# Spring Security with JWT

As per document : **5. JWT-handson**

Student Name: Divyansh Tiwari

SuperSet ID: 6394658

Week 4 – Spring REST

## Hands-On: Securing RESTful Web Services with Spring Security

### 1. Open project in IntelliJ IDEA

### 2. Adding Dependencies

Make sure your pom.xml contains the following dependencies:

*<dependency>*

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

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

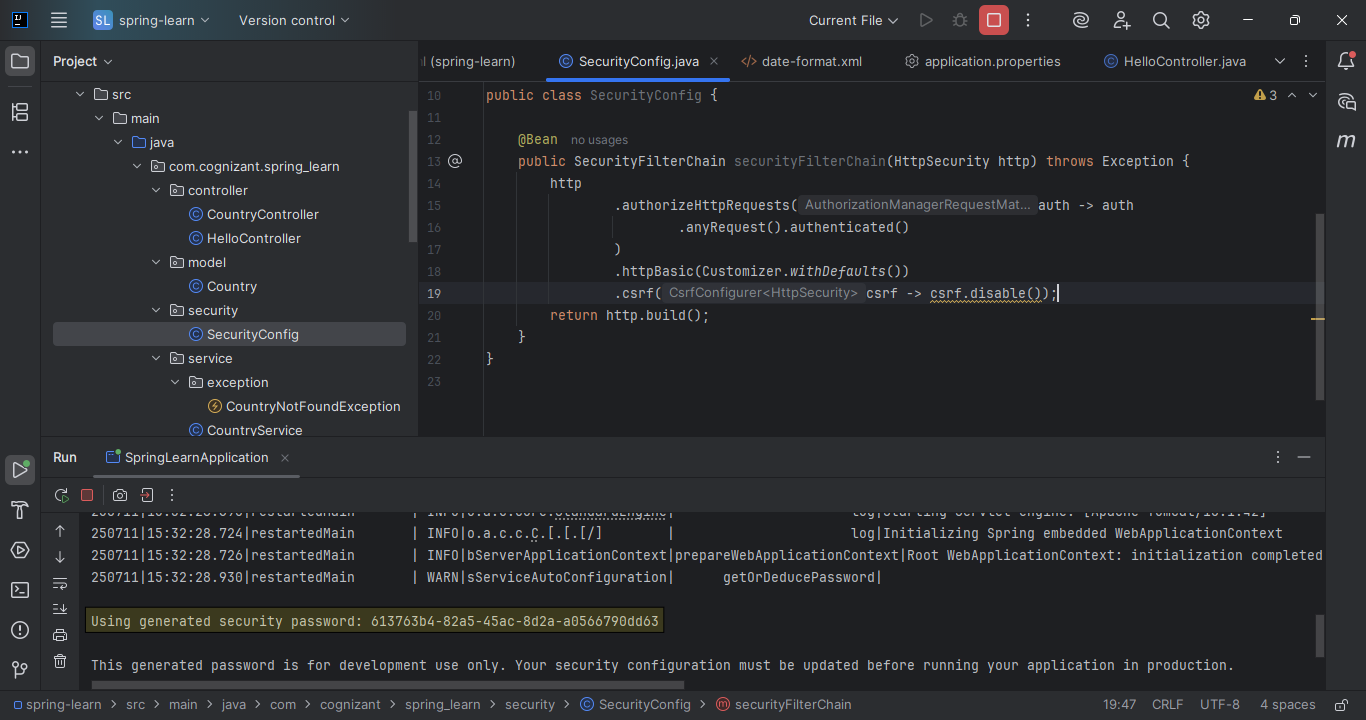
*</dependency>*

* spring-boot-starter-security
* jjwt (api, impl, jackson)

3. Run command “mvn clean package”

This makes sure that new dependencies are included in the project.

