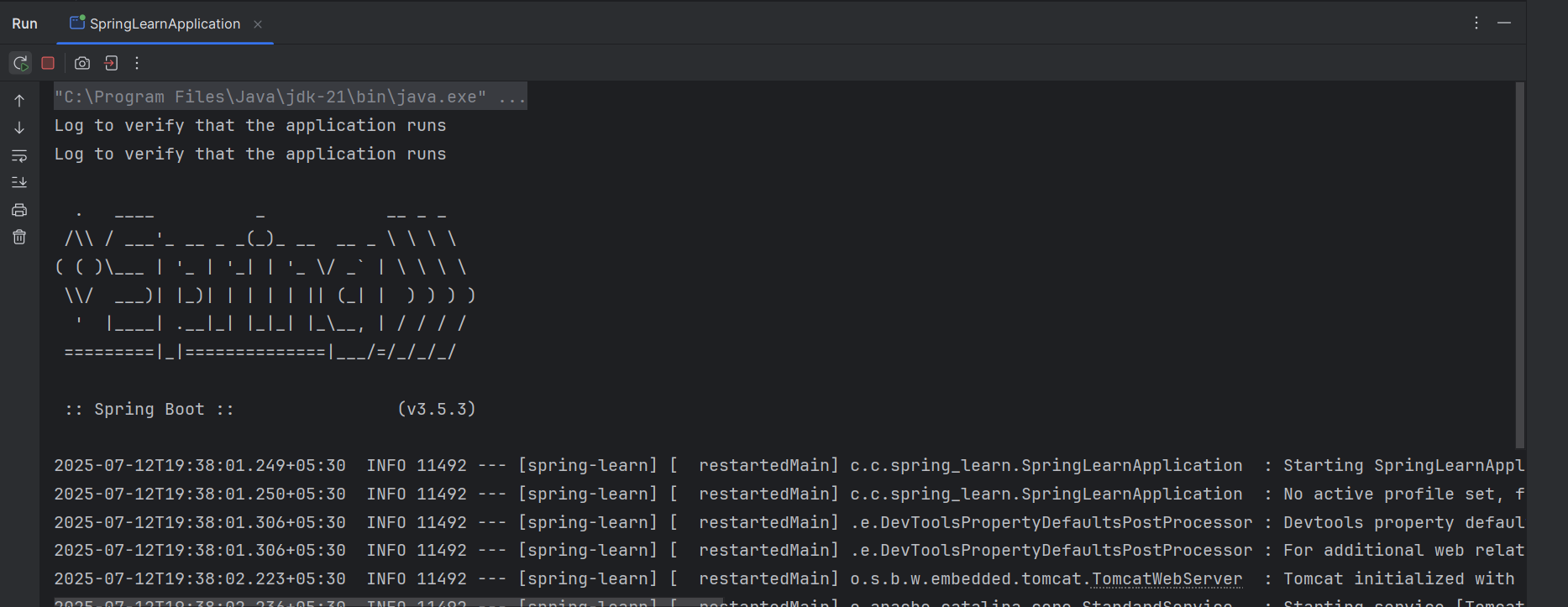
Q1) Create a Spring Web Project using Maven

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) {  
  
 System.*out*.println("Log to verify that the application runs");  
// System.out.println("Why'd it print it twice?");  
 SpringApplication.*run*(SpringLearnApplication.class, args);  
 }  
  
}

Output



Q2) Spring Core – Load Country from Spring Configuration XML

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);  
 *LOGGER*.debug("START main()");  
 *displayCountry*();  
 *LOGGER*.debug("END main()");  
 }  
  
 public static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 *LOGGER*.debug("Country : {}", country);  
 }  
}

