**Create a Spring Web Project using Maven:**

**1. Visit Spring Initializr:**

* Go to: <https://start.spring.io>

**2. Configure Spring Project:**

* **Group**: com.cognizant
* **Artifact**: spring-learn
* **Name**: spring-learn (auto-filled)
* **Description**: Demo project for Spring Web
* **Package Name**: com.cognizant.springlearn
* **Packaging**: Jar
* **Java Version**: Choose 17 (or 11 if your system only supports that)

**3. Add Dependencies:**

* Click Add Dependencies and select:
  + Spring Web
  + Spring Boot DevTools

**4. Generate Project:**

* Click GENERATE to download the ZIP.
* Save the ZIP to your Eclipse Workspace folder.
* Extract the ZIP file.

**BUILDING USING MAVEN (with Proxy)**

* Open Command Prompt / Terminal and navigate to the extracted folder:

**Now run the Maven build with proxy settings:**

mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456

**IMPORT PROJECT IN ECLIPSE**

1. Open Eclipse.
2. File > Import
3. Choose: Maven > Existing Maven Projects > Click Next.
4. Browse to the extracted project folder.
5. Ensure the project is selected and click Finish.

**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>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>

<!-- Spring Boot Web Starter -->

<dependency>

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

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Boot DevTools (runtime only) -->

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<!-- Spring Boot Starter Test -->

<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>

**SpringLearnApplication.java:**

package com.cognizant.springlearn;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

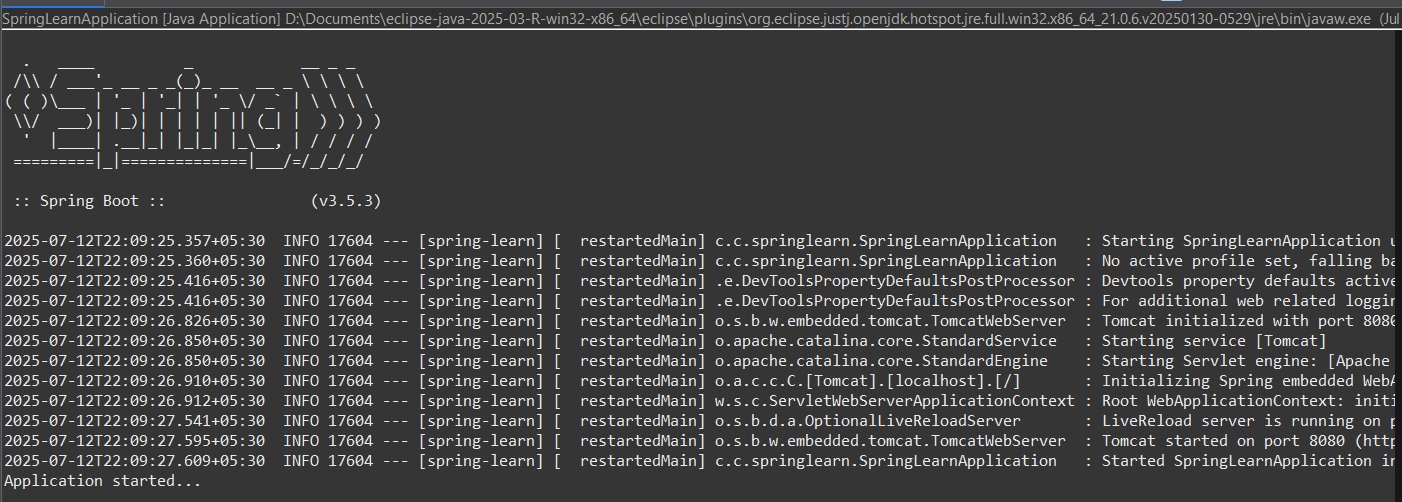
SpringApplication.run(SpringLearnApplication.class, args);

System.out.println("Application started...");

}

}

**OUTPUT:**

****

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

In your **pom.xml**, include:

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>2.0.7</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>2.0.7</version>

</dependency>

**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("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:**

<!-- src/main/resources/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.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) {

LOGGER.debug("START");

displayCountry();

LOGGER.debug("END");

}

public static void displayCountry() {

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

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

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

}

}

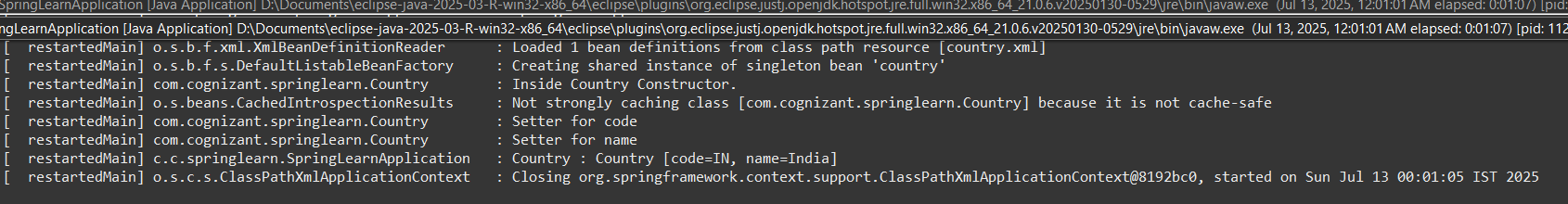
**application.properties:**

spring.application.name=spring-learn

logging.level.com.cognizant.springlearn=DEBUG

logging.level.root=DEBUG

**OUTPUT:**

****

**Hello World RESTful Web Service**

**1. Add Controller Class**

Create a new Java class:

**src/main/java/com/cognizant/springlearn/controller/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.debug("Start sayHello()");

String response = "Hello World!!";

LOGGER.debug("End sayHello()");

return response;

}

}

**application.properties:**

# Set server port to 8083

server.port=8083

# Logging

logging.level.root=DEBUG

logging.level.com.cognizant.springlearn.controller=DEBUG

**SpringLearnApplication.java:**

package com.cognizant.springlearn;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

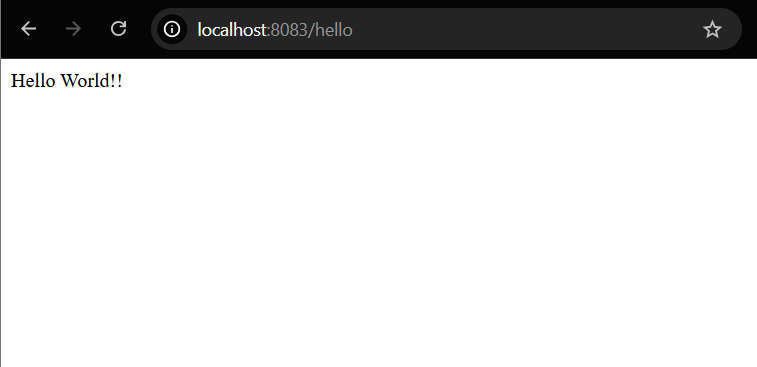
SpringApplication.run(SpringLearnApplication.class, args);

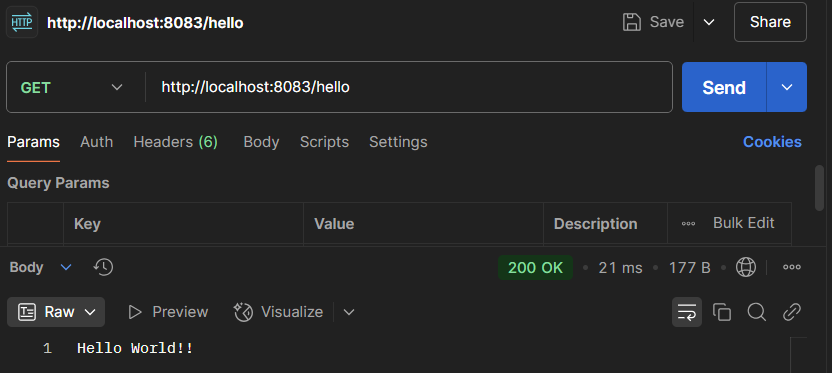
System.out.println("Application started...");

