**Mandatory Handson – 01**

**Create a Spring Web Project using Maven**

**SpringLearnApplication.java**

*package com.cognizant.spring\_learn;*

*import org.springframework.boot.SpringApplication;*

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

*@SpringBootApplication*

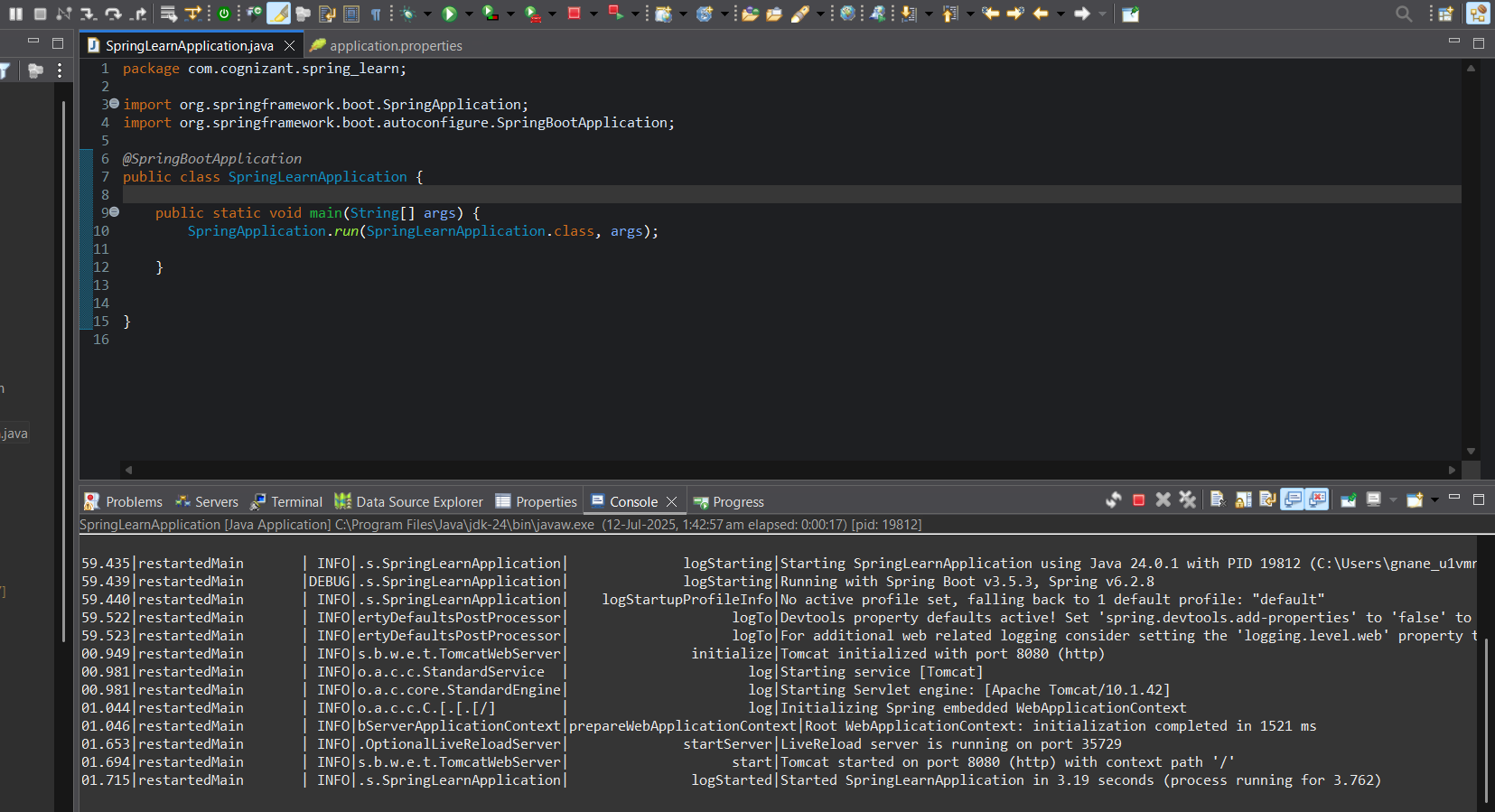
*public class SpringLearnApplication {*

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

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

*}*

*}*

****

**Mandatory Handson – 02**

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

**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****>*

**SpringLearnApplication.java**

*package com.cognizant.spring\_learn;*

*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) {*

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

*displayCountry();*

*}*

*public static 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");*

*}*

*}*

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("Getting code");

return code;

