beans>

**OUTPUT:**



Excersice 5: **REST - Get country based on country code**

**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="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <bean class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
 <bean class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
 <bean class="com.cognizant.spring\_learn.model.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
 </list>  
 </constructor-arg>  
 </bean>  
</beans>

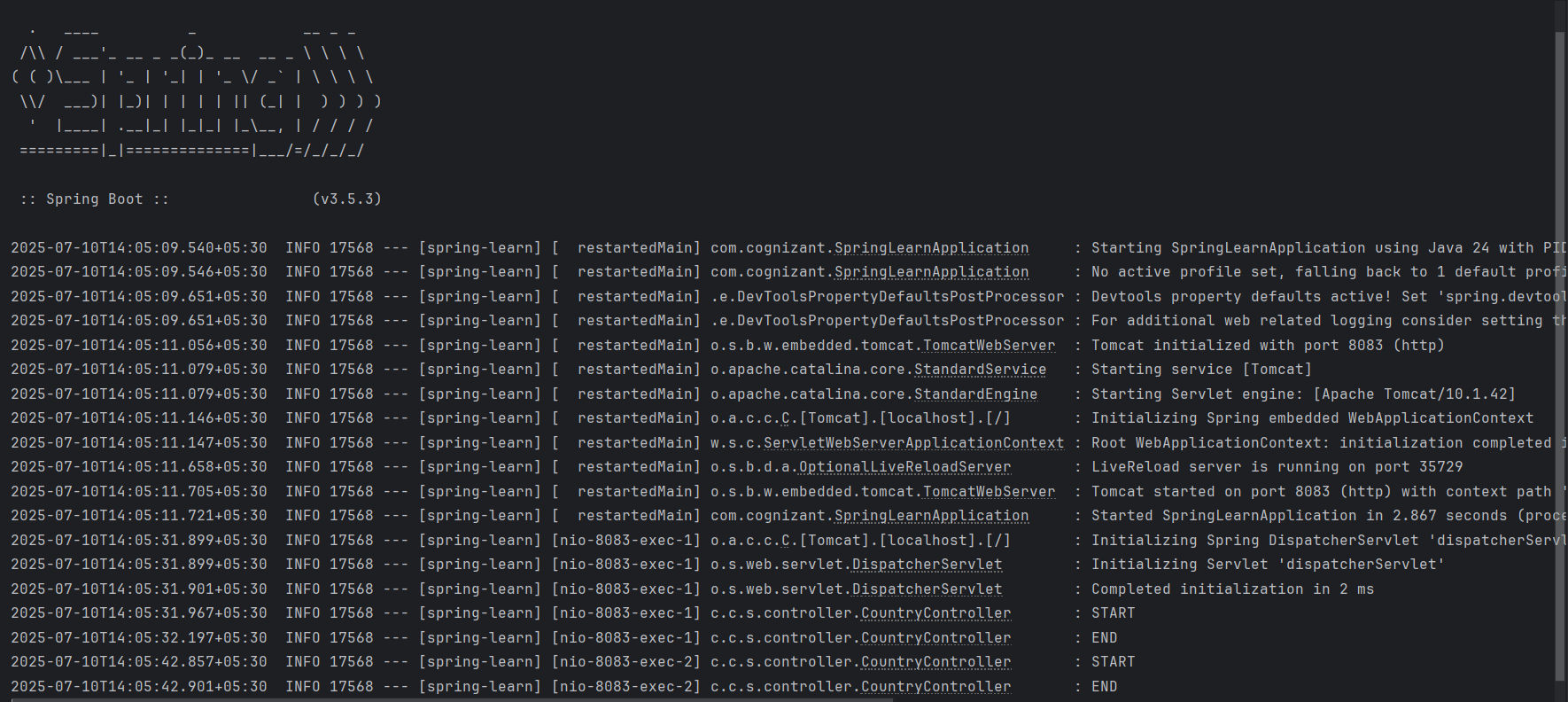
**CountryService.java:**

package com.cognizant.spring\_learn.service;  
  
import com.cognizant.spring\_learn.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> countryList = context.getBean("countryList", List.class);  
  
 return countryList.stream()  
 .filter(country -> country.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);  
 }  
}

