Week 4

# Exercise 1

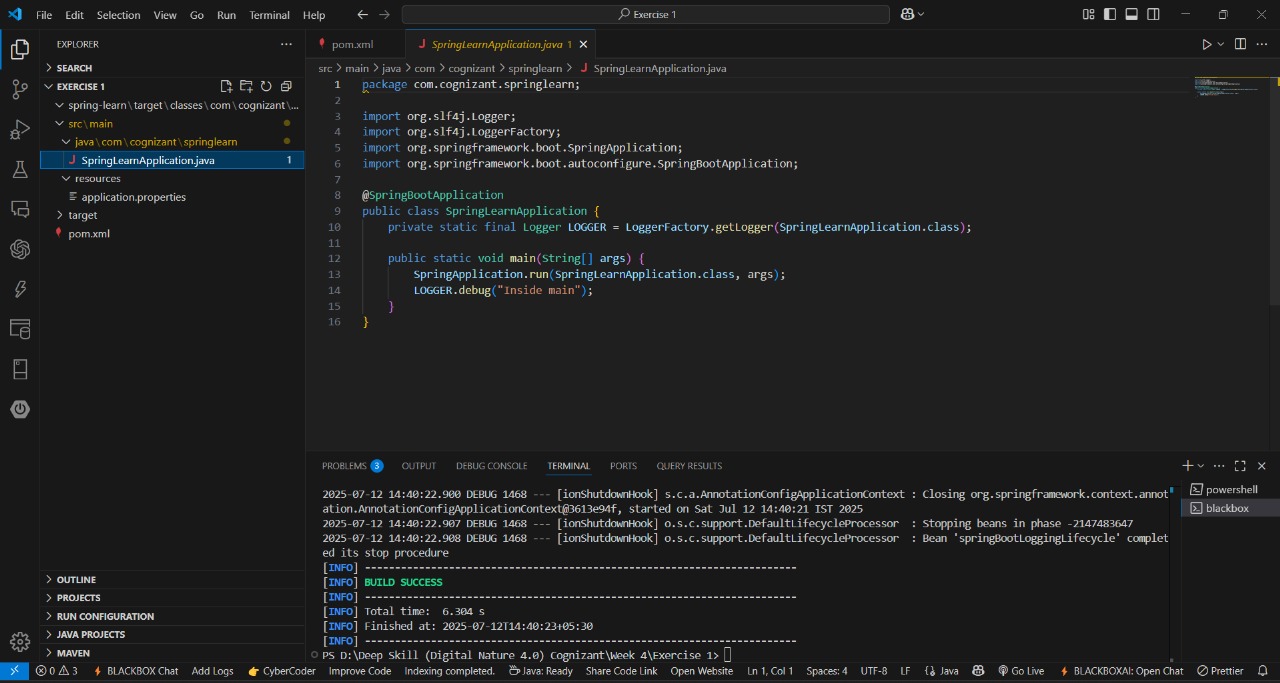
# pom.xml

<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>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.4</version>  
 <relativePath/> <!-- Required if not using local parent -->  
 </parent>  
  
 <properties>  
 <java.version>1.8</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Core Spring Boot -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 </dependency>  
  
 <!-- Dev Tools for hot reloading -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!-- Logging (optional) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-logging</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- Spring Boot Plugin: Required for spring-boot:run -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

# 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) {  
 SpringApplication.run(SpringLearnApplication.class, args);  
 LOGGER.debug("Inside main");  
 }  
}

## Output



# Exercise 2

# 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-core</artifactId>  
 <version>1.0</version>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.22</version>  
 </dependency>  
 <dependency>  
 <groupId>ch.qos.logback</groupId>  
 <artifactId>logback-classic</artifactId>  
 <version>1.2.3</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.codehaus.mojo</groupId>  
 <artifactId>exec-maven-plugin</artifactId>  
 <version>3.0.0</version>  
 <configuration>  
 <mainClass>com.cognizant.springlearn.SpringLearnApplication</mainClass>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

