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

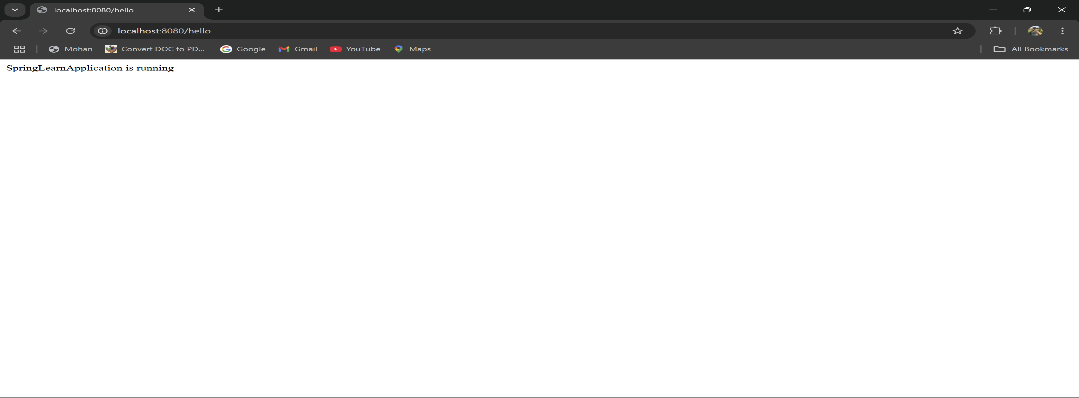
PROGRAM:

package org.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("Spring is running..");  
 SpringApplication.*run*(SpringLearnApplication.class,args);  
 }  
}

HELLOCONTROLLER.java:

package org.cognizant;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HelloController {  
 @GetMapping("/hello")  
 public String sayHello(){  
 return "SpringLearnApplication is running";  
 }  
}

**OUTPUT**:

****

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

**PROGRAM:**

**XML:**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:util="http://www.springframework.org/schema/util"  
 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  
 http://www.springframework.org/schema/util https://www.springframework.org/schema/util/spring-util.xsd">  
  
 <bean id="country" class="org.cognizant.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
</beans>

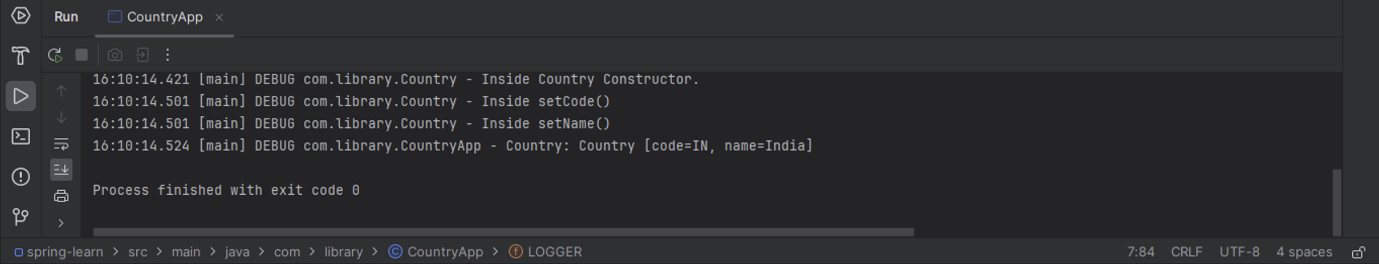
**Country:**

package org.cognizant;  
  
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 getName()");  
 this.name=name;  
 }  
 @Override  
 public String toString(){  
 return "Country [code=" +code+",name="+name+"]";  
 }  
}

**SpringLearnApplication:**

package org.cognizant;  
  
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");  
 ApplicationContext context = new ClassPathXmlApplicationContext("Country.xml");  
 Country country = (Country) context.getBean("country");  
 *LOGGER*.debug("Country: {}", country);  
 ((ClassPathXmlApplicationContext) context).close();  
 *LOGGER*.debug("End");  
 }  
}