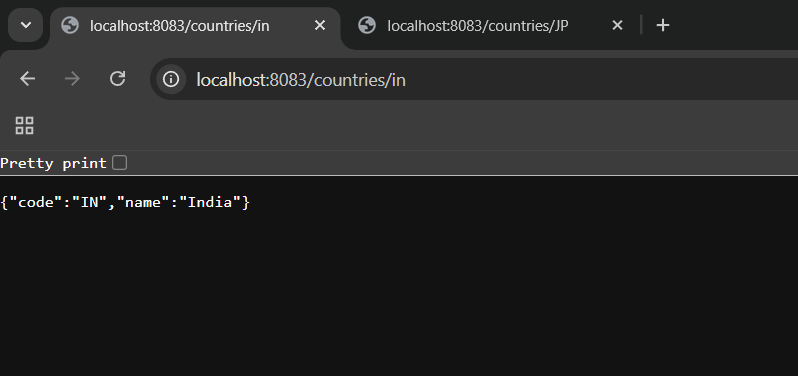
**CountryController.java:**

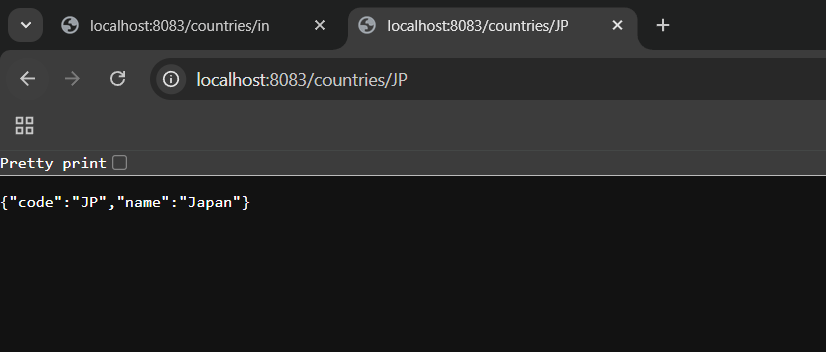
package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.model.Country;  
import com.cognizant.spring\_learn.service.CountryService;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
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 = context.getBean("in", Country.class);  
 *LOGGER*.info("END");  
 return country;  
 }  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 *LOGGER*.info("START");  
 Country country = countryService.getCountry(code);  
 *LOGGER*.info("END");  
 return country;  
 }  
}.

**Output:**

****

In browser:





Exersice 6: **Create authentication service that returns JWT**

**Program:**

* Spring Initializer setup.
* **Project**: Maven
* **Language**: Java
* **Spring Boot**: 3.1+ or 3.5.3
* **Group**: com.cognizant
* **Artifact**: jwt-auth
* **Name**: jwt-auth
* **Package name**: com.cognizant.jwt\_auth
* **Packaging**: Jar
* **Java Version**: 17 or 21
* Update the pom.xml

**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/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>jwt-auth</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>jwt-auth</name>  
 <description>JWT Auth Project</description>  
  
 <properties>  
 <java.version>24</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web and Security -->  
 <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>  
  
 <!-- JSON Web Token - JJWT -->  
 <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>  
  
 <!-- Servlet API -->  
 <dependency>  
 <groupId>jakarta.servlet</groupId>  
 <artifactId>jakarta.servlet-api</artifactId>  
 <version>6.0.0</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!-- Development and Testing -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-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>

* Create package named jwt\_auth and config,controller
* Path src/main/java/com/cognizant/jwt\_auth/controller, src/main/java/com/cognizant/jwt\_auth/config.
* Then create a class named AuthenticationController.java in src/main/java/com/cognizant/jwt\_auth/controller/AuthenticationController.java.