}

}

**OUTPUT:**





**REST - Country Web Service:**

**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"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/util

https://www.springframework.org/schema/util/spring-util.xsd">

<bean id="IN" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="US" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="JP" class="com.cognizant.springlearn.Country">

<property name="code" value="JP"/>

<property name="name" value="Japan"/>

</bean>

<bean id="DE" class="com.cognizant.springlearn.Country">

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

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

</bean>

</beans>

**src/main/java/com/cognizant/springlearn/controller/CountryController.java**

**CountryController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.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.\*;

import java.util.List;

@RestController

@RequestMapping("/country")

public class CountryController {

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

@GetMapping("/{code}")

public Country getCountryByCode(@PathVariable String code) {

LOGGER.debug("Start getCountryByCode: {}", code);

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

Country country;

try {

country = context.getBean(code.toUpperCase(), Country.class);

} catch (Exception e) {

throw new RuntimeException("Country not found for code: " + code);

}

LOGGER.debug("End getCountryByCode: {}", country);

return country;

}

@GetMapping

public List<Country> getAllCountries() {

LOGGER.debug("Start getAllCountries");

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

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

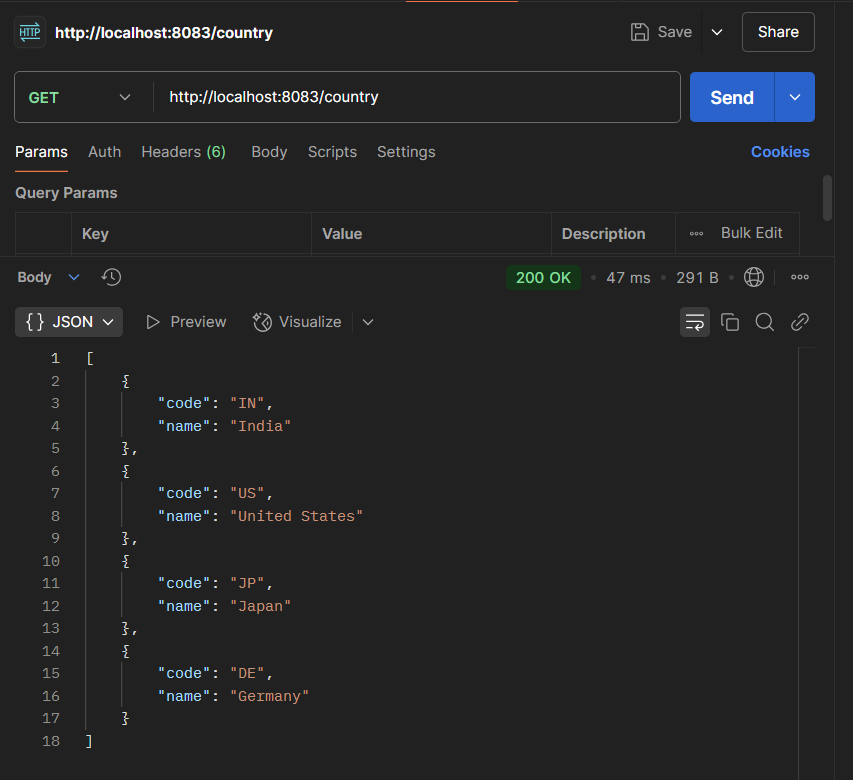
LOGGER.debug("End getAllCountries: {}", countries);