**Screenshot :** Console logging application start-up and password generation



## Hands-On: Creating users and roles in Spring Security

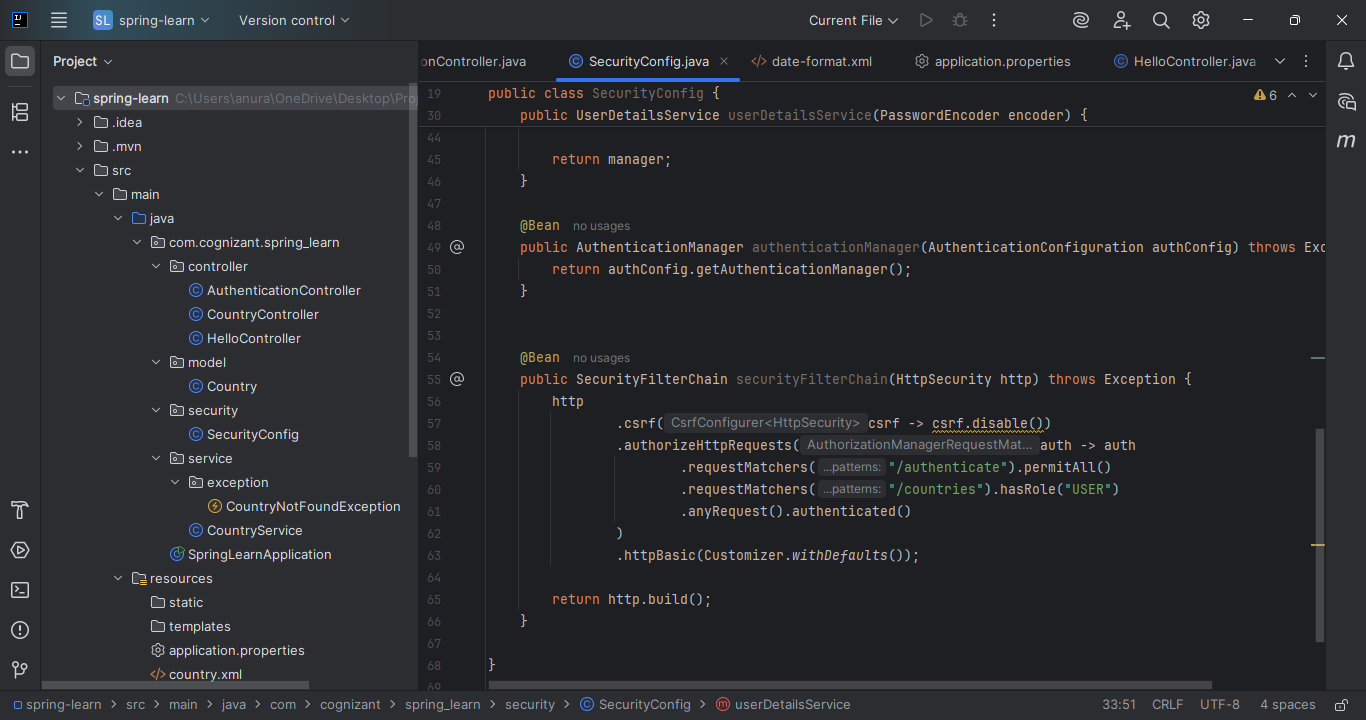
Because WebSecurityConfigurerAdapter is **removed** in Spring Security 6.x, we used **beans and lambdas** for configuration instead of @Override configure().

Defined two in-memory users (user, admin) with password pwd and roles USER and ADMIN respectively.

**Code Location:** SecurityConfig.java

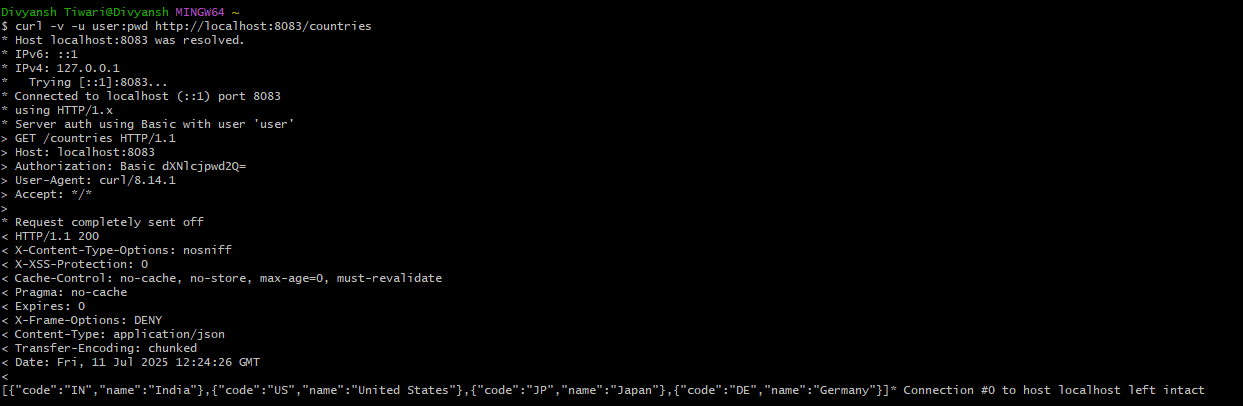
*@Bean  
public UserDetailsService userDetailsService(PasswordEncoder encoder) {  
 InMemoryUserDetailsManager manager = new InMemoryUserDetailsManager();  
 manager.createUser(User.withUsername("admin")  
 .password(encoder.encode("pwd"))  
 .roles("ADMIN")  
 .build());  
 manager.createUser(User.withUsername("user")  
 .password(encoder.encode("pwd"))  
 .roles("USER")  
 .build());  
 return manager;  
}*