}

public void setCode(String code) {

***LOGGER***.debug("Setting code: {}", code);

this.code = code;

}

public String getName() {

***LOGGER***.debug("Getting name");

return name;

}

public void setName(String name) {

***LOGGER***.debug("Setting name: {}", name);

this.name = name;

}

*@Override*

public String toString() {

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

}

}

application.properties

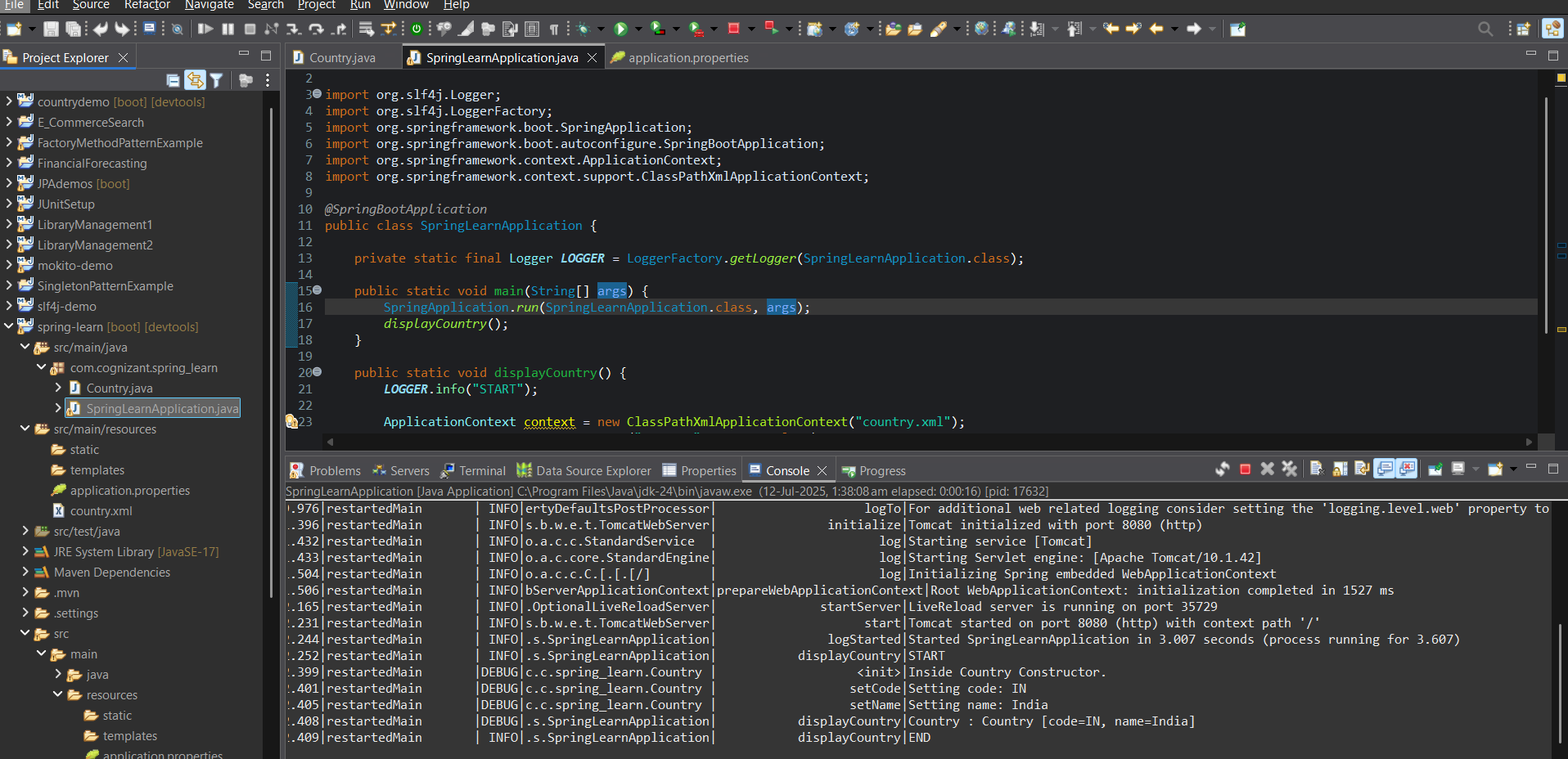
*spring.application.name=spring-learn*

*logging.level.org.springframework=info*

*logging.level.com.cognizant.spring\_learn=debug*

*logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n*

**Output:**



**Mandatory Handson – 03**

**Hello World RESTful Web Service**

HelloController.java

*package com.cognizant.spring\_learn.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 msg ="Hello World!!";*