**AuthenticationController.java:**

package com.cognizant.jwt\_auth.controller;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import io.jsonwebtoken.security.Keys;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
import jakarta.servlet.http.HttpServletRequest;  
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 final Key secretKey = Keys.*hmacShaKeyFor*("ThisIsASecretKeyThatIsAtLeast32Chars!".getBytes());  
  
 @GetMapping("/authenticate")  
 public Map<String, String> authenticate(HttpServletRequest request) {  
 String authHeader = request.getHeader("Authorization");  
  
 if (authHeader == null || !authHeader.startsWith("Basic ")) {  
 throw new RuntimeException("Missing or invalid Authorization header");  
 }  
  
 String base64Credentials = authHeader.substring("Basic ".length());  
 String credentials = new String(Base64.*getDecoder*().decode(base64Credentials));  
 String[] values = credentials.split(":", 2);  
  
 String username = values[0];  
 String password = values[1];  
  
 if (!username.equals("user") || !password.equals("pwd")) {  
 throw new RuntimeException("Invalid credentials");  
 }  
  
 String token = Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.*currentTimeMillis*() + 600000)) // 10 minutes  
 .signWith(secretKey, SignatureAlgorithm.*HS256*)  
 .compact();  
  
 Map<String, String> response = new HashMap<>();  
 response.put("token", token);  
 return response;  
 }  
}

**SecurityConfig.java:**

package com.cognizant.jwt\_auth.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetails;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
@EnableWebSecurity  
public class SecurityConfig {  
  
 @Bean  
 public UserDetailsService userDetailsService() {  
 UserDetails user = User  
 .*withUsername*("user")  
 .password("{noop}pwd") // {noop} means no encoding  
 .roles("USER")  
 .build();  
 return new InMemoryUserDetailsManager(user);  
 }  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/authenticate").authenticated()  
 .anyRequest().permitAll()  
 )  
 .httpBasic(); // enables basic auth  
 return http.build();  
 }  
}

