**Handson Exercise 1 - Securing RESTful Web Services with Spring Security**

**CODE:**

**Country Class: -**

package com.cognizant.ex4handson2;  
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;  
 }  
}

**Country.xml file: -**

<?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="in" class="com.cognizant.ex4handson2.Country">  
 <property name="code" value="IN"/>  
 <property name="name" value="India"/>  
 </bean>  
  
 <bean id="us" class="com.cognizant.ex4handson2.Country">  
 <property name="code" value="US"/>  
 <property name="name" value="United States"/>  
 </bean>  
  
 <bean id="jp" class="com.cognizant.ex4handson2.Country">  
 <property name="code" value="JP"/>  
 <property name="name" value="Japan"/>  
 </bean>  
  
 <bean id="de" class="com.cognizant.ex4handson2.Country">  
 <property name="code" value="DE"/>  
 <property name="name" value="Germany"/>  
 </bean>  
  
 <!-- List of all countries -->  
 <bean id="countryList" class="java.util.ArrayList">  
 <constructor-arg>  
 <list>  
 <ref bean="in"/>  
 <ref bean="us"/>  
 <ref bean="jp"/>  
 <ref bean="de"/>  
 </list>  
 </constructor-arg>  
 </bean>  
</beans>

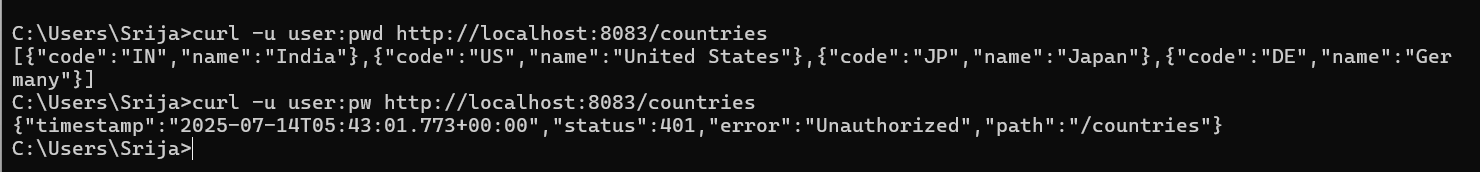
**CountryController: -**

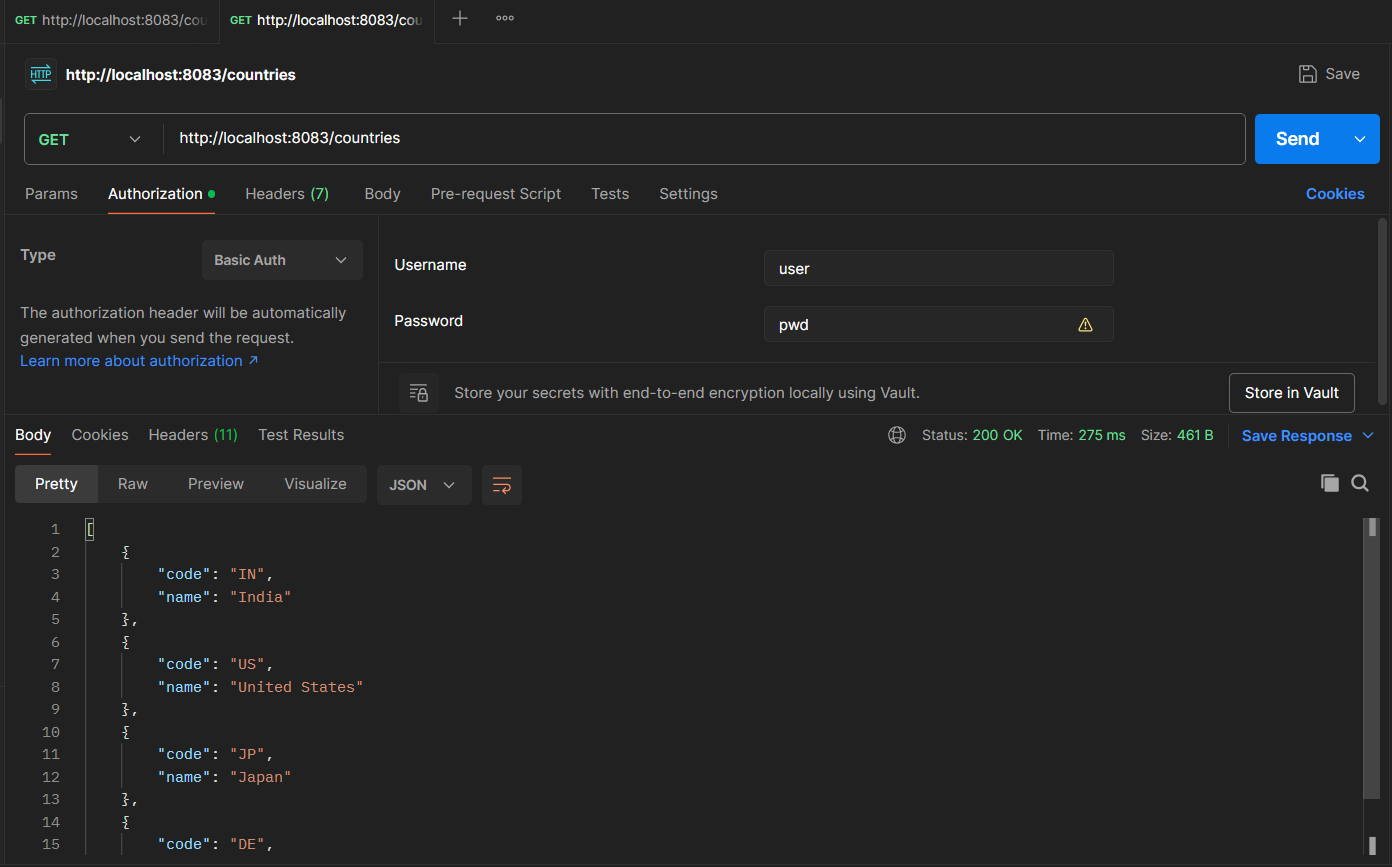
package com.cognizant.ex4handson2;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.web.bind.annotation.\*;   
@RestController  
public class CountryController {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CountryController.class);  
 @GetMapping("/countries")  
