**PHASE3-FOURTHPROJECT-SOURCECODE**

**SpringSecurityAuthApplication.java**

**package** com.SpringSecurityAuth;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** SpringSecurityAuthApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringSecurityAuthApplication.**class**, args);

}

}

**ServletInitializer.java**

**package** com.SpringSecurityAuth;

**import** org.springframework.boot.builder.SpringApplicationBuilder;

**import** org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

**public** **class** ServletInitializer **extends** SpringBootServletInitializer {

@Override

**protected** SpringApplicationBuilder configure(SpringApplicationBuilder application) {

**return** application.sources(SpringSecurityAuthApplication.**class**);

}

}

**CustomAuthenticationProvider.java**

**package** com.SpringSecurityAuth;

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

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

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

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

**import** org.springframework.stereotype.Component;

**import** java.util.ArrayList;

@Component

**public** **class** CustomAuthenticationProvider **implements** AuthenticationProvider {

@Override

**public** Authentication authenticate(Authentication authentication)

**throws** AuthenticationException {

String name = authentication.getName();

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

**if** (shouldAuthenticateAgainstThirdPartySystem()) {

// use the credentials

// and authenticate against the third-party system

**return** **new** UsernamePasswordAuthenticationToken(

name, password, **new** ArrayList<>());

} **else** {

**return** **null**;

}

}

**private** **boolean** shouldAuthenticateAgainstThirdPartySystem() {

// **TODO** Auto-generated method stub

**return** **false**;

}

@Override

**public** **boolean** supports(Class<?> authentication) {

**return** authentication.equals(UsernamePasswordAuthenticationToken.**class**);

}

}

**DemoController.java**

**package** com.SpringSecurityAuth;

**import** org.springframework.stereotype.Controller;

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

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

//Authorization

@Controller

**public** **class** DemoController {

@ResponseBody

@RequestMapping("/")

**public** String all() {

**return** "all simplilearn can access";

}

@ResponseBody

@RequestMapping("/admin")

**public** String admin() {

**return** "hi admins";

}

@ResponseBody

@RequestMapping("/user")

**public** String user() {

**return** "hi simplilearn users";

}

@ResponseBody

@RequestMapping("/user1")

**public** String user1() {

**return** "hi simplilearn users";

}

}

**AppSecurityConfig.java**

**package** com.SpringSecurityAuth;

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

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.ComponentScan;

**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.EnableWebSecurity;

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

**import** org.springframework.security.crypto.password.~~NoOpPasswordEncoder~~;

**import** org.springframework.security.crypto.password.PasswordEncoder;

@Configuration

@EnableWebSecurity

@ComponentScan("com.SpringSecurityAuth")

**public** **class** AppSecurityConfig **extends** ~~WebSecurityConfigurerAdapter~~ {

@Autowired

**private** CustomAuthenticationProvider authProvider;

**protected** **void** configure1(AuthenticationManagerBuilder auth) **throws** Exception {

auth.authenticationProvider(authProvider);

}

**protected** **void** configure1(HttpSecurity http) **throws** Exception {

http.authorizeRequests().anyRequest().authenticated()

.and().httpBasic();

}

//authentication

@Override

**protected** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception {

// chain of config

auth.inMemoryAuthentication()

.withUser("admin")

.password("admin")

.roles("ADMIN")

.and()

.withUser("user")

.password("user")

.roles("USER")

.and()

.withUser("user1")

.password("user1")

.roles("USER");

}

@Bean

**public** PasswordEncoder getpassword() {

//dont change the password

**return** ~~NoOpPasswordEncoder~~.~~getInstance~~();

}

//authorization

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.authorizeRequests()

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

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

.antMatchers("/user").hasRole("ADMIN")

.antMatchers("/user1").hasRole("USER")

.antMatchers("/user1").hasRole("ADMIN")

.antMatchers("/").permitAll().and().formLogin().and().httpBasic();

}

}

**SpringSecurityAuthApplicationTests**

**package** com.SpringSecurityAuth;

**import** org.junit.jupiter.api.Test;

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

@SpringBootTest

**class** SpringSecurityAuthApplicationTests {

@Test

**void** contextLoads() {

}

}

**SpringSecurityAuth/ *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>2.7.1</version>

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

</parent>

<groupId>com.SpringSecurityAuth</groupId>

<artifactId>SpringSecurityAuth</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>SpringSecurityAuth</name>

<description>Demo project for Authentication Provider in Spring Security</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>provided</scope>

</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>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>