**Spring Security Assignments**

1. Design and develop a Spring Security Hello World application by using default login form provided by spring security to secure URL access say, for example to access the content of an “admin” page, user need to enter valid credentials. User must also logged out if successfully logged in.

User Java Based and annotation based configuration and In-memory authentication.

HomeResource.java

**package** com.example.springbootsecurity.hi;

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

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

@RestController

**public** **class** HomeResource {

@GetMapping("/")

**public** String home() {

**return** ("<hi>Welcome</h1>");

}

@GetMapping("/user")

**public** String user() {

**return** ("<hi>Welcome User</h1>");

}

@GetMapping("/admin")

**public** String method() {

**return** ("<hi>Welcome Admin</h1>");

}

}

SecurityConfiguration.java

**package** com.example.springbootsecurity.hi;

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

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

**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;

@EnableWebSecurity

**public** **class** SecurityConfiguration **extends** WebSecurityConfigurerAdapter {

@Override

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

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

// Set your configuration on the auth object

auth.inMemoryAuthentication()

.withUser("admin")

.password("admin123")

.roles("ADMIN")

.and()

.withUser("user")

.password("pass")

.roles("USER");

}

@Bean

**public** PasswordEncoder getPasswordEncoder() {

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

}

}

@Override

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

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

http.authorizeRequests()

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

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

.antMatchers("/").permitAll()

.and().formLogin();

}

1. Modify the above application to use custom login form instead of default login form provided by spring security.

User Java Based and annotation based configuration and In-memory authentication.

CommonController.java

package com.example.springsecurityq2.ex;

import org.springframework.stereotype.Controller;

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

@Controller

public class CommonController {

@GetMapping("/page")

public String viewLoginPage() {

return "login";

}

}

SecurityConfig.java

package com.example.springsecurityq2.ex;

import org.springframework.context.annotation.Bean;

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.core.userdetails.User;

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

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

@Configuration

@EnableWebSecurity

public class SecurityConfig extends WebSecurityConfigurerAdapter {

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests().anyRequest()

.authenticated().and()

.formLogin().loginPage("/login")

.usernameParameter("email")

.permitAll()

.and().logout()

.permitAll();

}}

MyConfig.java

package com.example.springsecurityq2.ex;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration

public class MvcConfig implements WebMvcConfigurer{

@Override

public void addViewControllers(ViewControllerRegistry registry) {

registry.addViewController("/login").setViewName("login");

}

}

Login.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0">

<title>Login -TEJASWI</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"

crossorigin="anonymous" />

</head>

<body>

<div class="container-fluid text-center" align="center">

<form th:action="@{/login}" method="post" style="max-width: 300px; margin: 0 auto;">

<div class="border border-secondary p-3 rounded">

<p>

<input type="email" name="email" class="form-control" placeholder="E-mail" required />

</p>

<p>

<input type="password" name="password" class="form-control" placeholder="Password" required />

</p>

<p>

<input type="checkbox" name="remember-me" />&nbsp;Remember Me

</p>

<p>

<input type="submit" value="Login" class="btn btn-primary" />

</p>

</div>

</form>

</div>

</body>

</html>

1. Modify the above application and use database authentication using JDBC instead of in memory authentication.

User Java Based and annotation based configuration and In-memory authentication.

SecureResource.java

**package** com.example.springsecurityques3;

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

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

@RestController

**public** **class** HomeResource {

@GetMapping("/page")

**public** String viewLoginPage() {

**return** "login";

}

@GetMapping("/user")

**public** String viewuserLoginPage() {

**return** "login";

}

@GetMapping("/admin")

**public** String viewadminLoginPage() {

**return** "login";

}

}

mysecure.java

**package** com.example.springsecurityques3;

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

**import** org.springframework.web.servlet.config.annotation.ViewControllerRegistry;

**import** org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration

**public** **class** MyConfig **implements** WebMvcConfigurer{

@Override

**public** **void** addViewControllers(ViewControllerRegistry registry) {

registry.addViewController("/login").setViewName("login");

}

**public** **void** addViewuser(ViewControllerRegistry registry) {

registry.addViewController("/user").setViewName("login");

}

**public** **void** addViewadmin(ViewControllerRegistry registry) {

registry.addViewController("/admin").setViewName("login");

}

}

mysecurityconfig.java

**package** com.example.springsecurityques3;

**import** javax.sql.DataSource;

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

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

**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

**public** **class** SecurityConfiguration **extends** WebSecurityConfigurerAdapter {

@Autowired

DataSource datasource;

@Override

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

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

auth.jdbcAuthentication()

.dataSource(datasource);

}

@Override

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

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

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

.antMatchers("/user").hasAnyRole("ADMIN","USER")

.antMatchers("/").permitAll()

.and().formLogin().loginPage("/login")

.usernameParameter("username").permitAll()

.and().logout().permitAll();

}

@Bean

**public** PasswordEncoder getPasswordEncoder() {

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

}

}

Login.html

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0">

<title>Login - Ritambhara</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"

crossorigin="anonymous" />

</head>

<body>

<div class="container-fluid text-center" align="center">

<form th:action="@{/login}" method="post" style="max-width: 300px; margin: 0 auto;">

</div>

<div class="border border-secondary p-3 rounded">

<p>

<input type="username" name="username" class="form-control" placeholder="UserName" required />

</p>

<p>

<input type="password" name="password" class="form-control" placeholder="Password" required />

</p>

<p>

<input type="checkbox" name="remember-me" />&nbsp;Remember Me

</p>

<p>

<input type="submit" value="Login" class="btn btn-primary" />

</p>

</div>

</form>

</div>

</body>

</html>

Data.sql

INSERT INTO users(username,password,enabled)

values('user','pass',true);

INSERT INTO users(username,password,enabled)

values('admin','admin123',true);

INSERT INTO authorities(username,authority)

values('user','ROLE\_USER');

INSERT INTO authorities(username,authority)

values('admin','ROLE\_ADMIN');

Schema.sql

create table users(

username varchar\_ignorecase(50) not null primary key,

password varchar\_ignorecase(50) not null,

enabled boolean not null

);

create table authorities (

username varchar\_ignorecase(50) not null,

authority varchar\_ignorecase(50) not null,

constraint fk\_authorities\_users foreign key(username) references users(username)

);

create unique index ix\_auth\_username on authorities (username,authority);

1. Modify the above application to limit the number of login attempts
2. Modify the above application to implement “remember me” functionality.
3. Modify the above application to to secure the password by encoding it.(You may use some encryption technique)

ASSIGNMENT 2

1. Create login based application

package com.example.SpringBoot2;

import org.springframework.context.annotation.Bean;

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;

@EnableWebSecurity

public class SecurityConfiguration extends WebSecurityConfigurerAdapter{

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

//set my configuration on the auth object

auth.inMemoryAuthentication()

.withUser("hai")

.password("hai44")

.roles("USER")

.and()

.withUser("teja")

.password("teja44")

.roles("ADMIN");

}

@Bean

public PasswordEncoder getPasswordEncoder() {

return NoOpPasswordEncoder.getInstance();

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

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

.antMatchers("/user").hasAnyRole("USER","ADMIN")

.antMatchers("/").permitAll()

.and().formLogin();

}

}

package com.example.SpringBoot2;

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

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

@RestController

public class HomeResource {

@GetMapping("/")

public String home() {

return ("<h1>Welcome to Spring Security</h1>");

}

@GetMapping("/user")

public String user() {

return ("<h1>Welcome User</h1>");

}

@GetMapping("/admin")

public String admin() {

return ("<h1>Welcome Admin</h1>");

}

}