**Digital Nurture 4.0 Deep Skilling - Java FSE**  
**WEEK –4 Hands-on Exercises**  
**Module 7 - Spring REST using Spring Boot 3**

**1. spring-rest-handson**

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

**PROGRAM**

**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>springlearn</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>springlearn</name>

<description>Demo project for Spring REST</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>21</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-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</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>

**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);

System.out.println("\n\nThis is a simple Spring Application");

}

}

**SpringlearnApplicationTests.java**

package com.cognizant.springlearn;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SpringlearnApplicationTests {

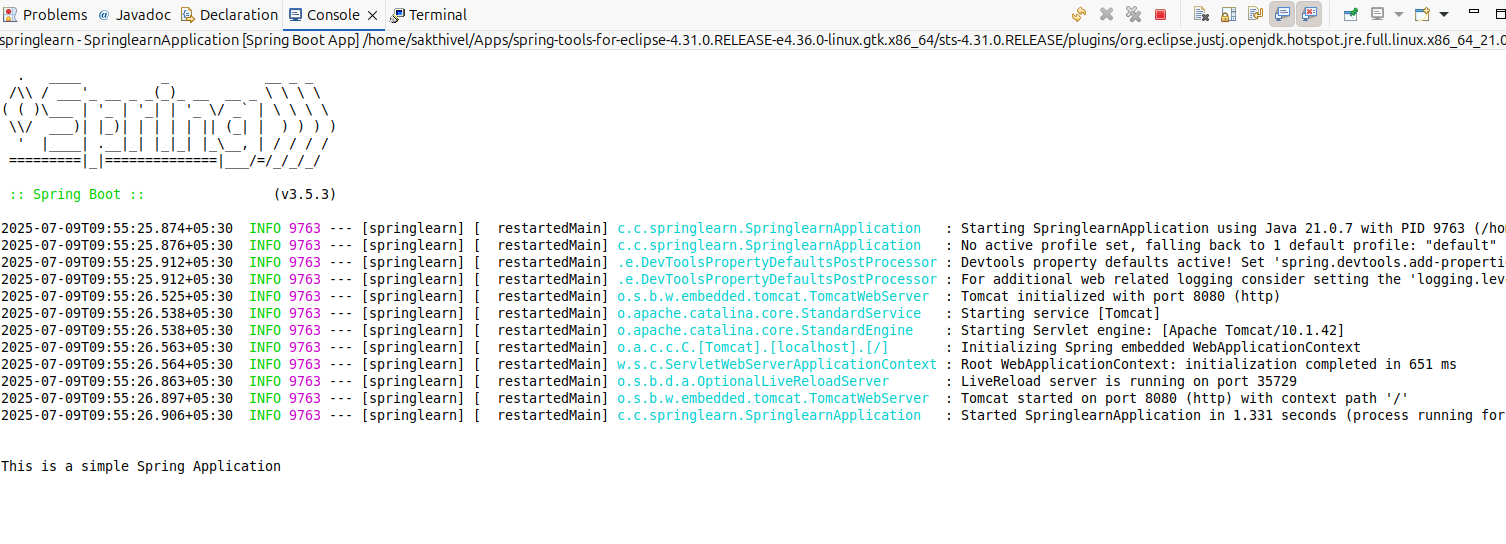
@Test

void contextLoads() {

}

}

**OUTPUT:**

****

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

**PROGRAM**

**SpringLearnApplication.java**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.text.SimpleDateFormat;

import java.util.Date;

@SpringBootApplication

public class SpringLearnApplication {

public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

displayDate();

}

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();

System.out.println(date);

}catch (Exception e) {

System.out.println(e.getMessage());

}

}

}

**SpringLearnApplicationTests.java**

package com.cognizant.springlearn;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SpringLearnApplicationTests {