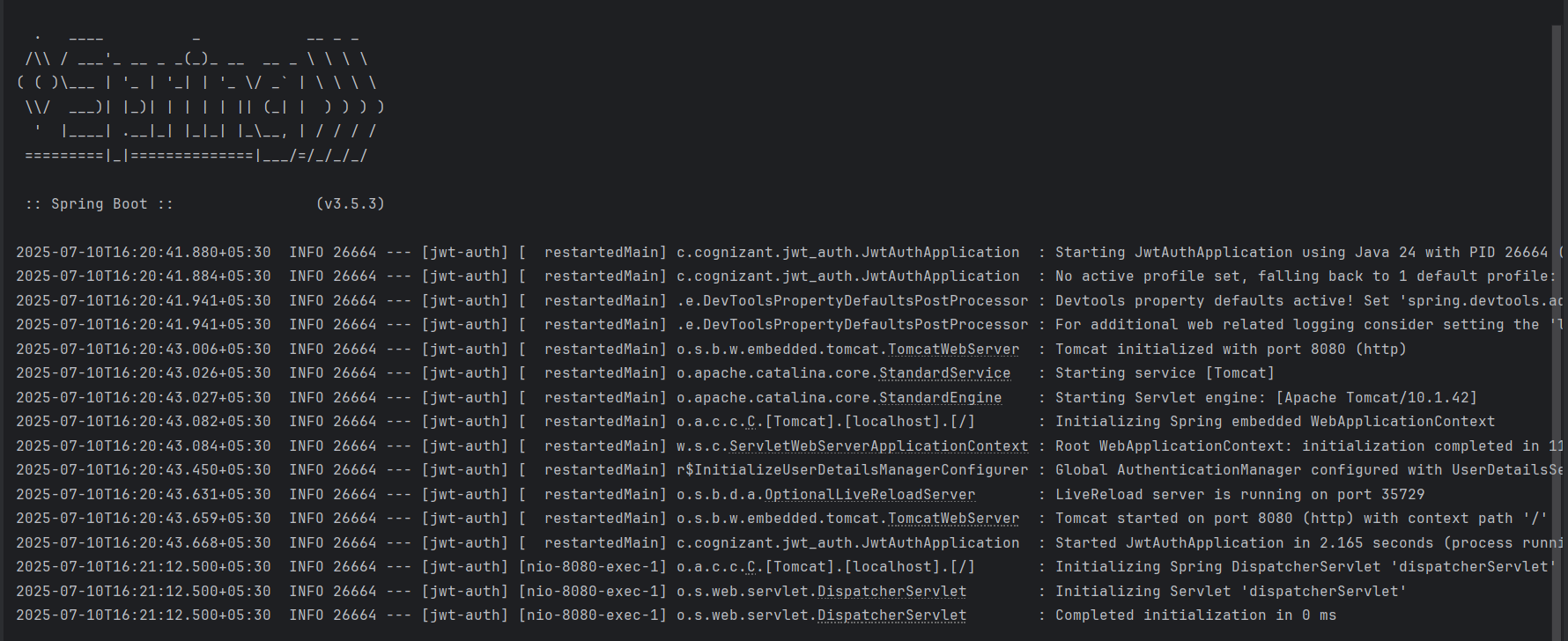
**Application.properties:**

spring.application.name=jwt-auth  
server.port=8080

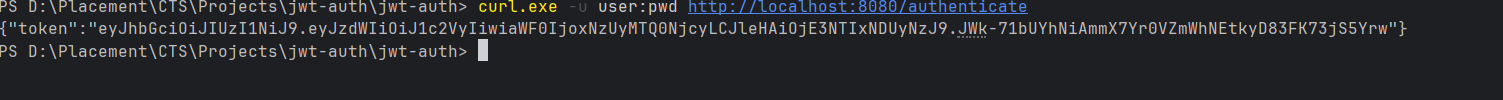
**JwtAuthApplication.java:**

package com.cognizant.jwt\_auth;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class JwtAuthApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(JwtAuthApplication.class, args);  
 }  
}

**Output:**

****

**The terminal token generation:**

****

**Additional important hands-on**

Exersice 7:**Demonstrate integration of RESTful Web Service of type GET and test the service using postman**.

**Program:**

* Configure the application.properties.

**Application.properties:**

spring.application.name=jwt-auth  
server.port=8080  
spring.security.user.name=  
spring.security.user.password=  
spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration

* Update the pom.xml

**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/>  
 </parent>  
  
 <groupId>com.cognizant</groupId>  
 <artifactId>jwt-auth</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>jwt-auth</name>  
 <description>JWT Auth Project</description>  
  
 <properties>  
 <java.version>24</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web and Security -->  
 <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>  
  
 <!-- JSON Web Token - JJWT -->  
 <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>  
  
 <!-- Servlet API -->  
 <dependency>  
 <groupId>jakarta.servlet</groupId>  
 <artifactId>jakarta.servlet-api</artifactId>  
 <version>6.0.0</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!-- Development and Testing -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.security</groupId>  
 <artifactId>spring-security-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-tx</artifactId>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Employee.xml:**

package com.cognizant.springrest.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private double salary;  
 private Department department;  
  
 public Employee() {}  
  
 public Employee(int id, String name, double salary, Department department) {  
 this.id = id;  
 this.name = name;  
 this.salary = salary;  
 this.department = department;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int 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 Department getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(Department department) {  
 this.department = department;  
 }  
}

**Employee.java:**

package com.cognizant.springrest.model;  
  
public class Employee {  
 private int id;  
 private String name;  
 private double salary;  
 private Department department;  
  
 public Employee() {}  
  
 public Employee(int id, String name, double salary, Department department) {  
 this.id = id;  
 this.name = name;  
 this.salary = salary;  
 this.department = department;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int 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 Department getDepartment() {  
 return department;  
 }  
  
 public void setDepartment(Department department) {  
 this.department = department;  
 }  
}

**Department.java:**

package com.cognizant.springrest.model;  
  
public class Department {  
 private int id;  
 private String name;  
  
 public Department() {}  
  
 public Department(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

**Skill.java:**

package com.cognizant.springrest.model;  
  
public class Skill {  
 private int id;  
 private String name;  
  
 public Skill() {}  
  
 public Skill(int id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

**EmployeeController.java:**

package com.cognizant.springrest.controller;  
  
import com.cognizant.springrest.model.Employee;  
import com.cognizant.springrest.service.EmployeeService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 @GetMapping("/employees")  
 public List<Employee> getAllEmployees() {  
 return employeeService.getAllEmployees();  
 }  
}