# 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("Getting code");  
 return code;  
 }  
  
 public void setCode(String code) {  
 LOGGER.debug("Setting code");  
 this.code = code;  
 }  
  
 public String getName() {  
 LOGGER.debug("Getting name");  
 return name;  
 }  
  
 public void setName(String name) {  
 LOGGER.debug("Setting name");  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country{" +  
 "code='" + code + '\'' +  
 ", name='" + name + '\'' +  
 '}';  
 }  
}

# SpringLearnApplication.java

package com.cognizant.springlearn;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
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();  
 displayDate();  
 }  
  
 public static void displayCountry() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 LOGGER.debug("Country : {}", country.toString());  
 }  
  
 public static void displayDate() {  
 ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");  
 SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);  
 try {  
 Date date = format.parse("31/12/2018");  
 System.out.println("Parsed Date: " + date);  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 }  
}

## Output

# Exercise 3 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   
 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>  
 <description>Spring Learn Application</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.13</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Starter Web -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Spring Boot Starter Logging (included by default) -->  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- Spring Boot Maven Plugin -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

# 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);  
 }  
}

# 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 of sayHello method");  
 String response = "Hello World!!";  
 logger.info("End of sayHello method");  
 return response;  
 }  
}

## Output

# Exercise 4 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 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>  
 <description>Spring Boot REST service for country</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.5</version>  
 <relativePath /> <!-- lookup parent from repository -->  
 </parent>  
  
 <properties>  
 <java.version>11</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-starter</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-databind</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <resources>  
 <resource>  
 <directory>src/main/resources</directory>  
 <filtering>false</filtering>  
 </resource>  
 </resources>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