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

*return msg; }*

*}*

**application.properties**

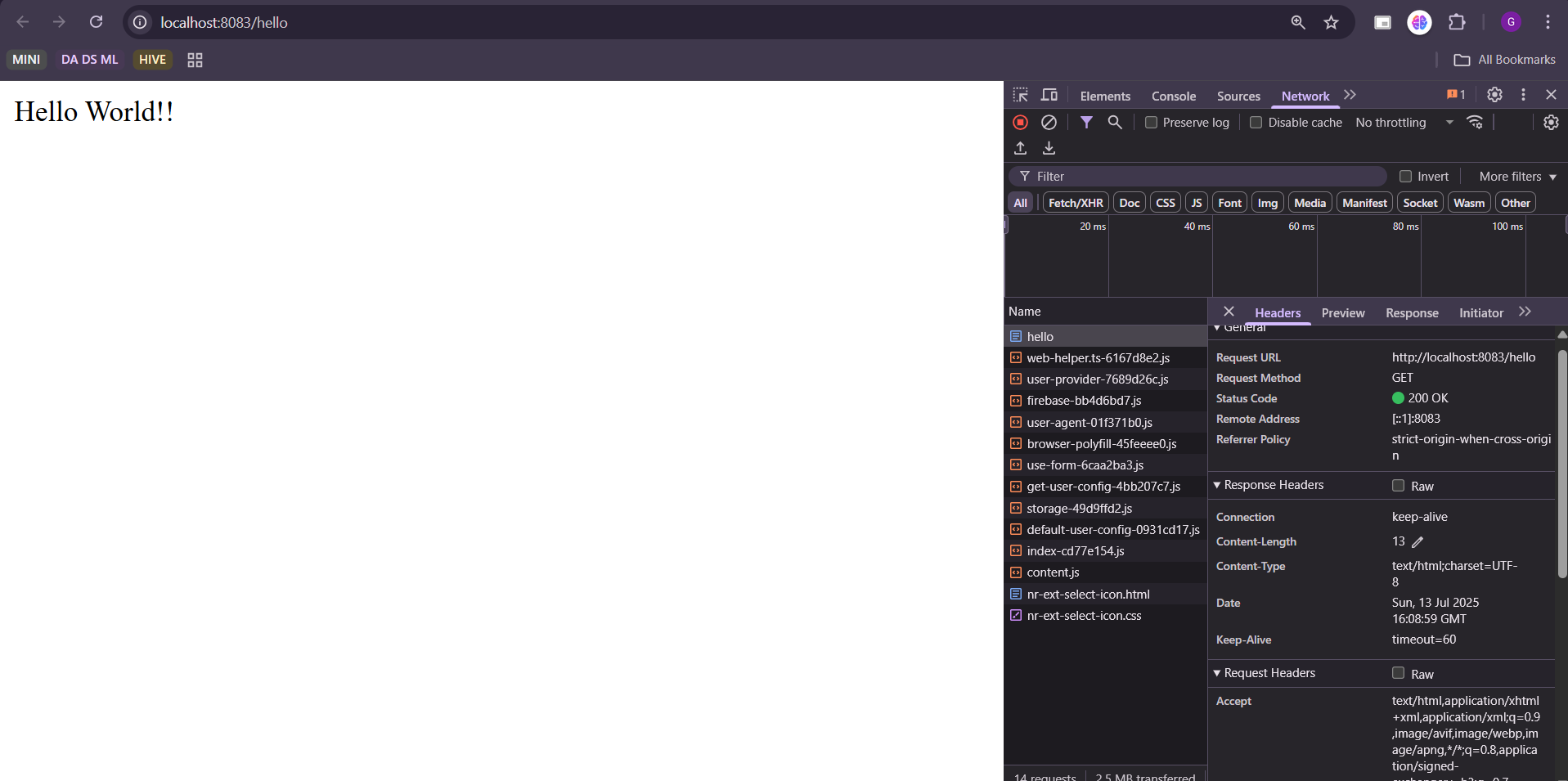
*spring.application.name=spring-learn*

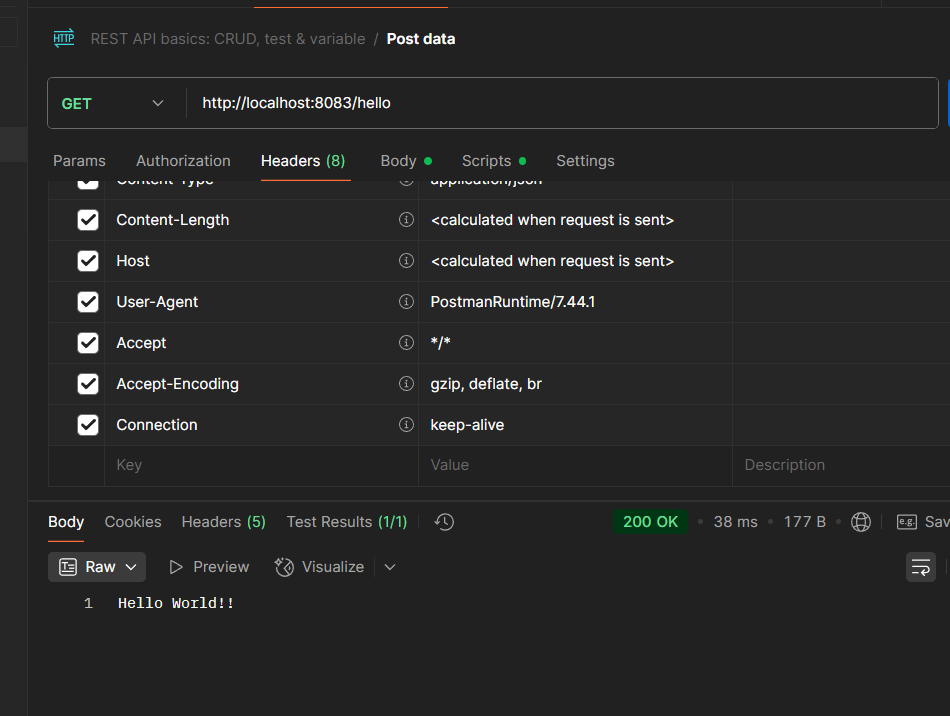
*logging.level.org.springframework=info*

*logging.level.com.cognizant.spring\_learn=debug*

*logging.pattern.console=%d{yyMMdd}|%d{HH:mm:ss.SSS}|%-20.20thread|%5p|%-25.25logger{25}|%25M|%m%n*

