



Spring Security

Lab80: Implementing Spring Security

Lab80:Working Steps:

1. Create the Dynamic Web Project.
2. Copy 21 Spring Jars to WEB-INF/lib folder.
3. **Copy 3 Spring Security jars to WEB-INF/lib folder.**
spring-security-config-5.1.3.RELEASE.jar
spring-security-core-5.1.3.RELEASE.jar
spring-security-web-5.1.3.RELEASE.jar
4. Write JLCWebInitializer class to initialize the DS

```
public class JLCWebInitializer extends
AbstractAnnotationConfigDispatcherServletInitializer {
....
}
```

5. **Write JLCSecurityInitializer class to initialize the Security Filter**

```
public class JLCSecurityInitializer extends
AbstractSecurityWebApplicationInitializer {
....
}
```

6. Write JLCWebConfig to Configure the Beans required
@EnableWebMvc
@Configuration
@ComponentScan({ "com.coursecube.spring" })
@Import({JLCSecurityConfig.class })
public class JLCWebConfig {

}

7. **Write JLCSecurityConfig as follows.**

```
@Configuration
@EnableWebSecurity
public class JLCSecurityConfig extends WebSecurityConfigurerAdapter {
.....
}
```

8. Write index.jsp
9. Write the UserController with /
10. Deploy and Run
http://localhost:54321/Lab80/ => index.jsp



Q)Who is handling Security?

Ans: Security Filter

Task 1: Decide the Roles of Application

Roles : Different Types of Users
Admin , StoreKeeper , Customer

Task 2: Decide Which Resources has to be protected.

Public : Any one can access without Authentication

@GetMapping("/") - All

@GetMapping("/viewBooks") - All

Protected: Any one can access after Authentication only

@GetMapping("/addBook") - Admin , StoreKeeper

@GetMapping("/editBook") - Admin , StoreKeeper

@GetMapping("/deleteBook") - Admin

@GetMapping("/placeOrder") - Customer

Points to Remember when you write Security Config Class

- A) Mark the Configuration class with @EnableWebSecurity
- B) Configuration class has to extend WebSecurityConfigurerAdapter
- C) Configure the AuthenticationManager.

Types of AuthenticationManager:

**inMemoryAuthentication (Pick this)

**jdbcAuthentication

**ldapAuthentication

- D) Configure HttpSecurity with the following details

***Resource Authorization (Done)

***Error Handling Configuration (Done)

***Login Configuration

***Logout Configuration

***Session Management



Spring Security

Lab80: Files required

1. index.jsp	2. mymenu.jsp
3. viewBooks.jsp	4. addBook.jsp
5. editBook.jsp	6. deleteBook.jsp
7. placeOrder.jsp	8. invalidAccess.jsp
9. UserController.java	10. JLCWebConfig.java
11. JLCSecurityConfig.java	12. JLCWebInitializer.java
13. SpringSecurityInitializer.java	

1. index.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<div align="center">
<br/><h2>
<a href='<c:url value="viewBooks"/>'> View Books</a>
<br/><br/>
<a href='<c:url value="addBook"/>'> Add Book</a>
<br/><br/>
<a href='<c:url value="editBook"/>'> Edit Book</a>
<br/><br/>
<a href='<c:url value="deleteBook"/>'> Delete Book</a>
<br/><br/>
<a href='<c:url value="placeOrder"/>'> Place Order</a>
</h2>
</div>
</body> </html>
```

2. mymenu.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<h2>
<a href='<c:url value="/" />'>Home</a>
| Welcome : ${pageContext.request.userPrincipal.name}
| <a href='<c:url value="logout"/>'>Logout</a>
</h2>
```



3. viewBooks.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<div align="right">
<c:import url="mymenu.jsp"/>
</div>
<div align="center">

<hr>
<h2> Display List of books here</h2>
<h2> Display List of books here</h2>
<h2> Display List of books here</h2>
<h2> Display List of books here</h2>
<h2> Display List of books here</h2>
</div>
</body> </html>
```

4.addBook.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>

<div align="right">
<c:import url="mymenu.jsp"/>
</div>

<div align="center">
<hr>
<h2> Add Book Form Here</h2>
<h2> Add Book Form Here</h2>
<h2> Add Book Form Here</h2>
<h2> Add Book Form Here</h2>
<h2> Add Book Form Here</h2>
</div>
</body> </html>
```



5.editBook.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<body>
<div>
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<div align="right">
<c:import url="mymenu.jsp"/>
</div>
<div align="center">
<hr>
<h2> Edit Book Form Here</h2>
<h2> Edit Book Form Here</h2>
<h2> Edit Book Form Here</h2>
<h2> Edit Book Form Here</h2>
<h2> Edit Book Form Here</h2>
</div>
</body>
</html>
```

6.deleteBook.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<div align="right">
<c:import url="mymenu.jsp"/>
</div>
<div align="center">

<hr>
<h2> Delete Book Form Here</h2>
<h2> Delete Book Form Here</h2>
<h2> Delete Book Form Here</h2>
<h2> Delete Book Form Here</h2>
<h2> Delete Book Form Here</h2>
</div>
</body>
</html>
```