**OUTPUT:**



**3. Hello World RESTful Web Service**

**PROGRAM:**

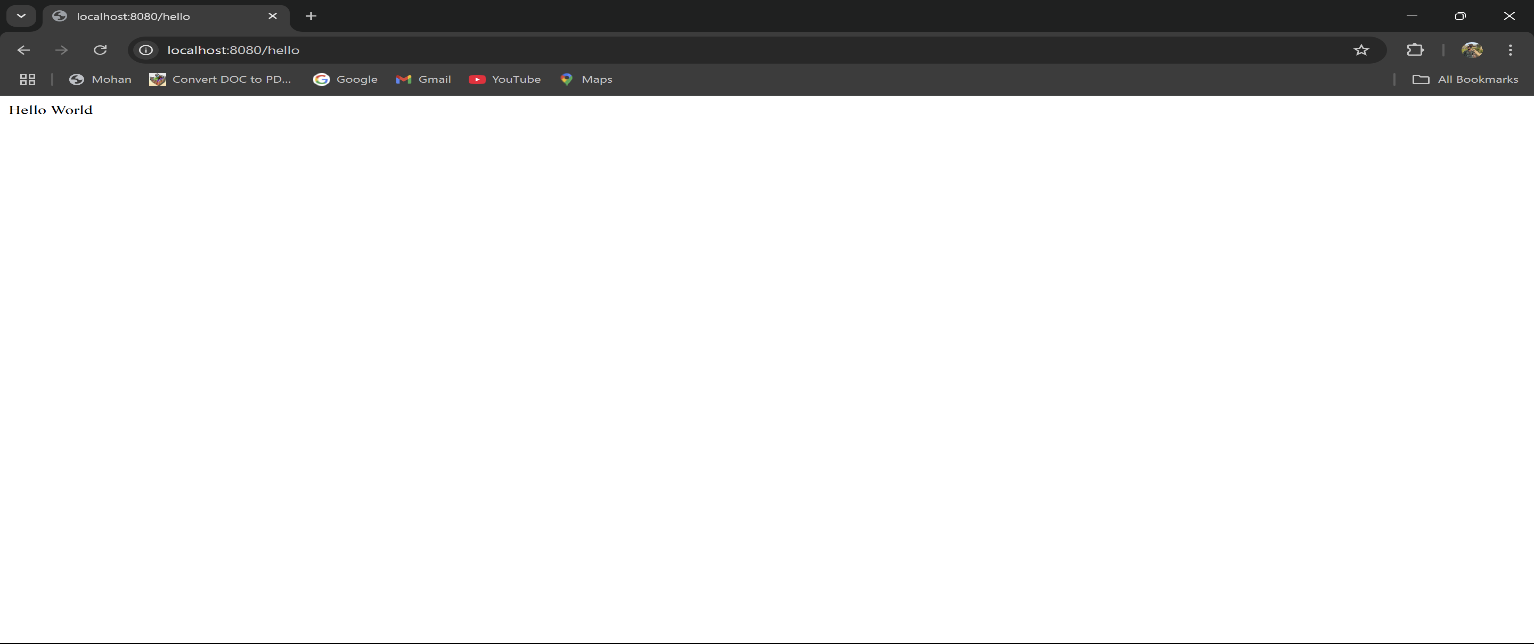
**SpringRestApiApplication :**

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

**HelloController:**

package com.example.springrestapi;  
  
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");  
 String message = "Hello World";  
 *LOGGER*.debug("End");  
 return message;  
 }  
}