**DepartmentCountroller.java:**

package com.cognizant.springrest.controller;  
  
import com.cognizant.springrest.model.Department;  
import com.cognizant.springrest.service.DepartmentService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.List;  
  
@RestController  
public class DepartmentController {  
  
 @Autowired  
 private DepartmentService departmentService;  
  
 @GetMapping("/departments")  
 public List<Department> getAllDepartments() {  
 return departmentService.getAllDepartments();  
 }  
}

**Employeedao.java:**

package com.cognizant.springrest.dao;  
  
import com.cognizant.springrest.model.Employee;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Repository;  
  
import java.util.Arrays;  
import java.util.List;  
  
@Repository  
public class EmployeeDao {  
  
 private static List<Employee> *EMPLOYEE\_LIST*;  
  
 public EmployeeDao() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");  
 Employee emp1 = context.getBean("employee1", Employee.class);  
 Employee emp2 = context.getBean("employee2", Employee.class);  
 *EMPLOYEE\_LIST* = Arrays.*asList*(emp1, emp2);  
 }  
  
 public List<Employee> getAllEmployees() {  
 return *EMPLOYEE\_LIST*;  
 }  
}

**Departmentdao.java:**

package com.cognizant.springrest.dao;  
  
import com.cognizant.springrest.model.Department;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Repository;  
  
import java.util.Arrays;  
import java.util.List;  
  
@Repository  
public class DepartmentDao {  
  
 private static List<Department> *DEPARTMENT\_LIST*;  
  
 public DepartmentDao() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("employee.xml");  
 Department dept1 = context.getBean("dept1", Department.class);  
 Department dept2 = context.getBean("dept2", Department.class);  
 *DEPARTMENT\_LIST* = Arrays.*asList*(dept1, dept2);  
 }  
  
 public List<Department> getAllDepartments() {  
 return *DEPARTMENT\_LIST*;  
 }  
}

**Employeeservice.java:**

package com.cognizant.springrest.service;  
  
import com.cognizant.springrest.dao.EmployeeDao;  
import com.cognizant.springrest.model.Employee;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
import java.util.List;  
  
@Service  
public class EmployeeService {  
  
 @Autowired  
 private EmployeeDao employeeDao;  
  
 @Transactional  
 public List<Employee> getAllEmployees() {  
 return employeeDao.getAllEmployees();  
 }  
}

