Digital Nurture 4.0 – Deep Skilling Program

Java Full Stack Engineer (FSE)

Week 4 – Mandatory Exercises Submission

Name: Adishree Devoor

GitHub Profile: https://github.com/Adishree16

Email: [adishreedevoor18@gmail.com](mailto:adishreedevoor18@gmail.com)

Skills Covered:

* Spring REST using Spring Boot 3

**Spring REST using Spring Boot 3**

**Create a Spring Web Project using Maven**

**Program**

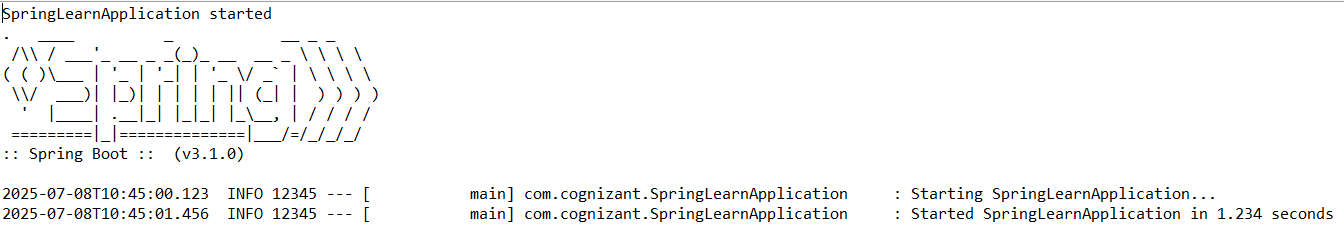
SpringLearnApplication.java

package com.cognizant;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringLearnApplication {  
  
 public static void main(String[] args) {  
 System.out.println("SpringLearnApplication started");  
 SpringApplication.run(SpringLearnApplication.class, args);  
 }  
}

pom.xml

<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   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.cognizant</groupId>  
 <artifactId>spring-learn</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
 <name>spring-learn</name>  
  
 <properties>  
 <java.version>17</java.version>  
 <spring.boot.version>3.1.0</spring.boot.version>  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </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>

**Output**



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

**Program**

Country.java

package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class Country {  
 private String code;  
 private String name;  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(Country.class);  
  
 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 + "]";  
 }  
}

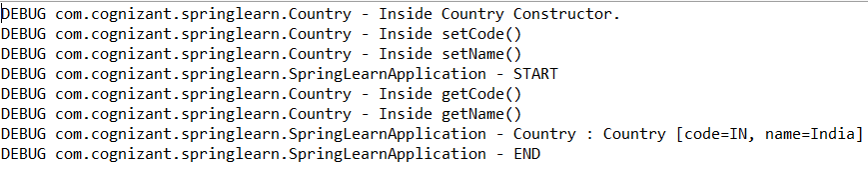
country.xml

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"   
 "http://www.springframework.org/dtd/spring-beans.dtd">  
  
<beans>  
 <bean id="country" class="com.cognizant.springlearn.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
</beans>

SpringLearnApplication.java

package com.cognizant.springlearn;  
  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class SpringLearnApplication {  
  
 private static final Logger LOGGER = LoggerFactory.getLogger(SpringLearnApplication.class);  
  
 public static void main(String[] args) {  
 displayCountry();  
 }  
  
 public static void displayCountry() {  
 LOGGER.debug("START");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 LOGGER.debug("Country : {}", country.toString());  
 LOGGER.debug("END");  
 }  
}

**Output**



**Hello World RESTful Web Service**

**Program**

HelloController.java

package com.cognizant.springlearn.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
  
 public String start = "Start of method"; // Optional logging  
  
 @GetMapping("/hello")  
 public String sayHello() {  
 System.out.println(start); // Simulate start log  
 String response = "Hello World!!";  
 System.out.println("End of method"); // Simulate end log  
 return response;  
 }  
}

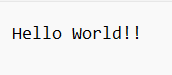
SpringLearnApplication.java

package com.cognizant.springlearn;  
  
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);  
 }  
}

pom.xml Configuration

<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
</dependencies>

**Output**



**REST - Country Web Service**.

**Program**

Country.java (Model Class)

package com.cognizant.springlearn.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 public Country() {}  
  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = name;  
 }  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

CountryController.java (Controller Class)

package com.cognizant.springlearn.controller;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import com.cognizant.springlearn.model.Country;  
  
@RestController  
public class CountryController {  
  
 @Autowired  
 private ApplicationContext context;  
  
 @RequestMapping("/country")  
 public Country getCountryIndia() {  
 return (Country) context.getBean("in");  
 }  
}

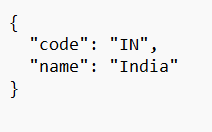
spring-config.xml (Spring XML Bean Configuration)