 public List<Country> getAllCountries(){  
 ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");  
 List<Country> countryList = (List<Country>) context.getBean("countryList");  
 return countryList;  
 }  
}

**SecurityConfig: -**

package com.cognizant.ex4handson2;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;  
import org.springframework.security.web.SecurityFilterChain;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
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.crypto.bcrypt.BCryptPasswordEncoder;  
import org.springframework.security.crypto.password.PasswordEncoder;  
  
@Configuration  
@EnableWebSecurity  
public class SecurityConfig {  
  
 @Bean  
 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {  
 http  
 .csrf(csrf -> csrf.disable())  
 .authorizeHttpRequests(auth -> auth  
 .requestMatchers("/countries").authenticated()  
 .anyRequest().permitAll()  
 )  
 .httpBasic();  
 return http.build();  
 }  
  
 @Bean  
 public UserDetailsService users() {  
 UserDetails user = User.*withUsername*("user")  
 .password(passwordEncoder().encode("pwd"))  
 .roles("USER")  
 .build();  
  
 UserDetails admin = User.*withUsername*("admin")  
 .password(passwordEncoder().encode("pwd"))  
 .roles("ADMIN")  
 .build();  
  
 return new InMemoryUserDetailsManager(user, admin);  
 }  
  
 @Bean  
 public PasswordEncoder passwordEncoder() {  
 return new BCryptPasswordEncoder();  
 }  
}

**OUTPUT:**



And I have also tried using postman  


If I type incorrect password/username:

