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

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>springcore</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Context (includes core, beans, context, and expression) -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
  
 <!-- Spring Beans (for XML parsing and bean handling) -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-beans</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
  
 <!-- Spring Core (required base module) -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-core</artifactId>  
 <version>5.3.30</version>  
 </dependency>  
  
 <!-- Logging dependencies -->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>2.0.9</version>  
 </dependency>  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-simple</artifactId>  
 <version>2.0.9</version>  
 </dependency>  
 </dependencies>  
  
</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 Country Code: {}", code);  
 return code;  
 }  
  
 public void setCode(String code) {  
 *LOGGER*.debug("Setting Country Code: {}", code);  
 this.code = code;  
 }  
  
 public String getName() {  
 *LOGGER*.debug("Getting Country Name: {}", name);  
 return name;  
 }  
  
 public void setName(String name) {  
 *LOGGER*.debug("Setting Country Name: {}", name);  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

SpringLearnApplication.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 Country Code: {}", code);  
 return code;  
 }  
  
 public void setCode(String code) {  
 *LOGGER*.debug("Setting Country Code: {}", code);  
 this.code = code;  
 }  
  
 public String getName() {  
 *LOGGER*.debug("Getting Country Name: {}", name);  
 return name;  
 }  
  
 public void setName(String name) {  
 *LOGGER*.debug("Setting Country Name: {}", name);  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country [code=" + code + ", name=" + name + "]";  
 }  
}

simplelogger.properties

org.slf4j.simpleLogger.defaultLogLevel=debug  
org.slf4j.simpleLogger.logFile=System.out  
org.slf4j.simpleLogger.showDateTime=true  
org.slf4j.simpleLogger.dateTimeFormat=HH:mm:ss  
org.slf4j.simpleLogger.showThreadName=true  
org.slf4j.simpleLogger.showLogName=true  
org.slf4j.simpleLogger.showShortLogName=false

**OUTPUT:**  
"C:\Program Files\Amazon Corretto\jdk17.0.7\_7\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.3\lib\idea\_rt.jar=59457" -Dfile.encoding=UTF-8 -classpath C:\Users\HP\IdeaProjects\Springcore\target\classes;C:\Users\HP\.m2\repository\org\springframework\spring-context\5.3.30\spring-context-5.3.30.jar;C:\Users\HP\.m2\repository\org\springframework\spring-aop\5.3.30\spring-aop-5.3.30.jar;C:\Users\HP\.m2\repository\org\springframework\spring-expression\5.3.30\spring-expression-5.3.30.jar;C:\Users\HP\.m2\repository\org\springframework\spring-beans\5.3.30\spring-beans-5.3.30.jar;C:\Users\HP\.m2\repository\org\springframework\spring-core\5.3.30\spring-core-5.3.30.jar;C:\Users\HP\.m2\repository\org\springframework\spring-jcl\5.3.30\spring-jcl-5.3.30.jar;C:\Users\HP\.m2\repository\org\slf4j\slf4j-api\2.0.9\slf4j-api-2.0.9.jar;C:\Users\HP\.m2\repository\org\slf4j\slf4j-simple\2.0.9\slf4j-simple-2.0.9.jar com.cognizant.springlearn.SpringLearnApplication

17:36:50 [main] DEBUG org.springframework.context.support.ClassPathXmlApplicationContext - Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@123772c4

17:36:50 [main] DEBUG org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loaded 1 bean definitions from class path resource [country.xml]

17:36:50 [main] DEBUG org.springframework.beans.factory.support.DefaultListableBeanFactory - Creating shared instance of singleton bean 'country'

17:36:50 [main] DEBUG com.cognizant.springlearn.Country - Inside Country Constructor.

17:36:50 [main] DEBUG com.cognizant.springlearn.Country - Setting Country Code: IN

17:36:50 [main] DEBUG com.cognizant.springlearn.Country - Setting Country Name: India

17:36:50 [main] DEBUG com.cognizant.springlearn.SpringLearnApplication - Country: Country [code=IN, name=India]

Process finished with exit code 0

**Hello World RESTful Web Service**

**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>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.4</version>  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <groupId>com.cognizant.springlearn</groupId>  
 <artifactId>helloworld</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>Hello World RESTful Web Service</name>  
 <description>Simple Spring Boot REST Application</description>  
 <packaging>jar</packaging>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Boot Web (REST controller + embedded Tomcat) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- Optional: Developer Tools (auto-restart, hot reload) -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
  
 <!-- Optional: Testing framework -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <!-- Maven Plugin for Spring Boot -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

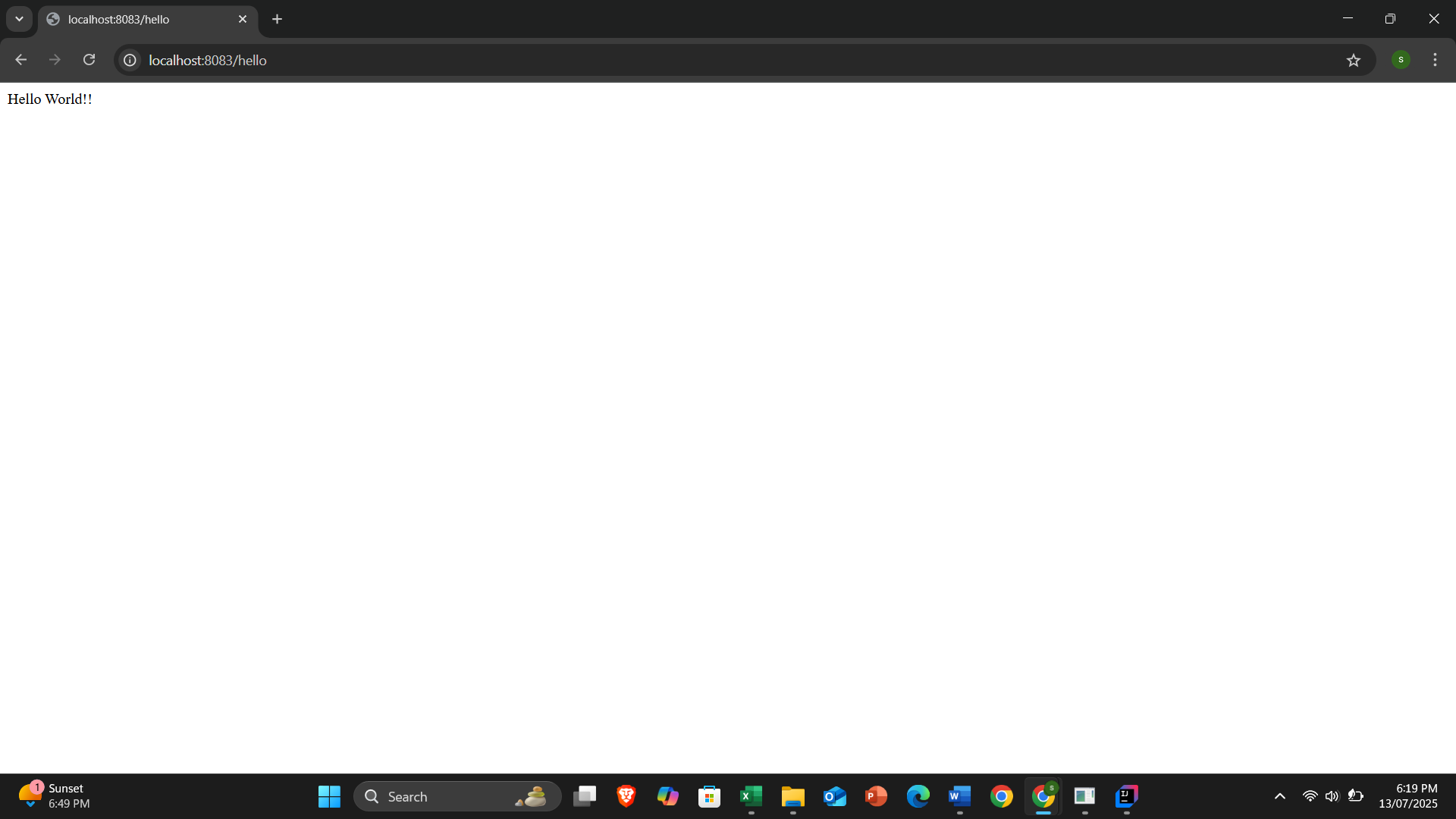
**HelloWorldApplication.java**

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

**HelloController.java**

package com.cognizant.springlearn.helloworld.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 sayHello()");  
 String response = "Hello World!!";  
 *LOGGER*.info("END sayHello()");  
 return response;  
 }  
}

**OUTPUT:**



**4. REST - Country Web Service**

**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>Country\_Web\_Service</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.4</version>  
 <relativePath/>  
 </parent>  
  
 <dependencies>  
 <!-- Spring Web for REST APIs -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <!-- XML Configuration Support -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 </dependency>  
  
 <!-- Logging -->  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-logging</artifactId>  
 </dependency>  
  
 <!-- SLF4J Logger -->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 </dependency>  
  
 </dependencies>  
  
 <properties>  
 <java.version>17</java.version>  
 </properties>  
  
 <build>  
 <plugins>  
 <!-- Maven Plugin for Spring Boot -->  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
import com.cognizant.springlearn.model.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*.info("START getCountryIndia()");  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 Country country = (Country) context.getBean("country", Country.class);  
 *LOGGER*.info("END getCountryIndia()");  
 return country;  
 }  
}

**Country.java**

package com.cognizant.springlearn.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 public Country() {  
 System.*out*.println("Inside Country Constructor.");  
 }  
  
 public String getCode() {  
 System.*out*.println("Getting Country Code");  
 return code;  
 }  
  
 public void setCode(String code) {  
 System.*out*.println("Setting Country Code");  
 this.code = code;  
 }  
  
 public String getName() {  
 System.*out*.println("Getting Country Name");  
 return name;  
 }  
  
 public void setName(String name) {  
 System.*out*.println("Setting Country Name");  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country{" + "code='" + code + '\'' + ", name='" + name + '\'' + '}';  
 }  
}

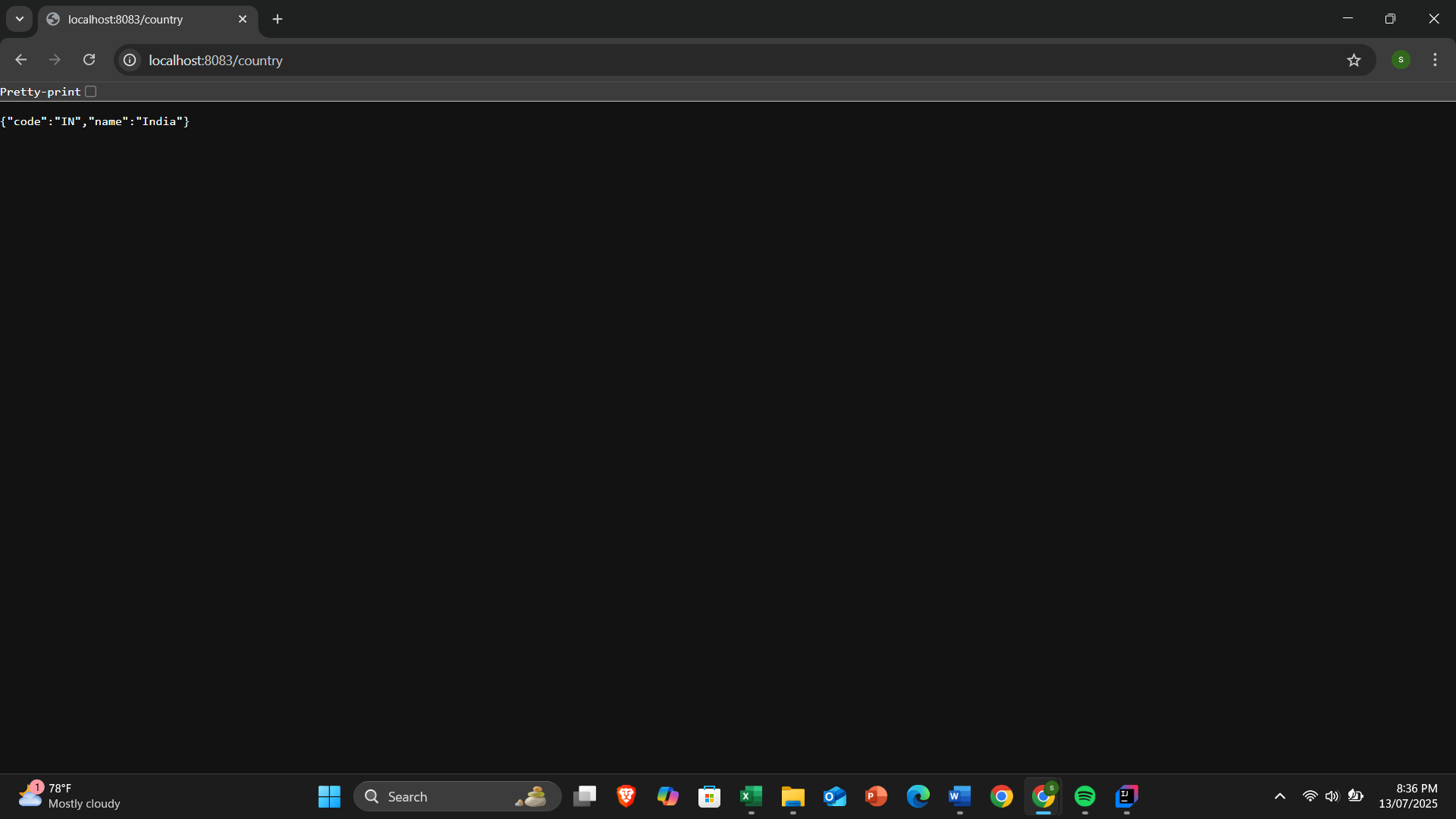
**SpringLearnApplication.java**

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);  
 }  
}

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

**OUTPUT:**



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

**CountryController.java**

package com.cognizant.springlearn.controller;  
  
import com.cognizant.springlearn.model.Country;  
import com.cognizant.springlearn.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) throws Exception {  
 *LOGGER*.info("START getCountry()");  
 Country country = countryService.getCountry(code);  
 *LOGGER*.info("END getCountry()");  
 return country;  
 }  
}

**Country.java**

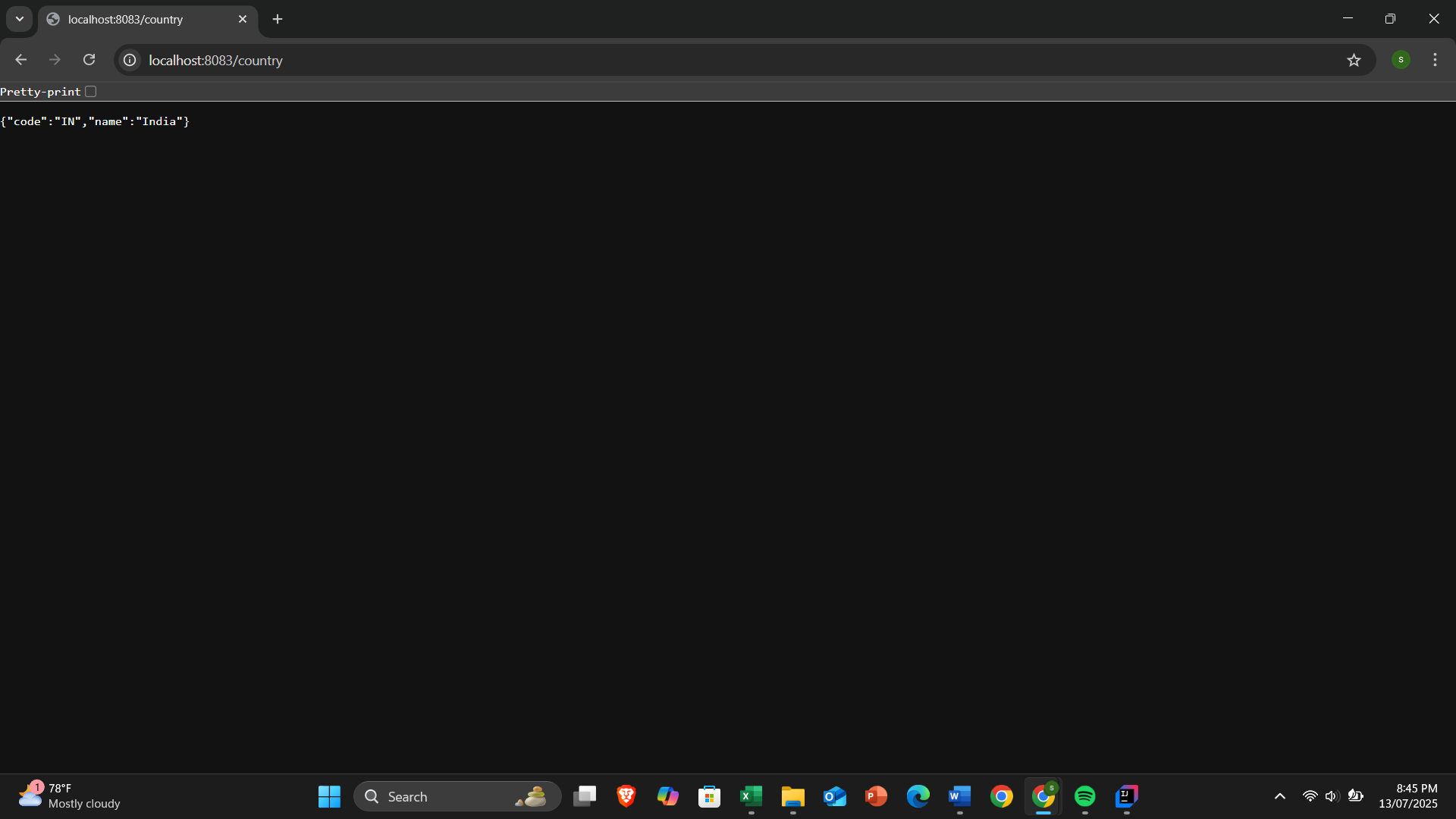
package com.cognizant.springlearn.model;  
  
public class Country {  
 private String code;  
 private String name;  
  
 public Country() {}  
  
 public Country(String code, String name) {  
 this.code = code;  
 this.name = name;  
 }  
  
 public String getCode() {  
 return code;  
 }  
  
 public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

**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);  
 }  
}

**CountryService.java**

package com.cognizant.springlearn.service;  
  
import com.cognizant.springlearn.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) throws Exception {  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countries = (List<Country>) context.getBean("countryList");  
  
 return countries.stream()  
 .filter(c -> c.getCode().equalsIgnoreCase(code))  
 .findFirst()  
 .orElseThrow(() -> new Exception("Country not found"));  
 }  
}

**OUTPUT:**

**6.Create authentication service that returns JWT**

**Config/SecurityConfig.java**

package com.example.jwt.config;  
  
import org.springframework.context.annotation.Bean;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
  
@EnableWebSecurity  
public class SecurityConfig {  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf().disable()  
 .authorizeHttpRequests(auth -> auth.anyRequest().permitAll());  
 return http.build();  
 }  
}

**controller/AuthController.java**

package com.example.jwt.controller;  
  
import com.example.jwt.util.JwtUtil;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestHeader;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.Base64;  
  
@RestController  
public class AuthController {  
  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<?> authenticate(@RequestHeader("Authorization") String authHeader) {  
 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];  
  
 if ("user".equals(username) && "pwd".equals(password)) {  
 String token = jwtUtil.generateToken(username);  
 return ResponseEntity.*ok*().body("{\"token\":\"" + token + "\"}");  
 }  
 }  
 return ResponseEntity.*status*(401).body("Unauthorized");  
 }  
}

**util/JwtAuthApplication.java**

package com.example.jwt;  
  