*server.port=8083*

****



**Mandatory Handson – 04**

**REST - Country Web Service**

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("Getting code");*

*return code;*

*}*

*public void setCode(String code) {*

***LOGGER****.debug("Setting code: {}", code);*

*this.code = code;*

*}*

*public String getName() {*

***LOGGER****.debug("Getting name");*

*return name;*

*}*

*public void setName(String name) {*

***LOGGER****.debug("Setting name: {}", name);*

*this.name = name;*

*}*

*@Override*

*public String toString() {*

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

*}*

*}*

Country.java

*package com.cognizant.spring\_learn.controller;*

*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;*

*import com.cognizant.spring\_learn.Country;*

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

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

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

*return country;*

*}*

*}*

**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">*

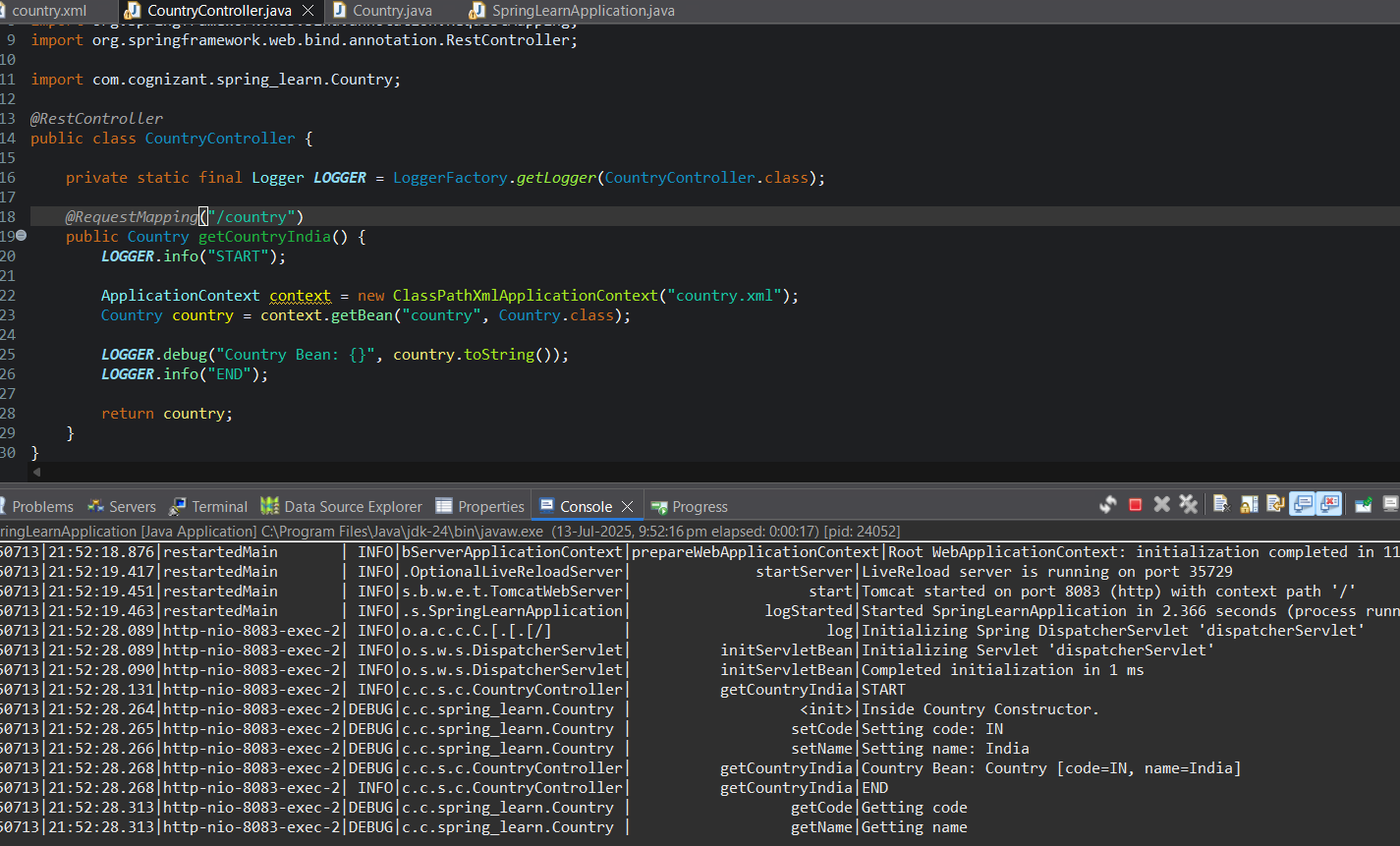
*<****bean*** *id="country" class="com.cognizant.spring\_learn.Country">*