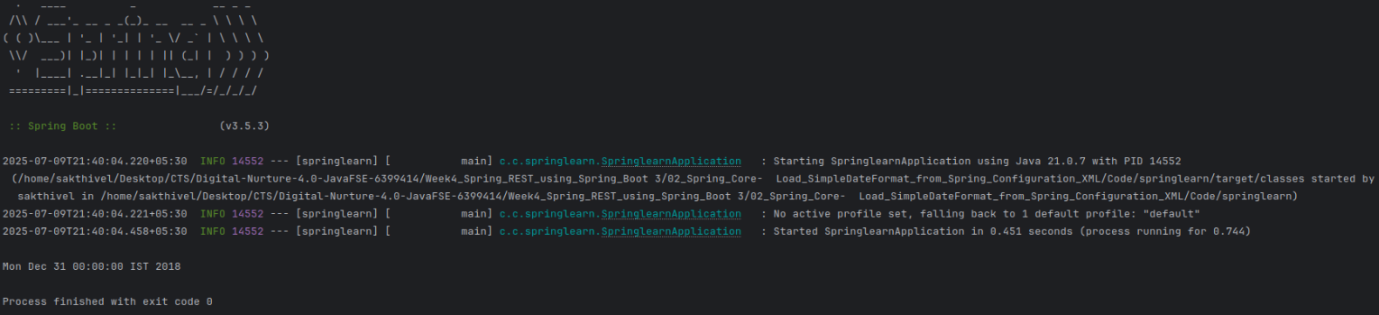
@Test

void contextLoads() {

}

}

**OUTPUT:**

****

**2. spring-rest-handson**

**a. Hello World RESTful Web Service**

**PROGRAM**

**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);

}

}

**HelloAspect.java**

package com.cognizant.springlearn.aspect;

import com.cognizant.springlearn.SpringLearnApplication;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class HelloAspect {

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

@Before("execution(String com.cognizant.springlearn.controller.HelloController.sayHello())")

public void beforeHelloLogger() {

LOGGER.info("Before sayHello() Method");

}

@After("execution(String com.cognizant.springlearn.controller.HelloController.sayHello())")

public void afterHelloLogger() {

LOGGER.info("After sayHello() Method");

}

}

**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 {

@GetMapping("/hello")

public String sayHello() {

return "Hello World!!";

}

}

**SpringLearnApplicationTests.java**

package com.cognizant.springlearn;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SpringLearnApplicationTests {