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);  
 }  
}

**application.properties**server.port=8090

**OUTPUT:**

"C:\Program Files\Amazon Corretto\jdk17.0.7\_7\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.3\lib\idea\_rt.jar=56552" -Dfile.encoding=UTF-8 -classpath C:\Users\HP\IdeaProjects\Authentication\_Service\_JWT\target\classes;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-starter-web\3.2.4\spring-boot-starter-web-3.2.4.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-starter\3.2.4\spring-boot-starter-3.2.4.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot\3.2.4\spring-boot-3.2.4.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-autoconfigure\3.2.4\spring-boot-autoconfigure-3.2.4.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-starter-logging\3.2.4\spring-boot-starter-logging-3.2.4.jar;C:\Users\HP\.m2\repository\ch\qos\logback\logback-classic\1.4.14\logback-classic-1.4.14.jar;C:\Users\HP\.m2\repository\ch\qos\logback\logback-core\1.4.14\logback-core-1.4.14.jar;C:\Users\HP\.m2\repository\org\slf4j\slf4j-api\2.0.12\slf4j-api-2.0.12.jar;C:\Users\HP\.m2\repository\org\apache\logging\log4j\log4j-to-slf4j\2.21.1\log4j-to-slf4j-2.21.1.jar;C:\Users\HP\.m2\repository\org\apache\logging\log4j\log4j-api\2.21.1\log4j-api-2.21.1.jar;C:\Users\HP\.m2\repository\org\slf4j\jul-to-slf4j\2.0.12\jul-to-slf4j-2.0.12.jar;C:\Users\HP\.m2\repository\jakarta\annotation\jakarta.annotation-api\2.1.1\jakarta.annotation-api-2.1.1.jar;C:\Users\HP\.m2\repository\org\springframework\spring-core\6.1.5\spring-core-6.1.5.jar;C:\Users\HP\.m2\repository\org\springframework\spring-jcl\6.1.5\spring-jcl-6.1.5.jar;C:\Users\HP\.m2\repository\org\yaml\snakeyaml\2.2\snakeyaml-2.2.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-starter-json\3.2.4\spring-boot-starter-json-3.2.4.jar;C:\Users\HP\.m2\repository\com\fasterxml\jackson\datatype\jackson-datatype-jdk8\2.15.4\jackson-datatype-jdk8-2.15.4.jar;C:\Users\HP\.m2\repository\com\fasterxml\jackson\datatype\jackson-datatype-jsr310\2.15.4\jackson-datatype-jsr310-2.15.4.jar;C:\Users\HP\.m2\repository\com\fasterxml\jackson\module\jackson-module-parameter-names\2.15.4\jackson-module-parameter-names-2.15.4.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-starter-tomcat\3.2.4\spring-boot-starter-tomcat-3.2.4.jar;C:\Users\HP\.m2\repository\org\apache\tomcat\embed\tomcat-embed-core\10.1.19\tomcat-embed-core-10.1.19.jar;C:\Users\HP\.m2\repository\org\apache\tomcat\embed\tomcat-embed-el\10.1.19\tomcat-embed-el-10.1.19.jar;C:\Users\HP\.m2\repository\org\apache\tomcat\embed\tomcat-embed-websocket\10.1.19\tomcat-embed-websocket-10.1.19.jar;C:\Users\HP\.m2\repository\org\springframework\spring-web\6.1.5\spring-web-6.1.5.jar;C:\Users\HP\.m2\repository\org\springframework\spring-beans\6.1.5\spring-beans-6.1.5.jar;C:\Users\HP\.m2\repository\io\micrometer\micrometer-observation\1.12.4\micrometer-observation-1.12.4.jar;C:\Users\HP\.m2\repository\io\micrometer\micrometer-commons\1.12.4\micrometer-commons-1.12.4.jar;C:\Users\HP\.m2\repository\org\springframework\spring-webmvc\6.1.5\spring-webmvc-6.1.5.jar;C:\Users\HP\.m2\repository\org\springframework\spring-context\6.1.5\spring-context-6.1.5.jar;C:\Users\HP\.m2\repository\org\springframework\spring-expression\6.1.5\spring-expression-6.1.5.jar;C:\Users\HP\.m2\repository\org\springframework\boot\spring-boot-starter-security\3.2.4\spring-boot-starter-security-3.2.4.jar;C:\Users\HP\.m2\repository\org\springframework\spring-aop\6.1.5\spring-aop-6.1.5.jar;C:\Users\HP\.m2\repository\org\springframework\security\spring-security-config\6.2.3\spring-security-config-6.2.3.jar;C:\Users\HP\.m2\repository\org\springframework\security\spring-security-core\6.2.3\spring-security-core-6.2.3.jar;C:\Users\HP\.m2\repository\org\springframework\security\spring-security-crypto\6.2.3\spring-security-crypto-6.2.3.jar;C:\Users\HP\.m2\repository\org\springframework\security\spring-security-web\6.2.3\spring-security-web-6.2.3.jar;C:\Users\HP\.m2\repository\io\jsonwebtoken\jjwt\0.9.1\jjwt-0.9.1.jar;C:\Users\HP\.m2\repository\com\fasterxml\jackson\core\jackson-databind\2.15.4\jackson-databind-2.15.4.jar;C:\Users\HP\.m2\repository\com\fasterxml\jackson\core\jackson-annotations\2.15.4\jackson-annotations-2.15.4.jar;C:\Users\HP\.m2\repository\com\fasterxml\jackson\core\jackson-core\2.15.4\jackson-core-2.15.4.jar com.example.jwt.JwtAuthApplication