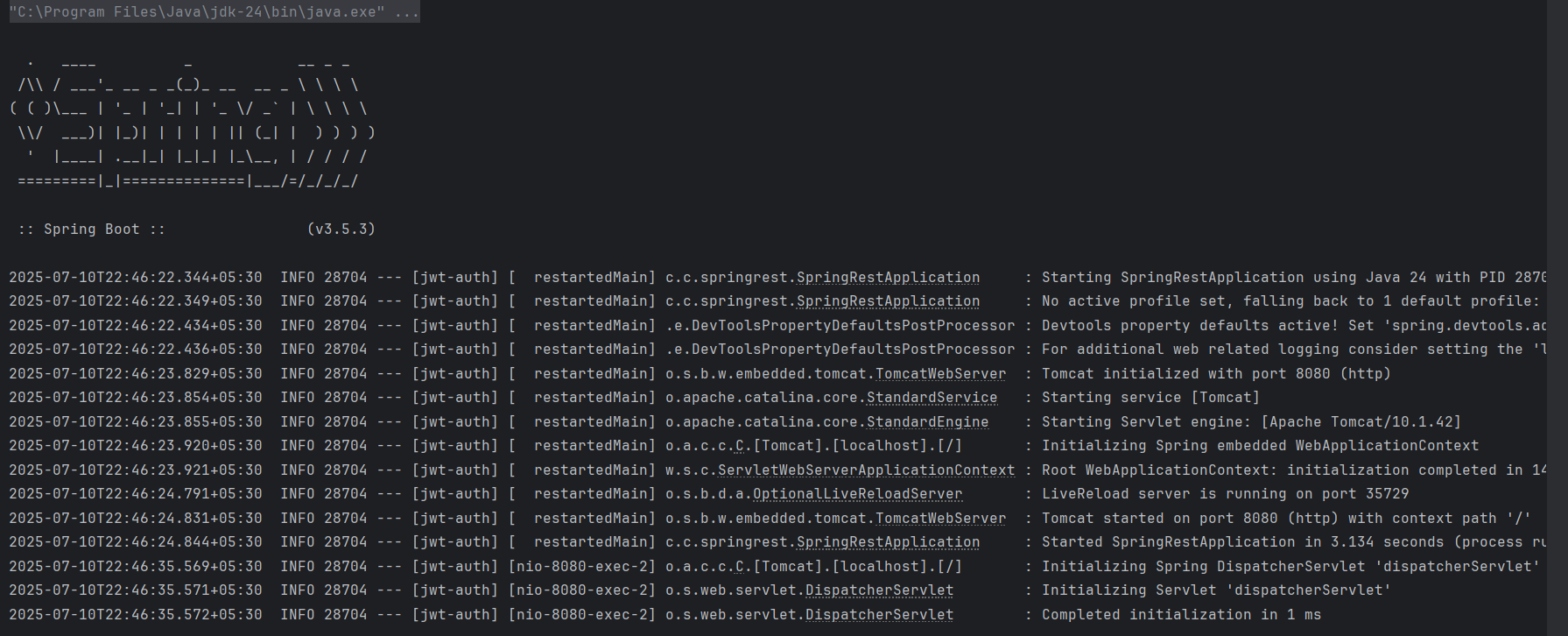
**Departmentservice.java:**

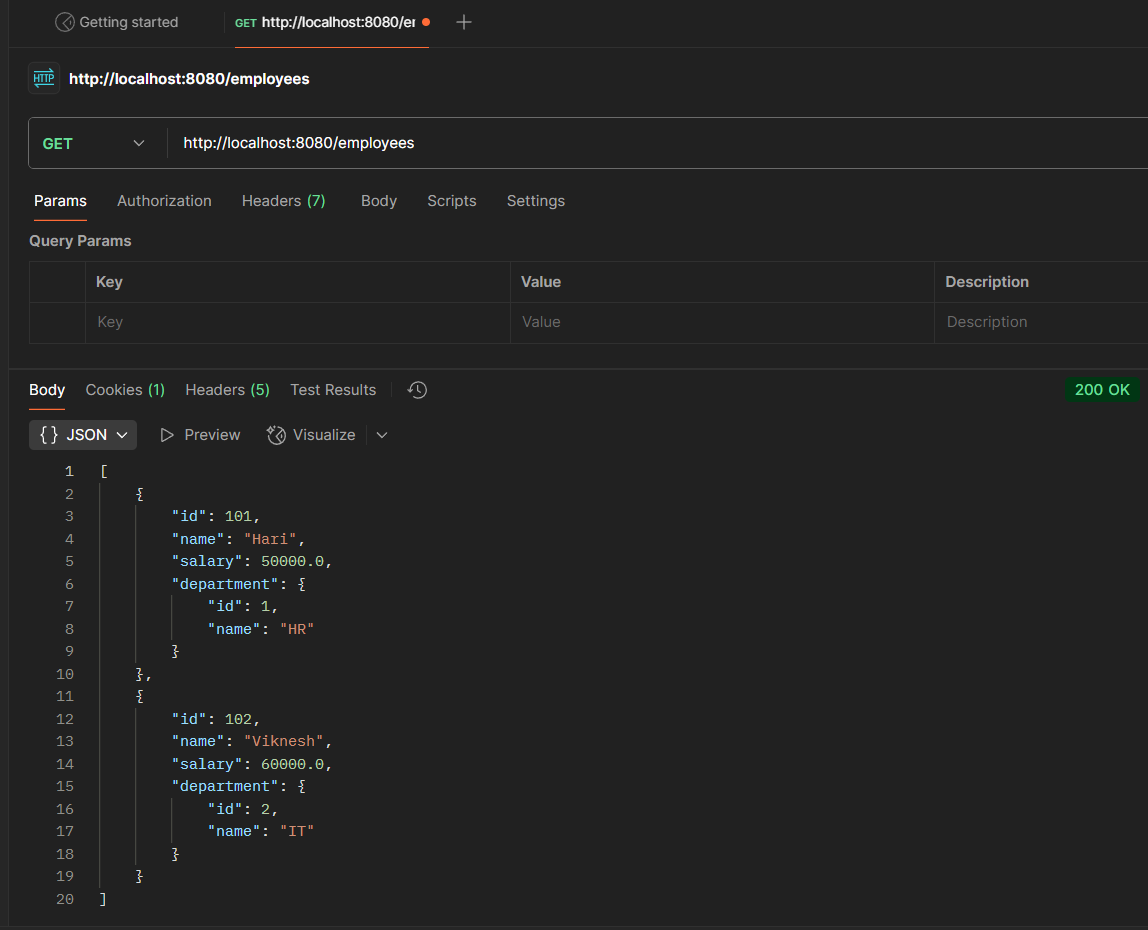
package com.cognizant.springrest.service;  
  