@Test

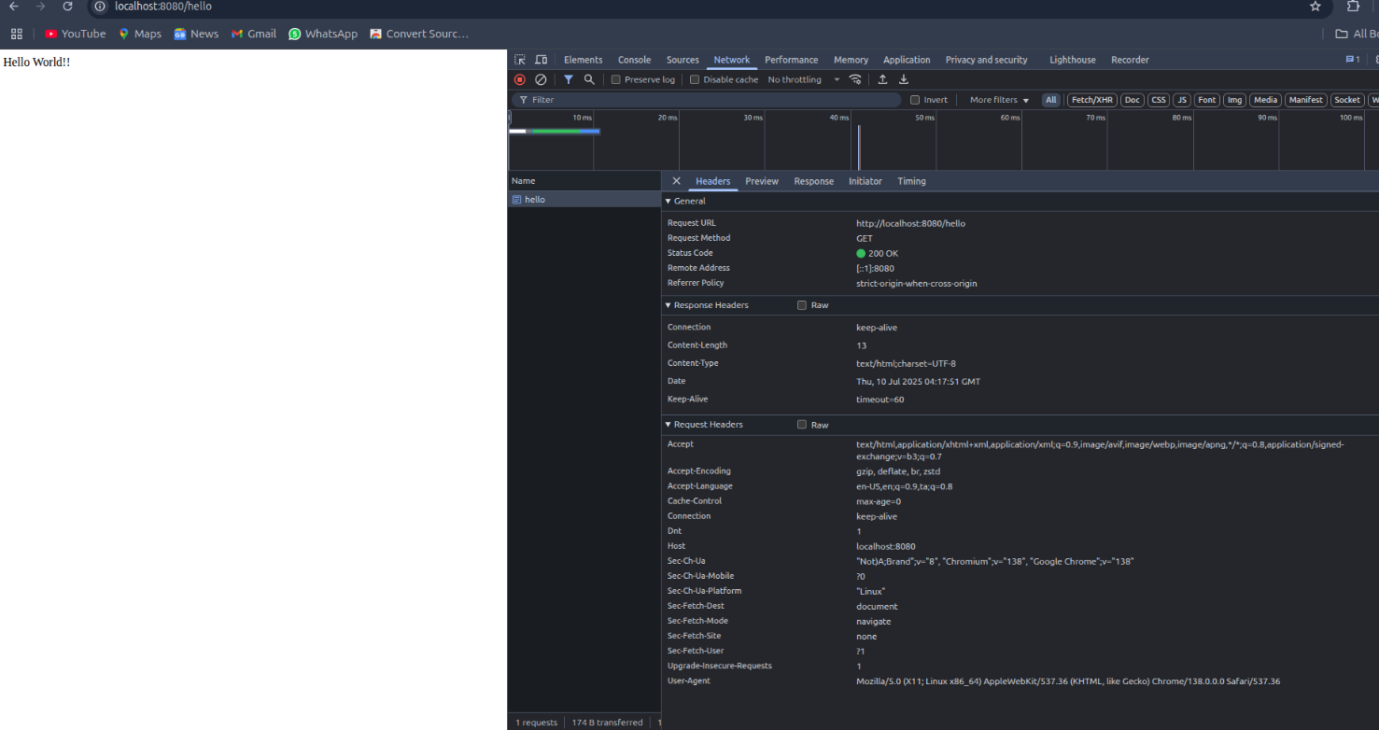
void contextLoads() {

}

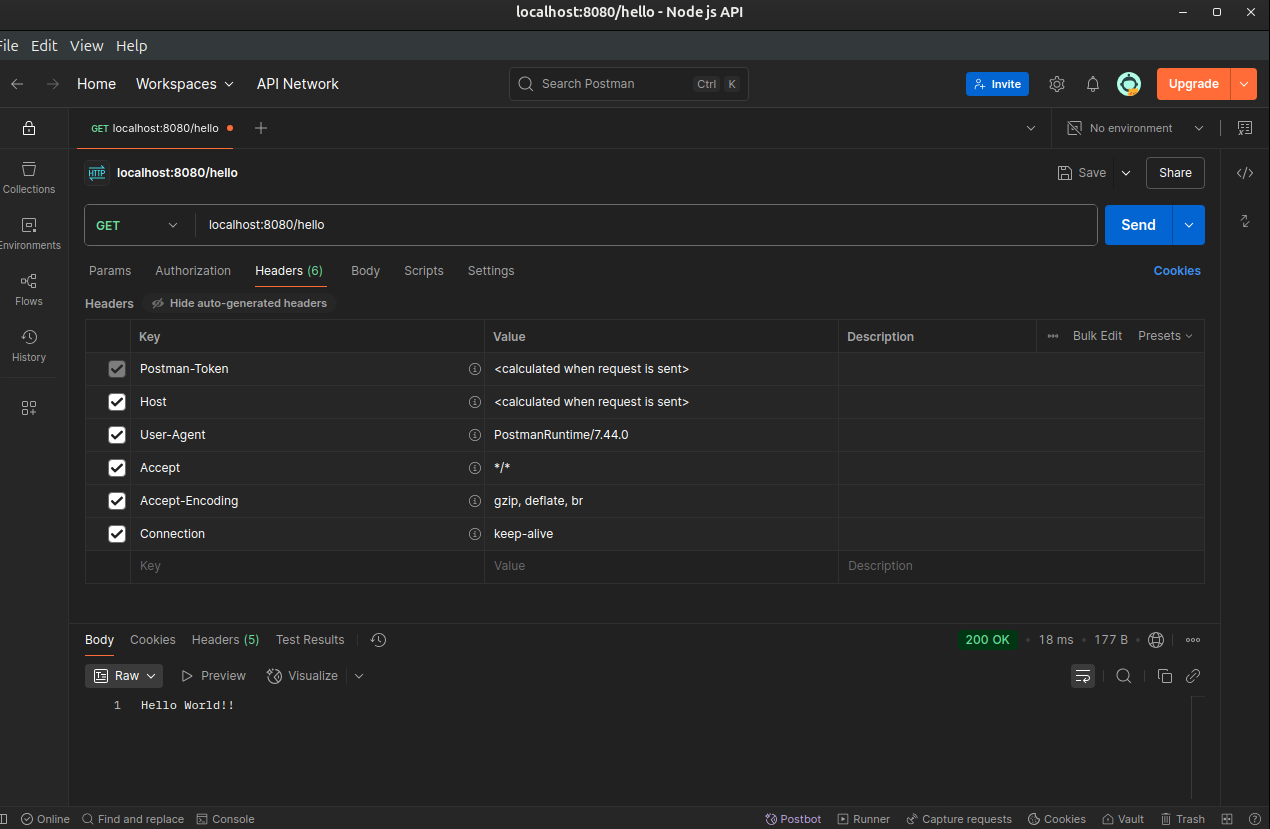
}

**OUTPUT:**

**Browser.png**

****

**Postman.png**

****

**Terminal.png**

****

**b. REST - Country Web Service**

**PROGRAM**

**CountryController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.model.Country;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

@GetMapping("/country")

public Country getCountryIndia() {

return new ClassPathXmlApplicationContext("springConfig.xml").getBean("india", Country.class);

}

}

**Country.java**

package com.cognizant.springlearn.model;

public class Country {

private String code;

private String name;

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);

}

}

**springConfig.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="india" class="com.cognizant.springlearn.model.Country">

<constructor-arg name="code" value="IN" />

<constructor-arg name="name" value="India" />

</bean>

</beans>

**SpringLearnApplicationTests.java**

package com.cognizant.springlearn;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

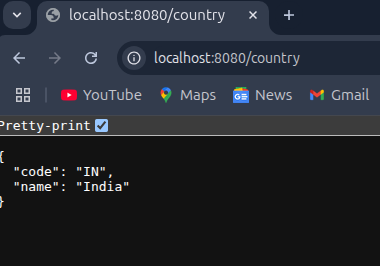
class SpringLearnApplicationTests {

@Test

void contextLoads() {

}}

**OUTPUT:**

****

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

**PROGRAM**

**CountryController.java**

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.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CountryController {

private CountryService countryService;

public CountryController(@Autowired CountryService countryService) {