:: Spring Boot :: (v3.2.4)

2025-07-13T22:10:43.368+05:30 INFO 13856 --- [ main] com.example.jwt.JwtAuthApplication : Starting JwtAuthApplication using Java 17.0.7 with PID 13856 (C:\Users\HP\IdeaProjects\Authentication\_Service\_JWT\target\classes started by HP in C:\Users\HP\IdeaProjects\Authentication\_Service\_JWT)

2025-07-13T22:10:43.372+05:30 INFO 13856 --- [ main] com.example.jwt.JwtAuthApplication : No active profile set, falling back to 1 default profile: "default"

2025-07-13T22:10:45.125+05:30 INFO 13856 --- [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)

2025-07-13T22:10:45.145+05:30 INFO 13856 --- [ main] o.apache.catalina.core.StandardService : Starting service [Tomcat]

2025-07-13T22:10:45.146+05:30 INFO 13856 --- [ main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.19]

2025-07-13T22:10:45.241+05:30 INFO 13856 --- [ main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext

2025-07-13T22:10:45.242+05:30 INFO 13856 --- [ main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1769 ms

2025-07-13T22:10:45.818+05:30 WARN 13856 --- [ main] .s.s.UserDetailsServiceAutoConfiguration :

Using generated security password: 4749e74e-6965-4fef-a8f6-f40445584df0

This generated password is for development use only. Your security configuration must be updated before running your application in production.

2025-07-13T22:10:45.988+05:30 INFO 13856 --- [ main] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.session.DisableEncodeUrlFilter@24eeac69, org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@18539cb1, org.springframework.security.web.context.SecurityContextHolderFilter@1697f2b3, org.springframework.security.web.header.HeaderWriterFilter@1192b58e, org.springframework.web.filter.CorsFilter@748f93bb,org.springframework.security.web.csrf.CsrfFilter@2676dc05, org.springframework.security.web.authentication.logout.LogoutFilter@7377781e, org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter@6bc24e72, org.springframework.security.web.authentication.ui.DefaultLoginPageGeneratingFilter@463561c5,org.springframework.security.web.authentication.ui.DefaultLogoutPageGeneratingFilter@7f2d31af, org.springframework.security.web.authentication.www.BasicAuthenticationFilter@6d2a2560, org.springframework.security.web.savedrequest.RequestCacheAwareFilter@3134153d, org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@767599a7, org.springframework.security.web.authentication.AnonymousAuthenticationFilter@2e7157c7, org.springframework.security.web.access.ExceptionTranslationFilter@31142d58, org.springframework.security.web.access.intercept.AuthorizationFilter@21eee94f]

2025-07-13T22:10:46.139+05:30 INFO 13856 --- [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path ''

2025-07-13T22:10:46.154+05:30 INFO 13856 --- [ main] com.example.jwt.JwtAuthApplication : Started JwtAuthApplication in 3.3 seconds (process running for 3.795)