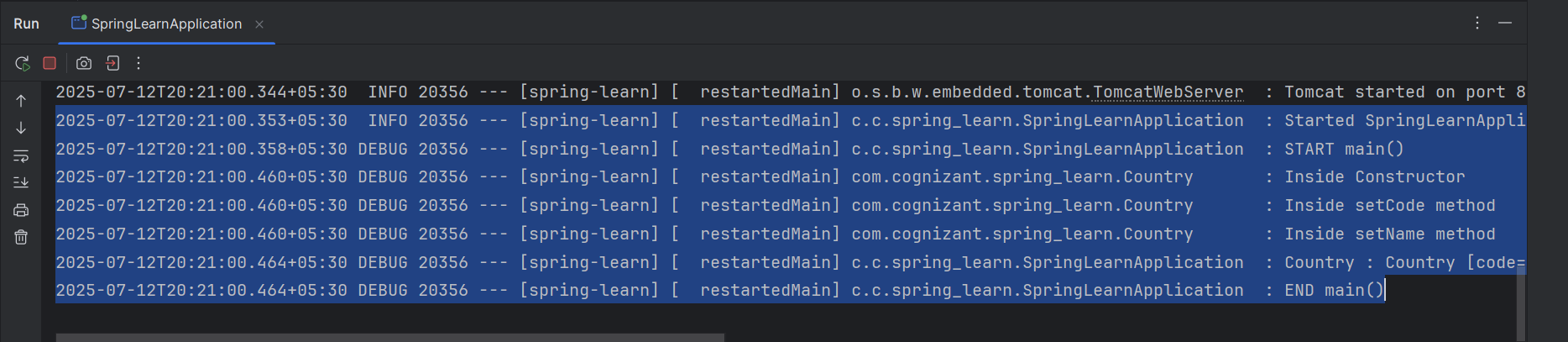
Country.java

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 Constructor");  
 }  
  
 public String getCode() {  
 *LOGGER*.debug("Inside getCode method");  
 return code;  
 }  
  
 public void setCode(String code) {  
 *LOGGER*.debug("Inside setCode method");  
 this.code = code;  
 }  
  
 public String getName() {  
 *LOGGER*.debug("Inside getName method");  
 return name;  
 }  
  
 public void setName(String name) {  
 *LOGGER*.debug("Inside setName method");  
 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  
 http://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

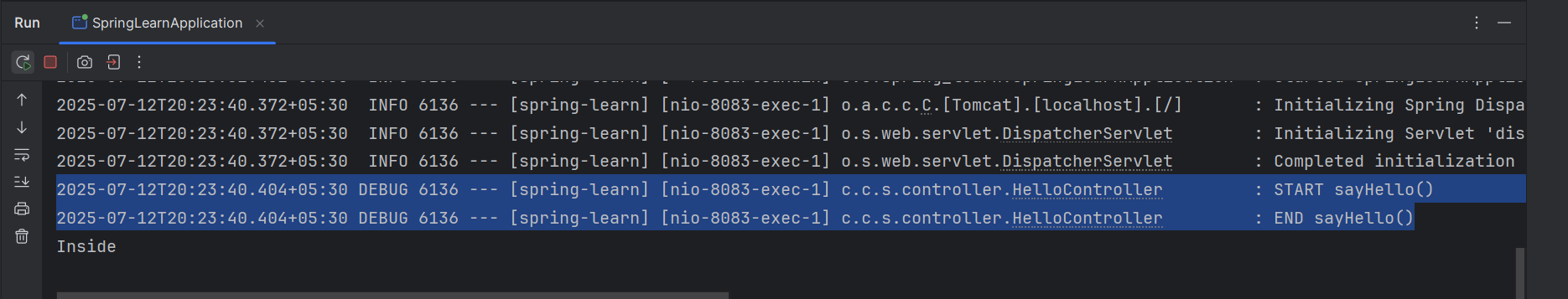


Q3) Hello World RESTful Web Service

SpringApplication.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);  
 }  
}