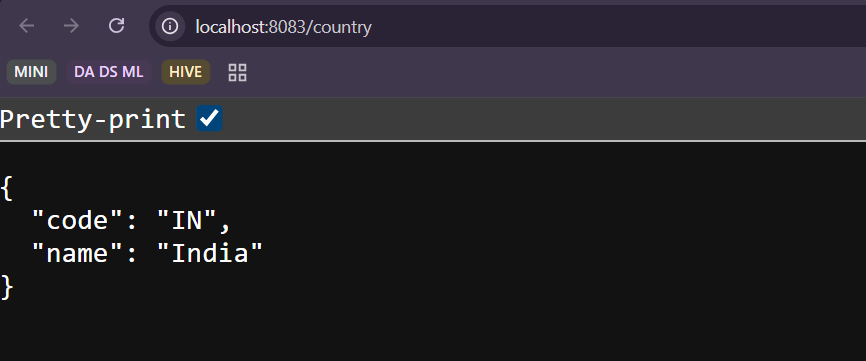
*<****property*** *name="code" value="IN" />*

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

*</****bean****>*

*</****beans****>*





**Mandatory Handson – 05**

**REST - Get country based on country code** 

**Country.xml**

*<****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.spring\_learn.Country">*

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

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

*</****bean****>*

*<****bean*** *id="us" class="com.cognizant.spring\_learn.Country">*

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

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

*</****bean****>*

*<****bean*** *id="de" class="com.cognizant.spring\_learn.Country">*

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

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

*</****bean****>*

*<****bean*** *id="jp" class="com.cognizant.spring\_learn.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.xml**

*package com.cognizant.spring\_learn.service;*

*import java.util.List;*

*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 com.cognizant.spring\_learn.Country;*

*@Service*

*public class CountryService {*

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

*public Country getCountry(String code) {*

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

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

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

*for (Country country : countries) {*

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

*return country;*

*}*

*}*

*return null;*

*}*

*}*

**Country.xml**

*package com.cognizant.spring\_learn.controller;*

*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.GetMapping;*

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

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

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

*import com.cognizant.spring\_learn.Country;*

*import com.cognizant.spring\_learn.service.CountryService;*

*@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("country", Country.class);*