# SpringLearnApplication.java

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

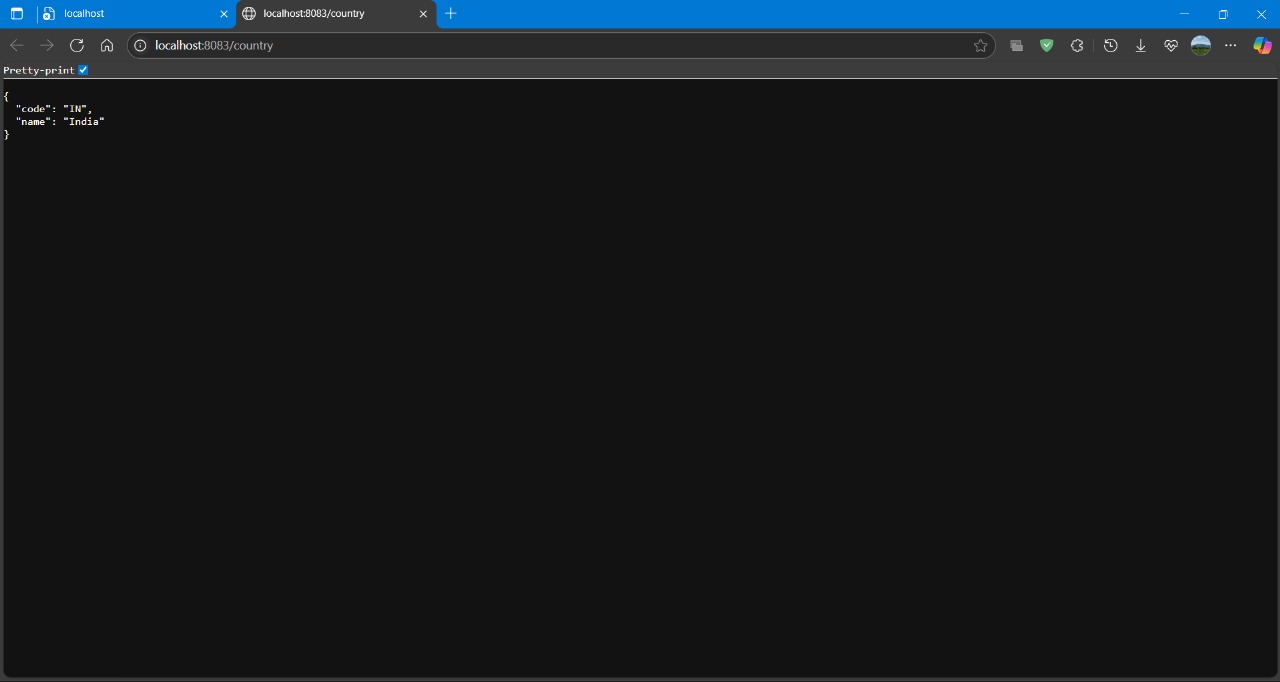
# CountryController.java

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("india");  
 }  
}

# Country.java

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;  
 }  
}

## Output



# Exercise 5 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   
 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>  
 <description>Spring Boot Country REST Service</description>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.7.5</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <properties>  
 <java.version>11</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web Starter -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- JAXB for XML parsing -->  
 <dependency>  
 <groupId>javax.xml.bind</groupId>  
 <artifactId>jaxb-api</artifactId>  
 <version>2.3.1</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.glassfish.jaxb</groupId>  
 <artifactId>jaxb-runtime</artifactId>  
 <version>2.3.1</version>  
 </dependency>  
  
 <!-- Lombok for getter/setter (optional) -->  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </dependency>  
  
 <!-- Spring Boot Test -->  
 <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>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

# 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);  
 }  
}

# CountryController.java

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

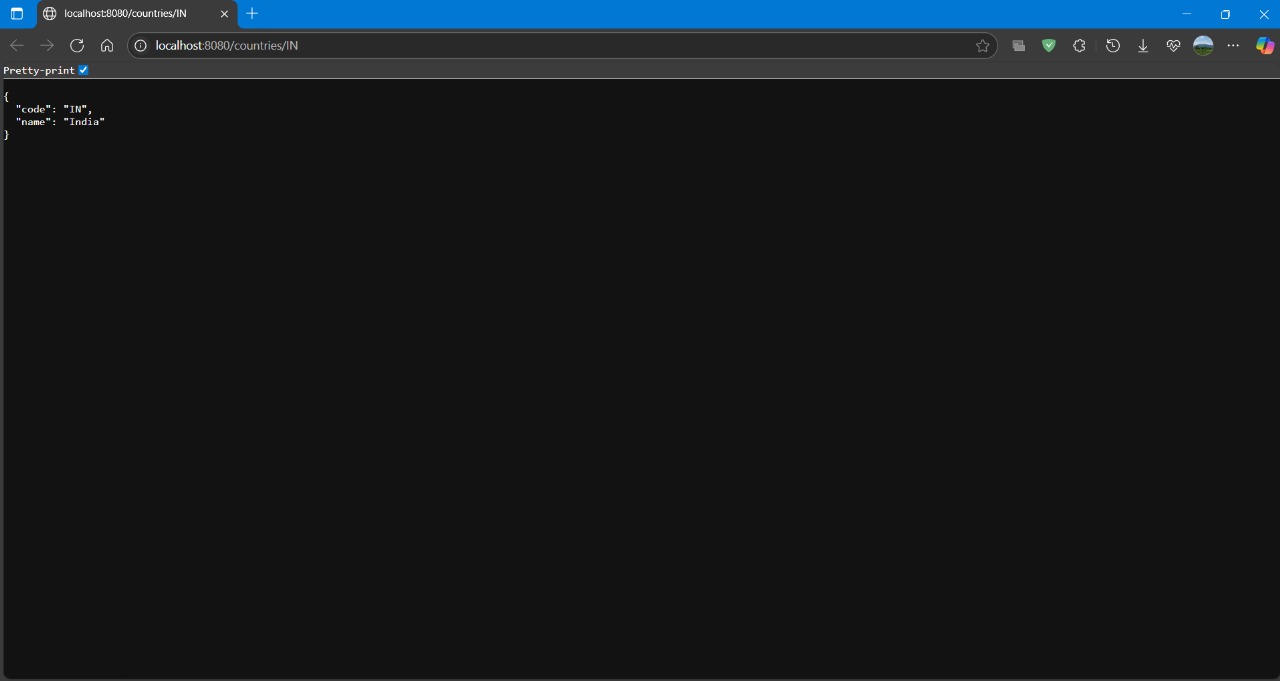
# Country.java

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;  
 }  
}

# CountryService.java

package com.cognizant.springlearn.service;  
  
import com.cognizant.springlearn.model.Country;  
  
import org.springframework.stereotype.Service;  
import org.w3c.dom.\*;  
import javax.xml.parsers.\*;  
import java.io.InputStream;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Optional;  
  