OUTPUT

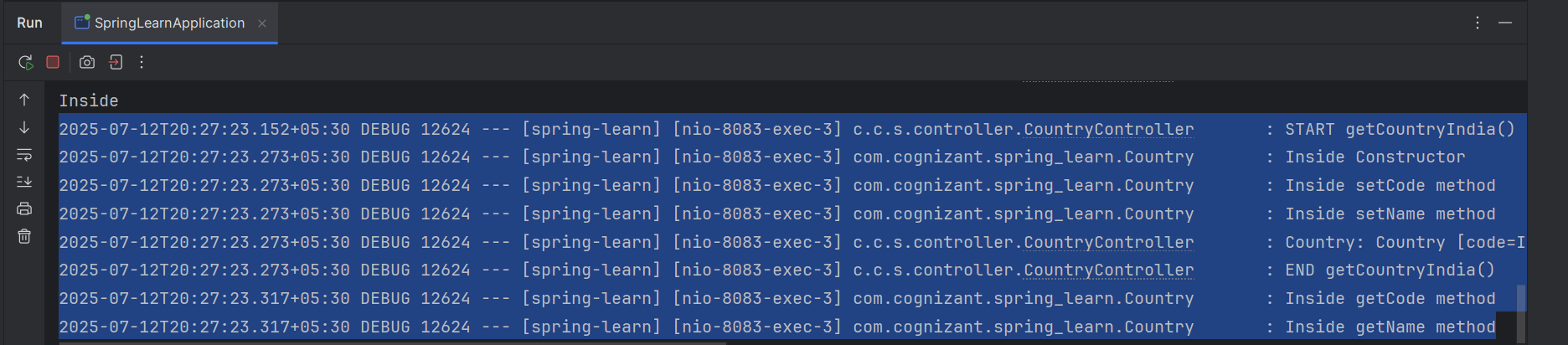


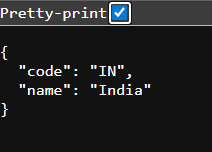
Q4) REST - Country Web Service

CountryController.java

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.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*.debug("START getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = context.getBean("country", Country.class);  
 *LOGGER*.debug("Country: {}", country);  
 *LOGGER*.debug("END getCountryIndia()");  
 return country;  
 }  
}

OUTPUT





Q5) REST - Get country based on country code

CountryService.java

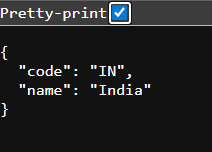
package com.cognizant.spring\_learn.service;  
  
import com.cognizant.spring\_learn.Country;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = context.getBean("countryList", List.class);  
  
 return countries.stream()  
 .filter(country -> country.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);   
 }  
}

countries.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  
 http://www.springframework.org/schema/beans/spring-beans.xsd">  
  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <bean class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="IN" />  
 <property name="name" value="India" />  
 </bean>  
 <bean class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="US" />  
 <property name="name" value="United States" />  
 </bean>  
 <bean class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="DE" />  
 <property name="name" value="Germany" />  
 </bean>  
 <bean class="com.cognizant.spring\_learn.Country">  
 <property name="code" value="JP" />  
 <property name="name" value="Japan" />  
 </bean>  
 </list>  
 </constructor-arg>  
 </bean>  
  
</beans>

CountryController.java

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.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.web.bind.annotation.\*;  
  
@RestController  
public class CountryController {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryController.class);  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 *LOGGER*.debug("START getCountry(): code={}", code);  
 Country result = countryService.getCountry(code);  
 *LOGGER*.debug("END getCountry()");  
 return result;  
 }  
}



Q6) Create authentication service that returns JWT

AuthenticationController.java

package com.cognizant.spring\_learn.controller;  
  
import com.cognizant.spring\_learn.util.JwtUtil;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.Base64;  
  
import javax.servlet.http.HttpServletRequest;  
  
@RestController  
public class AuthenticationController {  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(HttpServletRequest request) {  
 String authHeader = request.getHeader("Authorization");  
  
 if (authHeader == null || !authHeader.startsWith("Basic ")) {  
 return ResponseEntity.*status*(401).body("Missing or invalid Authorization header.");  
 }   
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] decoded = Base64.*getDecoder*().decode(base64Credentials);  
 String credentials = new String(decoded);  
 String[] values = credentials.split(":", 2);  
  
 String username = values[0];  
 String password = values[1];  
  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String token = JwtUtil.generateToken(username);  
 return ResponseEntity.*ok*().body("{\"token\":\"" + token + "\"}");  
 } else {  
 return ResponseEntity.*status*(401).body("Invalid username or password");  
 }  
 }  
}

SecurityConfig.java

package com.cognizant.spring\_learn.config;  
  
import org.springframework.context.annotation.\*;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf().disable()  
 .authorizeRequests()  
 .antMatchers("/authenticate").permitAll()  
 .anyRequest().authenticated()  
 .and()  
 .httpBasic();  
 return http.build();  
 }  
}

JwtUtil.java

package com.cognizant.spring\_learn.util;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
  
import java.util.Date;  
  
public class JwtUtil {  
  
 private static final String *SECRET\_KEY* = "secretkey123";  
 private static final long *EXPIRATION\_TIME* = 10 \* 60 \* 1000;   
  
 public static String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.*currentTimeMillis*()))  
 .setExpiration(new Date(System.*currentTimeMillis*() + *EXPIRATION\_TIME*))  
 .signWith(SignatureAlgorithm.HS256, *SECRET\_KEY*)  
 .compact();  
 }  
}

Output

{

"token": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNzEwNTE0MjI2LCJleHAiOjE3MTA1MTQ4MjZ9.xOuCnd3bcKjdE7NdPYFdz2pZTOlDGH7EIo2P0zHb9xA"

}