**Source code for SpringSecurity:**

1. **CustomAuthenticationProvider.java:**

package com.simplilearn.demo;

import org.springframework.security.authentication.\*;

import org.springframework.security.core.\*;

import org.springframework.security.core.authority.SimpleGrantedAuthority;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

public class CustomAuthenticationProvider implements AuthenticationProvider {

List<User> obj = new ArrayList<User>();

public CustomAuthenticationProvider() {

obj.add(new User("vamshi", "vamshi@123", "Vamshi", "ROLE\_USER"));

obj.add(new User("vyshna","vyshna@143", "Vyshna", "ROLE\_ADMIN"));

}

@Override

public Authentication authenticate(Authentication authentication) throws AuthenticationException {

String name = authentication.getName();

String password = authentication.getCredentials().toString();

Optional<User> authenticatedUser = obj.stream().filter(

user -> user.getName().equals(name) && user.getPassword().equals(password)

).findFirst();

if(!authenticatedUser.isPresent()){

throw new BadCredentialsException("Invalid Credentials");

}

List<GrantedAuthority> authorities = new ArrayList<>();

authorities.add(new SimpleGrantedAuthority(authenticatedUser.get().getRole()));

Authentication auth = new UsernamePasswordAuthenticationToken(name, password, authorities);

return auth;

}

@Override

public boolean supports(Class<?> aClass) {

return aClass.equals(UsernamePasswordAuthenticationToken.class);

}

}

1. **SpringSecuirtApplication.java:**

package com.simplilearn.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.ComponentScan;

@SpringBootApplication

@ComponentScan("com.simplilearn.demo")

public class SpringSecurityApplication {

public static void main(String[] args) {

SpringApplication.run(SpringSecurityApplication.class, args);

}

}

1. **User.java:**

package com.simplilearn.demo;

public class User {

private int Id;

private String Name;

private String Email;

private String Password;

private String Role;

public User() { }

public User(String name, String email, String password, String role) {

Name = name;

Email = email;

Password = password;

Role = role;

}

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public String getEmail() {

return Email;

}

public void setEmail(String email) {

Email = email;

}

public String getPassword() {

return Password;

}

public void setPassword(String password) {

Password = password;

}

public String getRole() {

return Role;

}

public void setRole(String role) {

Role = role;

}

public int getId() {

return Id;

}

public void setId(int id) {

Id = id;

}

@Override

public String toString() {

return "User [Id=" + Id + ", Name=" + Name + ", Email=" + Email + ", Password=" + Password + ", Role=" + Role

+ "]";

}

}

1. **UserController.java:**

package com.simplilearn.demo;

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

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

@RestController

public class UserController {

@RequestMapping("/")

public String hello(){

return "Hello World";

}

@RequestMapping("/protected")

public String protectedHello(){

return "Hello World - Protected";

}

@RequestMapping("/admin")

public String admin(){

return "Hello World - Admin";

}

}

1. **UserSecurityConfig:**

package com.simplilearn.demo;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

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

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

@SuppressWarnings("deprecation")

@Configuration

public class UserSecurityConfig extends WebSecurityConfigurerAdapter{

@Override

protected void configure(HttpSecurity http) throws Exception {

http.httpBasic().and().authorizeRequests()

.antMatchers("/").permitAll()

.antMatchers("/protected").hasRole("USER")

.antMatchers("/admin").hasRole("ADMIN");

}

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.authenticationProvider(new CustomAuthenticationProvider());

}

}

1. **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>2.7.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.simplilearn.demo</groupId>

<artifactId>SpringSecurity</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Phase3\_Lesson4\_SpringSecurity</name>

<description>Spring Security Authentication</description>

<properties>

<java.version>1.8</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-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>