this.countryService = countryService;

}

@GetMapping("/country/{code}")

public Country getCountryByCode(@PathVariable String code) {

return countryService.getCountryByCode(code);

}

}

**Country.java**

package com.cognizant.springlearn.model;

import lombok.\*;

@Getter @Setter

@NoArgsConstructor @AllArgsConstructor

@ToString

public class Country {

private String code;

private String name;

}

**CountryRepository.java**

package com.cognizant.springlearn.repository;

import com.cognizant.springlearn.model.Country;

import org.springframework.beans.factory.annotation.Autowired;

import java.util.ArrayList;

import java.util.List;

public class CountryRepository {

private List<Country> countries;

public CountryRepository( @Autowired List<Country> countries) {

this.countries = new ArrayList<>(countries);

}

public Country findByCode(String code) {

for(int i=0;i<countries.size();i++)

if(countries.get(i).getCode().equalsIgnoreCase(code))

return countries.get(i);

return new Country();

}

}

**CountryService.java**

package com.cognizant.springlearn.service;

import com.cognizant.springlearn.model.Country;

import com.cognizant.springlearn.repository.CountryRepository;

import org.springframework.beans.factory.annotation.Autowired;

public class CountryService {

private CountryRepository countryRepository;

public CountryService(@Autowired CountryRepository countryRepository) {

this.countryRepository = countryRepository;

}

public Country getCountryByCode(String code) {

return countryRepository.findByCode(code);