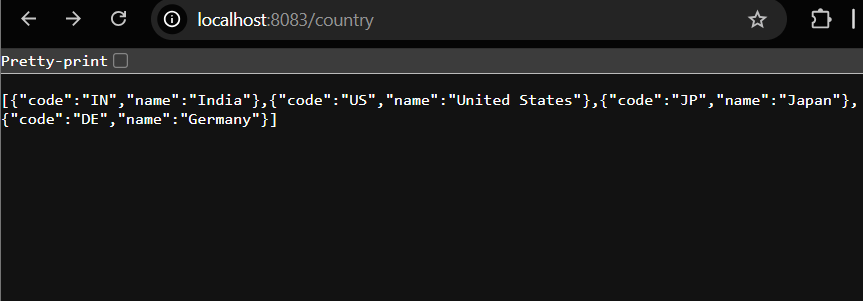
return countries;

}

}

**OUTPUT:**



****

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

**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"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/util

https://www.springframework.org/schema/util/spring-util.xsd">

<bean id="IN" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="US" class="com.cognizant.springlearn.Country">

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

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

</bean>

<bean id="JP" class="com.cognizant.springlearn.Country">

<property name="code" value="JP"/>

<property name="name" value="Japan"/>

</bean>

<bean id="DE" class="com.cognizant.springlearn.Country">

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

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

</bean>

</beans>

**CountryController.java:**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.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.\*;

import java.util.List;

@RestController

@RequestMapping("/country")

public class CountryController {

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

@GetMapping("/{code}")

public Country getCountryByCode(@PathVariable String code) {

LOGGER.debug("Start getCountryByCode: {}", code);

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

Country country;

try {

country = context.getBean(code.toUpperCase(), Country.class);

} catch (Exception e) {

throw new RuntimeException("Country not found for code: " + code);

}

LOGGER.debug("End getCountryByCode: {}", country);

return country;

}

@GetMapping

public List<Country> getAllCountries() {

LOGGER.debug("Start getAllCountries");

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

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

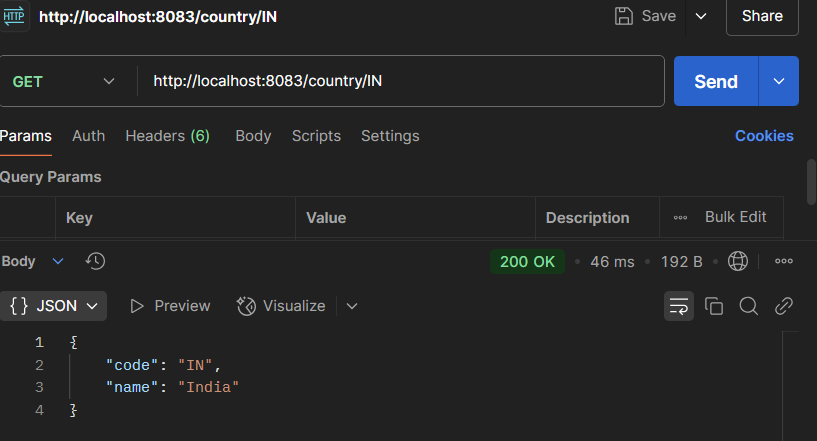
LOGGER.debug("End getAllCountries: {}", countries);

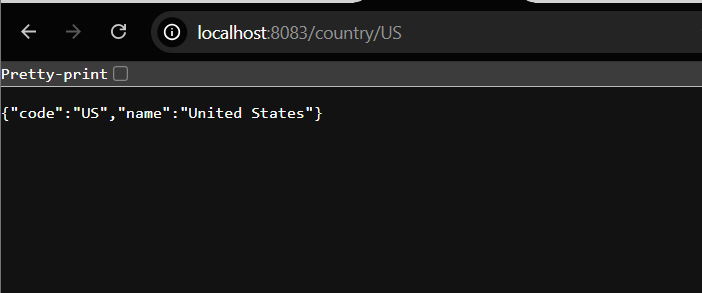
return countries;