**OUTPUT:**



**4. REST - Country Web Service**

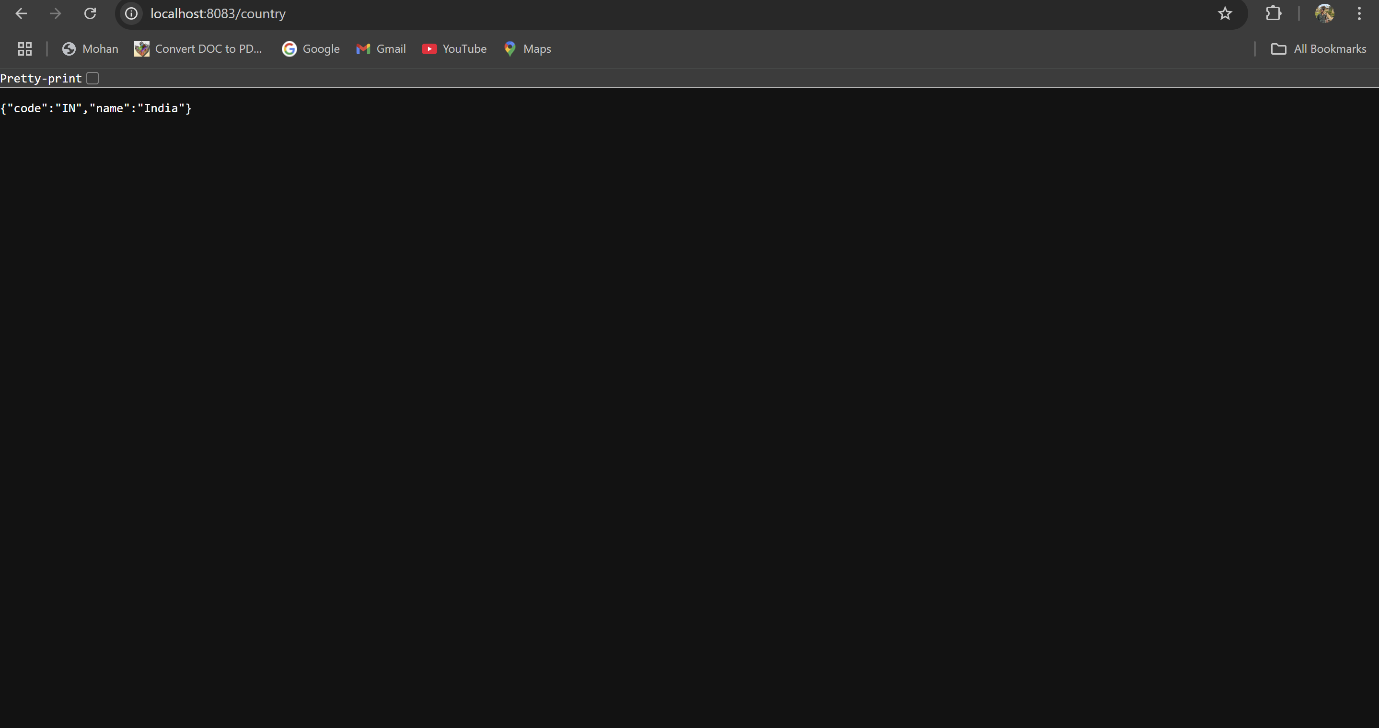
**PROGRAM:**

package com.example.springrestapi;  
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(){  
 ApplicationContext contex =new ClassPathXmlApplicationContext("country.xml");  
 Country country=(Country)contex.getBean("in");  
 return country;  
 }  
}

COUNTRY .JAVA:

package com.example.springrestapi;  
  
public class Country {  
 private String code;  
 private String 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:**

****

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

**PROGRAM:**

**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="in" class="com.example.springrestapi.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 </list>  
 </constructor-arg>  
 </bean>  
</beans>

**COUNTRY.JAVA:**

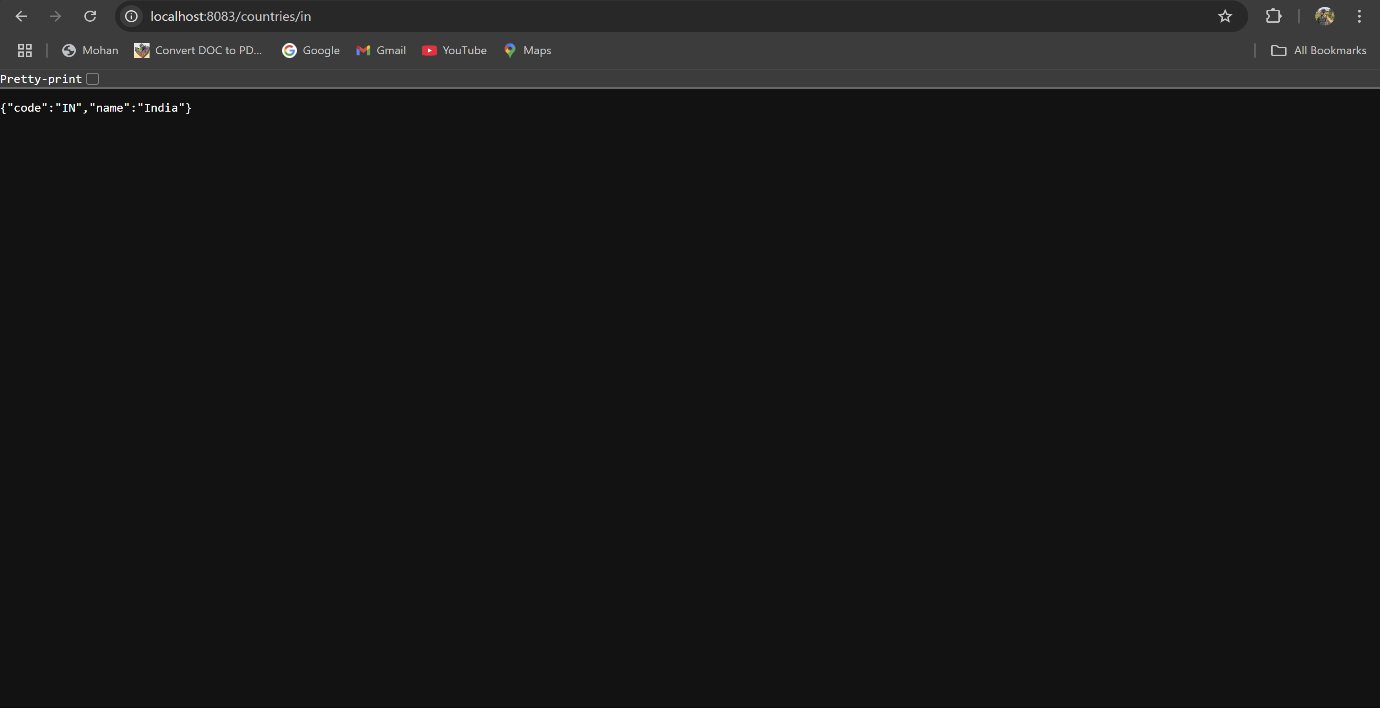
package com.example.springrestapi;  
  
public class Country {  
 private String code;  
 private String name;  
 public String getCode(){  
 return code;  
 }  
 public void setCode(String code){  
 this.code=code;  
 }  
 public String getName(){  
 return name;  
 }  
 public void setName(String name){  
 this.name=name;  
 }  
}

**COUNTRYCONTROLLER:**

package com.example.springrestapi;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
  
import java.util.List;

@RestController  
public class CountryController {  
 @GetMapping("/countries/{code}")  
 public Country getCountry(@PathVariable String code){  
 ApplicationContext context =new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList =context.getBean("countryList", List.class);  
 return countryList.stream().filter(country -> country.getCode().equalsIgnoreCase(code)).findFirst().orElseThrow(()-> new RuntimeException("Country is not found"));  
 }  
}

**OUTPUT:**

****

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

**PROGRAM:**

**SECURITYCONFIG.JAVA:**

package com.example.springrestapi;  
  
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.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.config.annotation.web.configurers.HttpBasicConfigurer;  
import org.springframework.security.core.userdetails.User;  
import org.springframework.security.core.userdetails.UserDetails;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.provisioning.InMemoryUserDetailsManager;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.stereotype.Component;  
  
@Configuration  
@EnableWebSecurity  
public class SecurityConfig {  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http.csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(authz -> authz  
 .requestMatchers("/authenticate").authenticated()  
 .anyRequest().permitAll()  
 )  
 .httpBasic(Customizer.*withDefaults*());  
  
 return http.build();  
 }  
 @Bean  
 public UserDetailsService users() {  
 UserDetails user = User  
 .*withUsername*("user")  
 .password("{noop}pwd")   
 .roles("USER")  
 .build();  
 return new InMemoryUserDetailsManager(user);  
 }  
}