@Service  
public class CountryService {  
  
 public Country getCountry(String code) {  
 List<Country> countries = getCountriesFromXml();  
 if (countries == null) {  
 return null;  
 }  
 Optional<Country> country = countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst();  
 return country.orElse(null);  
 }  
  
 private List<Country> getCountriesFromXml() {  
 List<Country> countries = new ArrayList<>();  
 try {  
 InputStream is = getClass().getClassLoader().getResourceAsStream("country.xml");  
 if (is == null) {  
 return null;  
 }  
 DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
 DocumentBuilder builder = factory.newDocumentBuilder();  
 Document doc = builder.parse(is);  
 doc.getDocumentElement().normalize();  
  
 NodeList nodeList = doc.getElementsByTagName("country");  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node node = nodeList.item(i);  
 if (node.getNodeType() == Node.ELEMENT\_NODE) {  
 Element element = (Element) node;  
 String code = element.getElementsByTagName("code").item(0).getTextContent();  
 String name = element.getElementsByTagName("name").item(0).getTextContent();  
 countries.add(new Country(code, name));  
 }  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 return null;  
 }  
 return countries;  
 }  
}

## Output



# Exercise 6 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>jwt</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>jwt</name>  
 <description>JWT Authentication Example with Spring Boot</description>  
  
 <properties>  
 <java.version>21</java.version>  
 </properties>  
  
 <dependencies>  
  
 <!-- Spring Boot Web -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Spring Security -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-security</artifactId>  
 </dependency>  
  
 <!-- JWT 0.11.5 (works with Java 11+) -->  
 <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> <!-- or jjwt-gson if you prefer -->  
 <version>0.11.5</version>  
 <scope>runtime</scope>  
 </dependency>  
  
 <!-- Dev Tools (Optional) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <!-- Test (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>  
 </plugin>  
 </plugins>  
 </build>  
</project>

# JwtApplication.java

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

# SecurityConfig.java

package com.cognizant.jwt.config;  
  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.context.annotation.Bean;  
  
@Configuration  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf().disable()  
 .authorizeHttpRequests()  
 .requestMatchers("/authenticate").permitAll()  
 .anyRequest().authenticated()  
 .and().httpBasic(); // Basic Auth  
 return http.build();  
 }  
}

# AuthController.java

package com.cognizant.jwt.controller;  
  
import com.cognizant.jwt.util.JwtUtil;  
import jakarta.servlet.http.HttpServletRequest;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.Base64;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public String generateToken(HttpServletRequest request) {  
 String authHeader = request.getHeader("Authorization");  
  
 if (authHeader != null && authHeader.startsWith("Basic ")) {  
 String base64Credentials = authHeader.substring("Basic ".length());  
 byte[] credDecoded = Base64.getDecoder().decode(base64Credentials);  
 String credentials = new String(credDecoded);  
 String[] values = credentials.split(":", 2);  
 String username = values[0];  
 String password = values[1];  
  
 // You can replace this with your actual user validation  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String token = jwtUtil.generateToken(username);  
 return "{\"token\":\"" + token + "\"}";  
 }  
 }  
  
 return "{\"error\":\"Invalid credentials\"}";  
 }  
}

# JwtUtil.java

package com.cognizant.jwt.util;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import org.springframework.stereotype.Component;  
  
import java.util.Date;  
  
@Component  
public class JwtUtil {  
  
 private final String SECRET\_KEY = "secret";  
  
 public String generateToken(String username) {  
 return Jwts.builder()  
 .setSubject(username)  
 .setIssuedAt(new Date(System.currentTimeMillis()))  
 .setExpiration(new Date(System.currentTimeMillis() + 1000 \* 60 \* 60)) // 1 hour  
 .signWith(SignatureAlgorithm.HS256, SECRET\_KEY)  
 .compact();  
 }  
}

# JwtApplicationTests.java

package com.cognizant.jwt;  
  
import org.junit.jupiter.api.Test;  
import org.springframework.boot.test.context.SpringBootTest;  
  
@SpringBootTest  
class JwtApplicationTests {  
  
 @Test  
 void contextLoads() {  
 }  
  
}

## Output