**Screenshot :** SecurityConfig class with in-memory users



**Screenshot :** Returned countries list in GitBash

*curl -v -u user:pwd* [*http://localhost:8090/countries*](http://localhost:8090/countries)

**

## Hands-On: Create authentication service that returns JWT  This step involves creating an authentication endpoint that returns a JWT when valid credentials are supplied.

## Only USER role can access /countries. Other endpoints require authentication.

1. Added jjwt 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>*

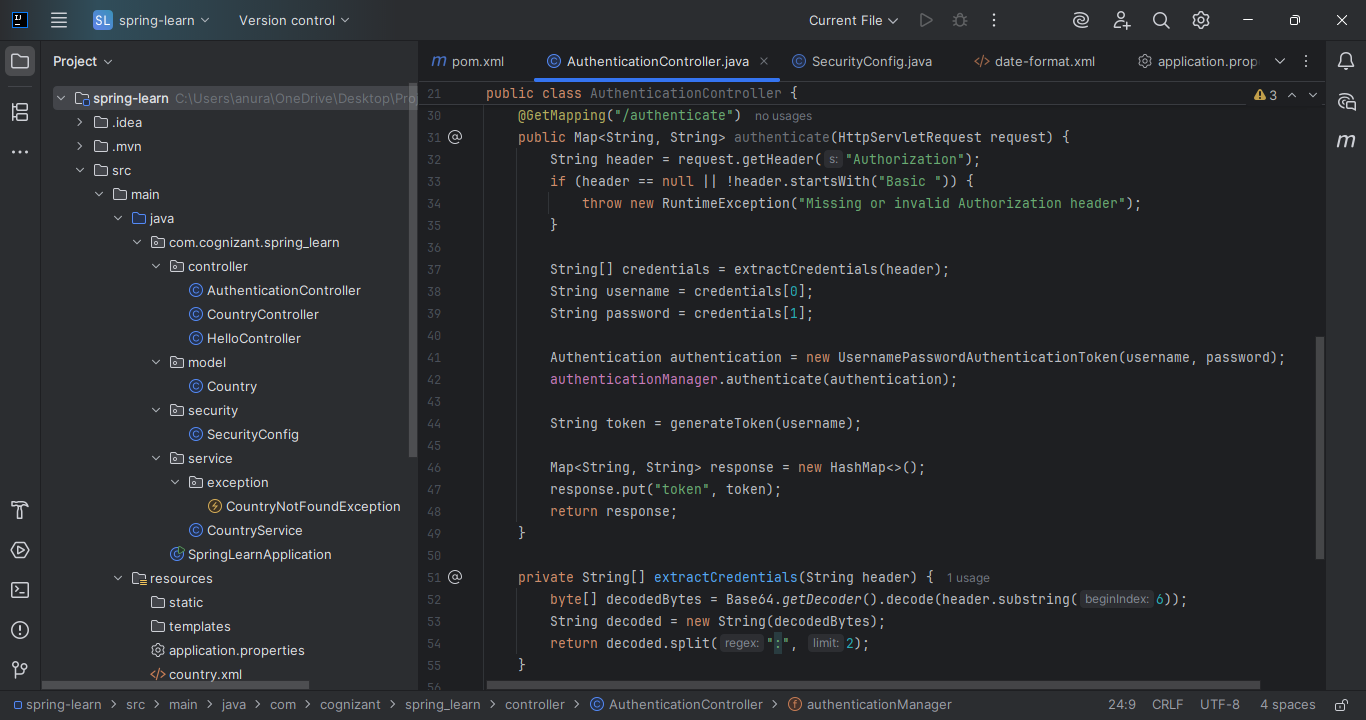
*<artifactId>jjwt-jackson</artifactId>*

*<version>0.11.5</version>*

*<scope>runtime</scope>*

*</dependency>*

1. Created Authentication Controller class:

****

**curl Commands & Expected Results:**

* curl -u user:pwd http://localhost:8083/countries

OUTPUT: Token String



* curl -u user:wrongpassword <http://localhost:8083/countries>