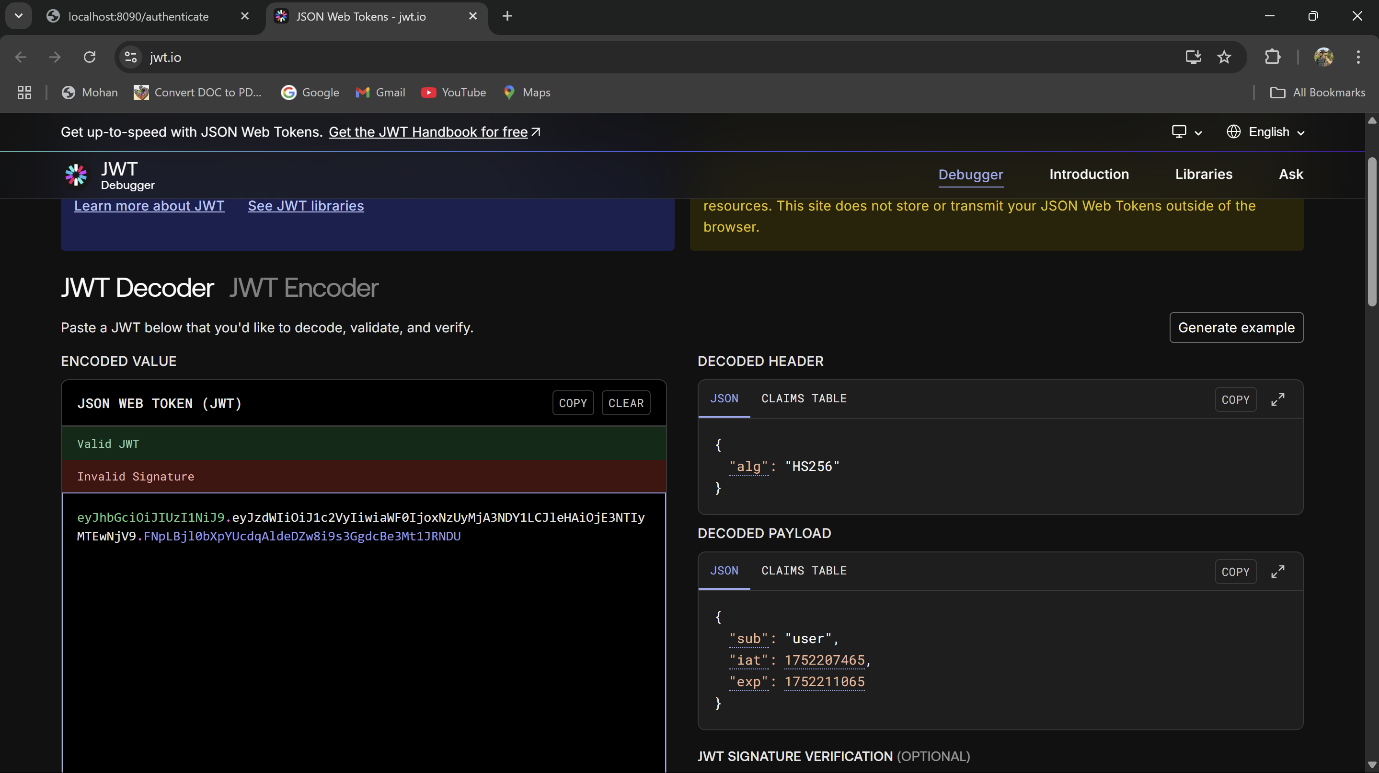
**JWTUNIT.JAVA:**

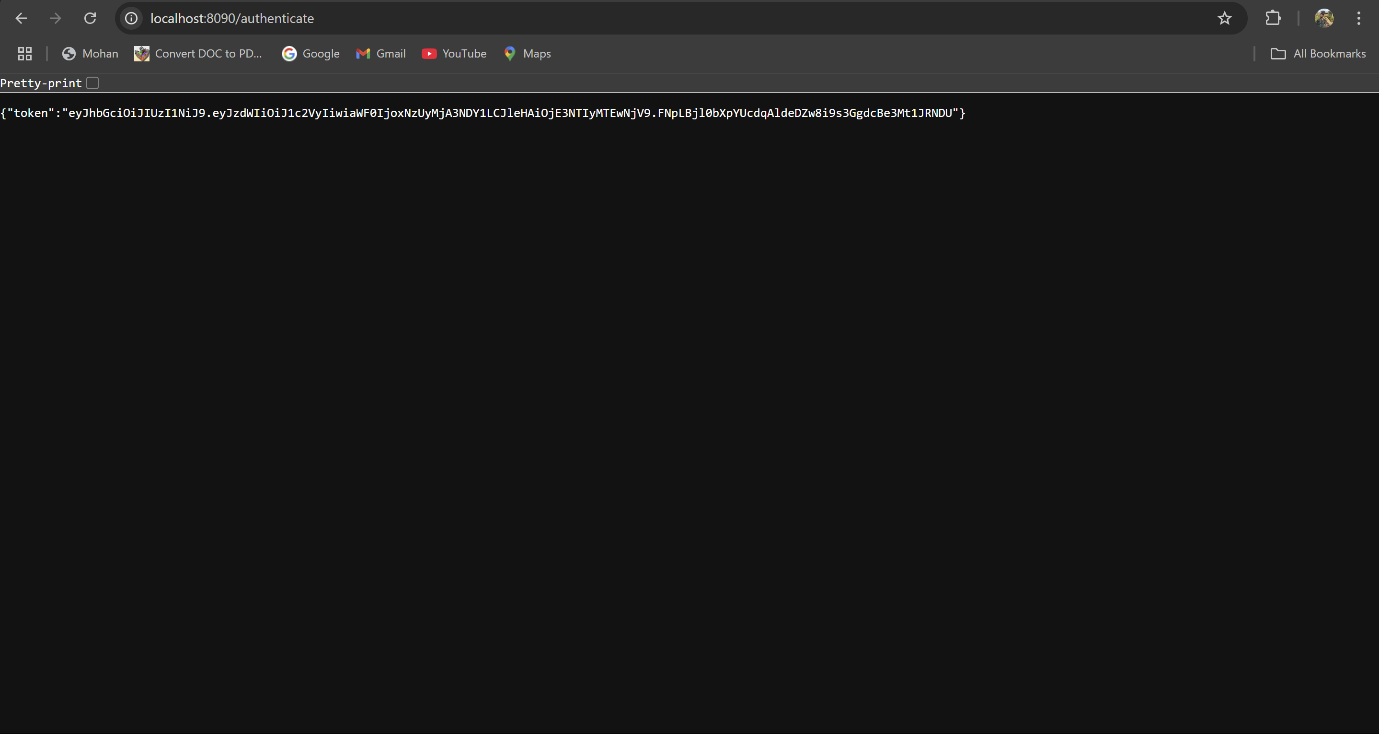
package com.example.springrestapi;  
  
import io.jsonwebtoken.Jwts;  
import io.jsonwebtoken.SignatureAlgorithm;  
import io.jsonwebtoken.security.Keys;  
import org.springframework.stereotype.Component;  
  
import java.security.Key;  
import java.util.Date;  
  
  
@Component  
public class JwtUtil {  
 private static final String *secret* = "mysupersecretkeymysupersecretkey";  
 private static final Key *key* = Keys.*hmacShaKeyFor*(*secret*.getBytes());  
 public String generateToken(String username) {  
 return Jwts.*builder*()  
 .setSubject(username)  
 .setIssuedAt(new Date())  
 .setExpiration(new Date(System.*currentTimeMillis*()+3600000))  
 .signWith(SignatureAlgorithm.*HS256*, *key*)  
 .compact();  
 }  
}

**AUTHCONTROLLER:**

package com.example.springrestapi;  
  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
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.nio.charset.StandardCharsets;  
import java.util.Base64;  
import java.util.HashMap;  
import java.util.Map;  
  
@RestController  
public class AuthController {  
 @Autowired  
 private JwtUtil jwtUtil;  
  
 @GetMapping("/authenticate")  
 public ResponseEntity<Map<String, String>>  
 authenticate(@RequestHeader("Authorization") String authHeader) {  
 if (authHeader == null || !authHeader.startsWith("Basic ")) {  
 return ResponseEntity.*status*(HttpStatus.*BAD\_REQUEST*).build();  
 }  
 try {  
 String base64Credentials = authHeader.substring(6);  
 byte[] credDecoded = Base64.*getDecoder*().decode(base64Credentials);  
 String credentials = new String(credDecoded, StandardCharsets.*UTF\_8*);  
 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);  
 Map<String, String> response = new HashMap<>();  
 response.put("token", token);  
 return ResponseEntity.*ok*(response);  
 } else {  
 return ResponseEntity.*status*(HttpStatus.*UNAUTHORIZED*).build();  
 }  
 } catch (Exception e){  
 return ResponseEntity.*badRequest*().build();  
 }  
 }  
}

**OUTPUT:**

****

****