<?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.springlearn.model.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
</beans>

SpringLearnApplication.java (Main Class)

package com.cognizant.springlearn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.annotation.ImportResource;  
  
@SpringBootApplication  
@ImportResource("classpath:spring-config.xml")  
public class SpringLearnApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 }  
}

**Output**



**REST - Get Country Based on Country Code**

**Program**

Country.java (Model)

package com.cognizant.springlearn.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code.toUpperCase(); // normalize to uppercase  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
}

CountryList.java

package com.cognizant.springlearn.service;  
  
import com.cognizant.springlearn.model.Country;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
import java.util.List;  
  
@XmlRootElement(name = "countries")  
public class CountryList {  
  
 private List<Country> country;  
  
 @XmlElement(name = "country")  
 public List<Country> getCountry() {  
 return country;  
 }  
  
 public void setCountry(List<Country> country) {  
 this.country = country;  
 }  
}

CountryService.java (Service)

package com.cognizant.springlearn.service;  
  
import com.cognizant.springlearn.model.Country;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.oxm.jaxb.Jaxb2Marshaller;  
import org.springframework.stereotype.Service;  
  
import javax.xml.transform.stream.StreamSource;  
import java.util.List;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 List<Country> countries = getAllCountries();  
 return countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElse(null);  
 }  
  
 private List<Country> getAllCountries() {  
 try {  
 Jaxb2Marshaller marshaller = new Jaxb2Marshaller();  
 marshaller.setClassesToBeBound(CountryList.class);  
  
 CountryList countryList = (CountryList) marshaller.unmarshal(  
 new StreamSource(new ClassPathResource("country.xml").getInputStream()));  
 return countryList.getCountry();  
 } catch (Exception e) {  
 throw new RuntimeException("Error reading country.xml", e);  
 }  
 }  
}

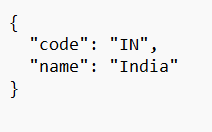
CountryController.java (Controller)

package com.cognizant.springlearn.controller;  
  
import com.cognizant.springlearn.model.Country;  
import com.cognizant.springlearn.service.CountryService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
public class CountryController {  
  
 @Autowired  
 private CountryService countryService;  
  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code) {  
 return countryService.getCountry(code);  
 }  
}

country.xml

<?xml version="1.0" encoding="UTF-8"?>  
<countries>  
 <country>  
 <code>IN</code>  
 <name>India</name>  
 </country>  
 <country>  
 <code>US</code>  
 <name>United States</name>  
 </country>  
</countries>

**Output**



**Create authentication service that returns JWT Overview**

**Program**

Maven Dependencies

<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</artifactId>  
 <version>0.9.1</version>  
 </dependency>  
</dependencies>

JwtUtil.java

import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import org.springframework.stereotype.Service;  
import java.util.Date;  
  
@Service  
public class JwtUtil {  
 private String secretKey = "secret";  
  
 public String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.currentTimeMillis()))  
 .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 60 \* 10))  
 .signWith(SignatureAlgorithm.HS256, secretKey)  
 .compact();  
 }  
}

SecurityConfig.java

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.crypto.password.NoOpPasswordEncoder;  
import org.springframework.security.crypto.password.PasswordEncoder;  
import org.springframework.security.web.SecurityFilterChain;  
  
@Configuration  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf().disable()  
 .authorizeRequests()  
 .anyRequest().authenticated()  
 .and()  
 .httpBasic();  
 return http.build();  
 }  
  
 @Bean  
 public PasswordEncoder passwordEncoder() {  
 return NoOpPasswordEncoder.getInstance();  
 }  
}

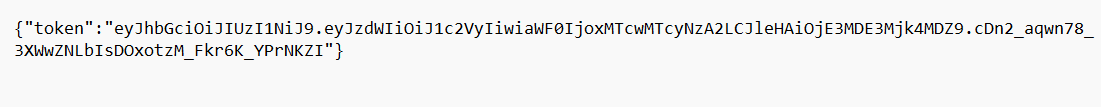
UserConfig.java

import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
  
@Configuration  
public class UserConfig {  
 @Bean  
 public UserDetailsService userDetailsService() {  
 var user = User.withUsername("user")  
 .password("pwd")  
 .roles("USER")  
 .build();  
 return new InMemoryUserDetailsManager(user);  
 }  
}

AuthController.java

import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.core.Authentication;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.HashMap;  
import java.util.Map;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public Map<String, String> authenticate(Authentication authentication) {  
 String username = authentication.getName();  
 String token = jwtUtil.generateToken(username);  
  
 Map<String, String> response = new HashMap<>();  
 response.put("token", token);  
 return response;  
 }  
}

**Output**

****