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

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

*return country;*

*}*

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

*public Country getCountry(@PathVariable String code) {*

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

*Country country = countryService.getCountry(code);*

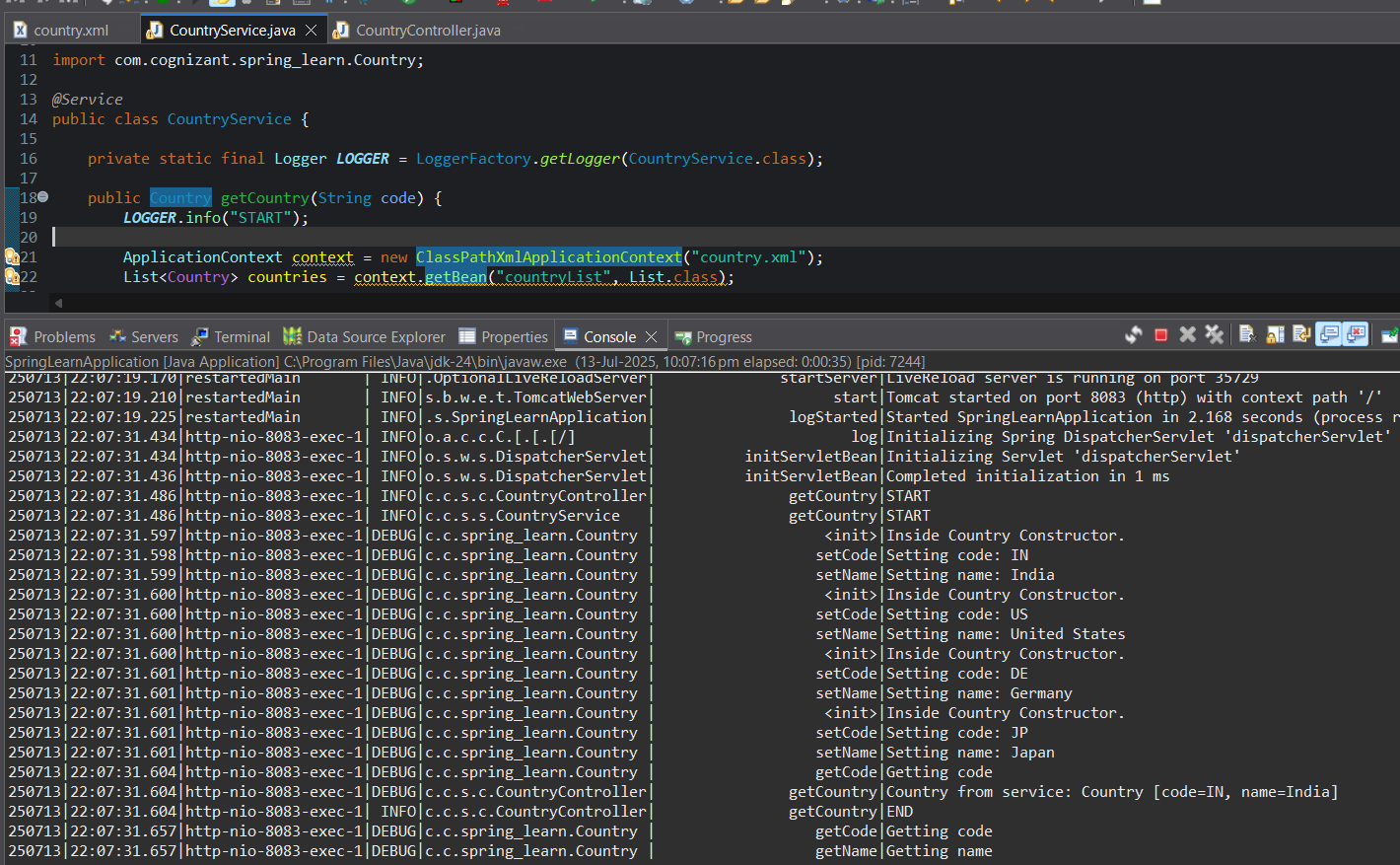
***LOGGER****.debug("Country from service: {}", country);*