}

}

**OUTPUT:**

****

****

**Create authentication service that returns JWT**

**pom.xml:**

**1. Add Required Dependencies**

<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>

<!-- Spring Security -->

<dependency>

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

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

</dependency>

**SecurityConfig.java:**

package com.cognizant.springlearn.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.Customizer;

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

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

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.provisioning.InMemoryUserDetailsManager;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public UserDetailsService userDetailsService() {

return new InMemoryUserDetailsManager(

User.withUsername("user").password("{noop}pwd").roles("USER").build(),

User.withUsername("admin").password("{noop}adminpwd").roles("ADMIN").build()

);

}

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

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

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").hasAnyRole("USER", "ADMIN")

.anyRequest().authenticated()

)

.httpBasic(Customizer.withDefaults());

return http.build();

}

}

**AuthController.java:**

package com.cognizant.springlearn.controller;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import java.util.Base64;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

@RestController

public class AuthController {

private Logger logger = LoggerFactory.getLogger(AuthController.class);

@GetMapping("/authenticate")

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

logger.info("START: authenticate()");

logger.debug("Authorization Header: {}", authHeader);

String user = getUser(authHeader);

logger.debug("Decoded User: {}", user);

String token = generateJwt(user);

Map<String, String> map = new HashMap<>();

map.put("token", token);

logger.info("END: authenticate()");

return map;

}

private String getUser(String authHeader) {

logger.info("START: getUser()");

String base64Credentials = authHeader.substring("Basic".length()).trim();

byte[] decoded = Base64.getDecoder().decode(base64Credentials);

String decodedString = new String(decoded);

String username = decodedString.split(":")[0];

logger.debug("Extracted Username: {}", username);

logger.info("END: getUser()");

return username;

}

private String generateJwt(String user) {

return Jwts.builder()

.setSubject(user)

.setIssuedAt(new Date())

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

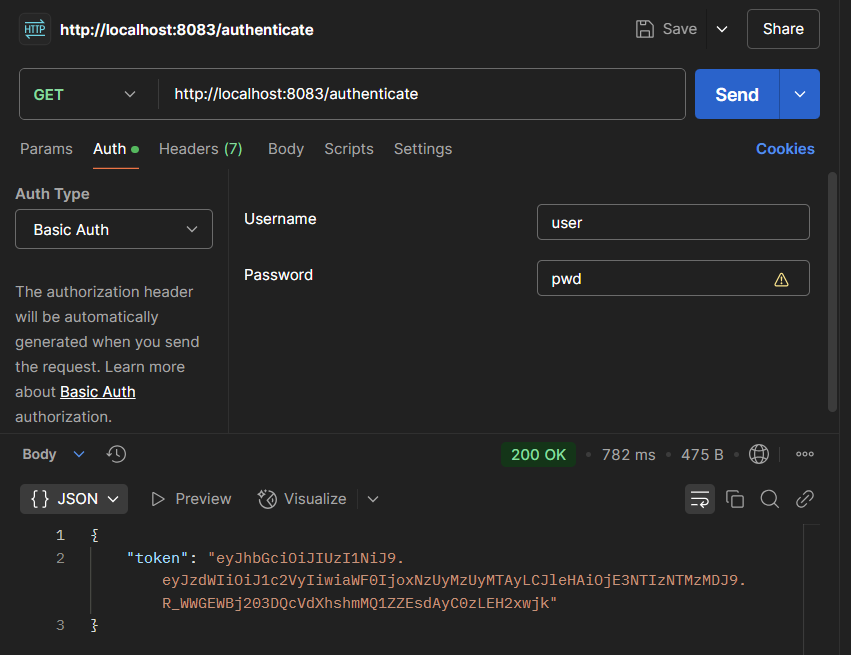
.signWith(Keys.hmacShaKeyFor("mysecretkeymysecretkeymysecretkey!".getBytes()), SignatureAlgorithm.HS256)

.compact();

}

}

**OUTPUT:**

****