import com.cognizant.springrest.dao.DepartmentDao;  
import com.cognizant.springrest.model.Department;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import java.util.List;  
  
@Service  
public class DepartmentService {  
  
 @Autowired  
 private DepartmentDao departmentDao;  
  
 public List<Department> getAllDepartments() {  
 return departmentDao.getAllDepartments();  
 }  
}

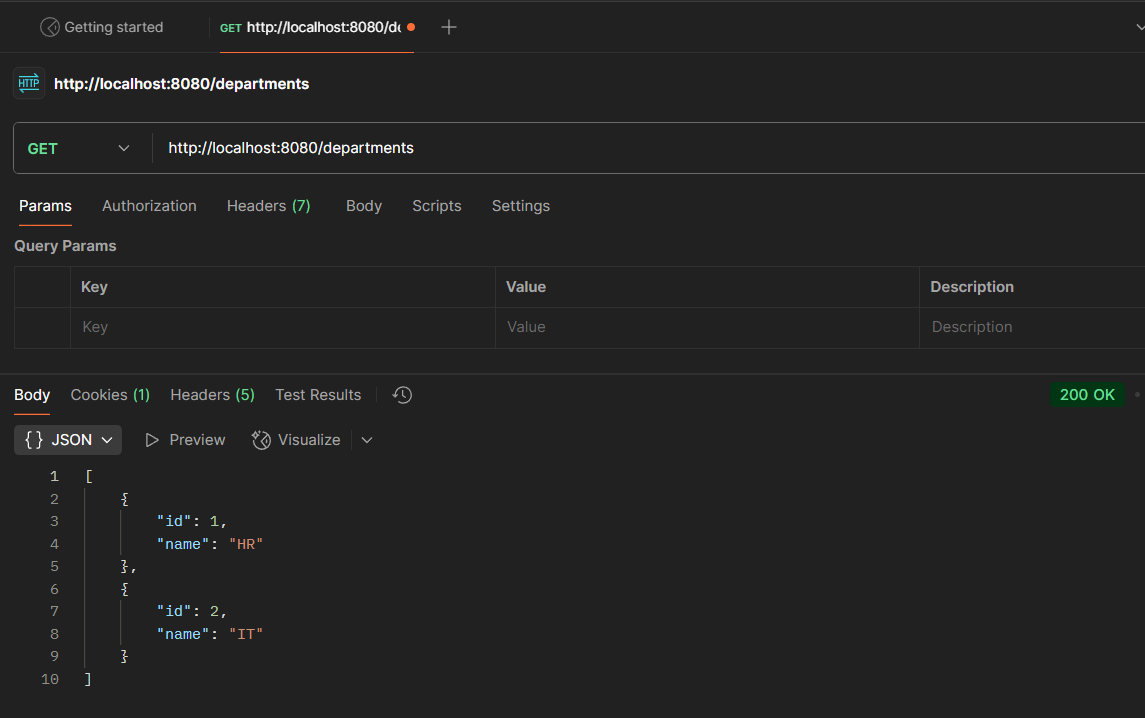
**SpringRestApplication.java:**

package com.cognizant.springrest;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringRestApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringRestApplication.class, args);  
 }  
}

**Output:**

****

**In postman -to get all employees:**

**To get all departments: **