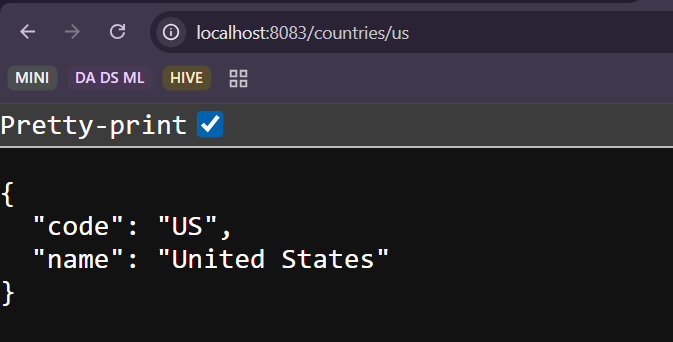
***LOGGER****.info("END");*

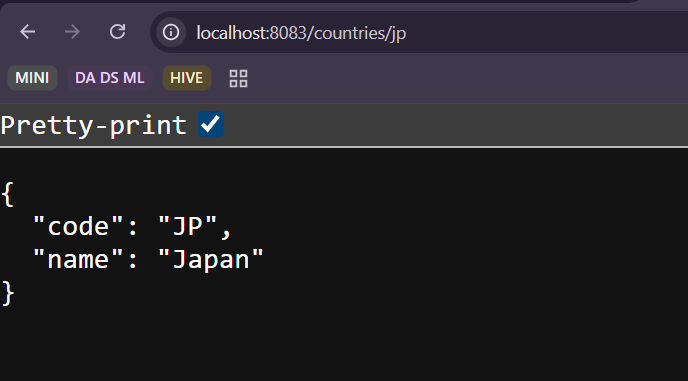
*return country;*

*}*

*}*

**





**Mandatory Handson – 06**

**REST - Get country based on country code** 

**pom.xml**

***Add new dependencies…***

*<?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>*

*<url/>*

*<licenses>*

*<license/>*

*</licenses>*

*<developers>*

*<developer/>*

*</developers>*

*<scm>*

*<connection/>*

*<developerConnection/>*

*<tag/>*

*<url/>*

*</scm>*

*<properties>*

*<java.version>17</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>*

*<dependency>*

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

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

*</dependency>*

*<dependency>*

*<groupId>javax.xml.bind</groupId>*

*<artifactId>jaxb-api</artifactId>*

*<version>2.3.1</version>*

*</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>*

*</dependencies>*

*<build>*

*<plugins>*

*<plugin>*

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

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

*</plugin>*

*</plugins>*

*</build>*

*</project>*

**application.properties**

*spring.application.name=spring-learn*

*server.port=8090*

*logging.level.com.cognizant=debug*

**SecurityConfig.java**

*package com.cognizant.spring\_learn.security;*

*import org.springframework.context.annotation.Bean;*

*import org.springframework.context.annotation.Configuration;*

*import org.springframework.security.authentication.AuthenticationManager;*

*import org.springframework.security.config.Customizer;*

*import org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration;*

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

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

*import org.springframework.security.core.userdetails.UserDetails;*

*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*

*public class SecurityConfig {*

*@Bean*

*public InMemoryUserDetailsManager userDetailsService(PasswordEncoder encoder) {*

*UserDetails user = User.withUsername("user")*

*.password(encoder.encode("pwd"))*

*.roles("USER")*

*.build();*

*return new InMemoryUserDetailsManager(user);*

*}*

*@Bean*

*public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {*

*http.csrf(csrf -> csrf.disable())*

*.httpBasic(Customizer.withDefaults())*

*.authorizeHttpRequests(auth -> auth*

*.requestMatchers("/authenticate").permitAll()*

*.anyRequest().authenticated());*

*return http.build();*

*}*

*@Bean*

*public PasswordEncoder passwordEncoder() {*

*return new BCryptPasswordEncoder();*

*}*

*@Bean*

*public AuthenticationManager authenticationManager(AuthenticationConfiguration config) throws Exception {*

*return config.getAuthenticationManager();*

*}*

*}*

**AuthenticationController.java**

*package com.cognizant.spring\_learn.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.RequestHeader;*

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

*import java.nio.charset.StandardCharsets;*

*import java.security.Key;*

*import java.util.Base64;*

*import java.util.Date;*

*import java.util.HashMap;*

*import java.util.Map;*

*@RestController*

*public class AuthenticationController {*

*@GetMapping("/authenticate")*

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

*String username = extractUsername(authHeader);*

*String token = generateJwtToken(username);*

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

*response.put("token", token);*

*return response;*

*}*

*private String extractUsername(String authHeader) {*

*String encoded = authHeader.substring("Basic ".length());*

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

*String decoded = new String(decodedBytes);*

*return decoded.split(":")[0];*

*}*

*private static final String* ***SECRET*** *= "mySuperSecureSecretKeyThatIs32Char!";*

*private String generateJwtToken(String user) {*

*Key key = Keys.hmacShaKeyFor(****SECRET****.getBytes(StandardCharsets.****UTF\_8****));*

*return Jwts.builder()*

*.setSubject(user)*

*.setIssuedAt(new Date())*

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

*.signWith(key, SignatureAlgorithm.****HS256****)*

*.compact();*

*}*

*}*