}

}

**SpringLearnApplication.java**

package com.cognizant.springlearn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.ImportAutoConfiguration;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ImportResource;

import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication

@ImportResource("classpath:country.xml")

public class SpringLearnApplication {

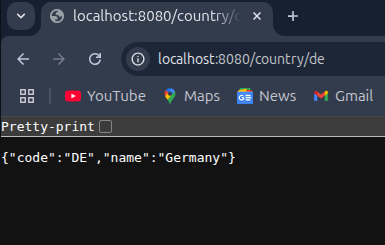
public static void main(String[] args) {

SpringApplication.run(SpringLearnApplication.class, args);

}

}

**OUTPUT:**

****

**5. JWT-handson**

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

**PROGRAM**

**SecurityConfig.java**

package com.cognizant.springlearn.config;

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.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

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

.authorizeHttpRequests(auth -> auth

.requestMatchers("/authenticate").permitAll()

.anyRequest().authenticated()

);

return http.build();

}

}

**AuthenticationController.java**

package com.cognizant.springlearn.controller;

import com.cognizant.springlearn.util.JwtUtil;

import jakarta.servlet.http.HttpServletRequest;

import org.springframework.http.ResponseEntity;

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

import java.util.Base64;

@RestController

public class AuthenticationController {

private final JwtUtil jwtUtil;

public AuthenticationController(JwtUtil jwtUtil) {

this.jwtUtil = jwtUtil;

}

@GetMapping("/authenticate")

public ResponseEntity<?> authenticate(HttpServletRequest request) {

String authHeader = request.getHeader("Authorization");

if (authHeader != null && authHeader.startsWith("Basic ")) {

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

String credentials = new String(Base64.getDecoder().decode(base64Credentials));

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");

}

}

**JwtUtil.java**

package com.cognizant.springlearn.util;

import io.jsonwebtoken.Jwts;

import io.jsonwebtoken.SignatureAlgorithm;

import io.jsonwebtoken.security.Keys;

import org.springframework.stereotype.Component;

import java.security.Key;

import java.util.Date;

@Component

public class JwtUtil {

private final Key secretKey = Keys.secretKeyFor(SignatureAlgorithm.HS256);

public String generateToken(String username) {

long now = System.currentTimeMillis();

return Jwts.builder()

.setSubject(username)

.setIssuedAt(new Date(now))

.setExpiration(new Date(now + 1000 \* 60 \* 10))

.signWith(secretKey)

.compact();

}

}

**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);

}

}

**SpringLearnApplicationTests.java**

package com.cognizant.springlearn;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

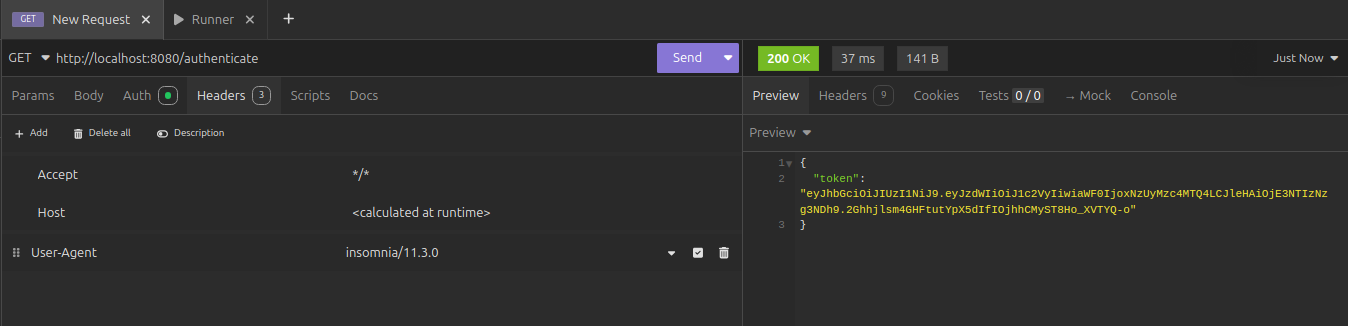
class SpringLearnApplicationTests {

@Test

void contextLoads() {

}}

**OUTPUT:**

****