7. placeOrder.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>

<div align="right">
<c:import url="mymenu.jsp"/>
</div>
<div align="center">

<hr>
<h2> Place Your Order Here</h2>
<h2> Place Your Order Here</h2>
<h2> Place Your Order Here</h2>
<h2> Place Your Order Here</h2>
<h2> Place Your Order Here</h2>
<h2> Place Your Order Here</h2>
</div>
</body>
</html>
```

8. invalidAccess.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<div align="right">
<c:import url="mymenu.jsp"/>
</div>
<div align="center">
<hr>
<h2 align="center">You are not authorized to access this page</h2>
</div>
</body>
</html>
```



9.UserController.java

```
package com.coursecube.spring;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 * @see      : www.jlcindia.com
 */
@Controller
public class UserController {

    @GetMapping( "/" )
    public String showIndexPage() {
        System.out.println("*** showIndexPage ***");
        return "index";
    }

    @GetMapping( "/viewBooks" )
    public String showViewBooksPage() {
        System.out.println("*** showViewBooksPage ***");
        return "viewBooks";
    }

    @GetMapping( "/addBook" )
    public String showAddBookPage() {
        System.out.println("*** showAddBookPage ***");
        return "addBook";
    }

    @GetMapping( "/editBook" )
    public String showEditBookPage() {
        System.out.println("*** showEditBookPage ***");
        return "editBook";
    }

    @GetMapping( "/deleteBook" )
    public String showDeleteBookPage() {
        System.out.println("*** showDeleteBookPage ***");
        return "deleteBook";
    }

    @GetMapping( "/placeOrder" )
    public String showPlaceOrderPage() {
        System.out.println("*** showPlaceOrderPage ***");
        return "placeOrder";
    }
}
```



10.JLCWebConfig.java

```
package com.coursecube.spring;

import org.springframework.context.annotation.*;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import org.springframework.web.servlet.view.InternalResourceViewResolver;
import org.springframework.web.servlet.view.JstlView;

@EnableWebMvc
@Configuration
@ComponentScan({ "com.coursecube.spring" })
@Import({JLCSecurityConfig.class })
public class JLCWebConfig {
    @Bean
    public InternalResourceViewResolver viewResolver() {
        InternalResourceViewResolver viewResolver = new InternalResourceViewResolver();
        viewResolver.setViewClass(JstlView.class);
        viewResolver.setPrefix("/WEB-INF/myjsps/");
        viewResolver.setSuffix(".jsp");
        return viewResolver;
    }
}
```

11.JLCSecurityConfig.java

```
package com.coursecube.spring;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag
erBuilder;
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.WebSecurityConfigurerAd
apter;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */

@Configuration
@EnableWebSecurity
```




```
public class JLCSecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
        auth.inMemoryAuthentication().withUser("srinivas").password("{noop}srinivas").roles("ADMIN");
        auth.inMemoryAuthentication().withUser("vas").password("{noop}vas").roles("STOREKEEPER");
        auth.inMemoryAuthentication().withUser("sd").password("{noop}sd").roles("CUSTOMER");
        auth.inMemoryAuthentication().withUser("ds").password("{noop}ds").roles("CUSTOMER");
        auth.inMemoryAuthentication().withUser("sri").password("{noop}sri").roles("CUSTOMER", "ADMIN");
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
            .antMatchers("/deleteBook**").access("hasRole('ROLE_ADMIN')")
            .antMatchers("/addBook**").access("hasAnyRole('ROLE_ADMIN','ROLE_STOREKEEPER')")
            .antMatchers("/editBook**").access("hasAnyRole('ROLE_ADMIN','ROLE_STOREKEEPER')")
            .antMatchers("/placeOrder**").access("hasAnyRole('ROLE_CUSTOMER')")
            .and().formLogin()
            .and().exceptionHandling().accessDeniedPage("/WEB-INF/myjsps/invalidAccess.jsp");
    }
}
```

13. JLCSecurityInitializer.java

```
package com.coursecube.spring;

import org.springframework.security.web.context.AbstractSecurityWebApplicationInitializer;

public class JLCSecurityInitializer extends AbstractSecurityWebApplicationInitializer {

}
```



Lab81: Spring Security:

Improvements required in Lab80:

- A) Display either login or logout
- B) Display the Role after the username.
Home Welcome : Login
Home Welcome : sd [ROLE_CUSTOMER] Logout
- C) Customize Login Page and Error Messages.
- D) Customize Logout Page
- E) Display the Links according to Role.

Lab81: Working Steps:

- 1) Copy Lab80 and paste as Lab81.

A) Display either login or logout

Steps for Task A:

- 2) Do the following to use the Security Tag Libs.
 - a) copy the following jar to lib folder.
spring-security-taglibs-5.1.3.RELEASE.jar
 - b) use the following taglib directive in required.jsp.
<%@ taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>
- 3) Update mymenu.jsp

```
<sec:authorize access="! isAuthenticated()">
| <a href='<c:url value="login"/>'>Login</a>
</sec:authorize>

<sec:authorize access="isAuthenticated()">
<a href="<c:url value='logout'/>"> Logout</a>
</sec:authorize>
```

B) Display the Role after the username.

Steps for Task A:

- 4) Write the method to get the roles of the current user.

```
List<String> getUserRoles() {
..
}
```



- 5) Get the Roles list from `getUserRoles()` method and store that as attribute. (do it in all the methods of controller).

```
List<String> myroles=getUserRoles();  
model.addAttribute("MyRoles", myroles);
```

- 6) Display the attribute in `mymenu.jsp`

```
${MyRoles}
```

C) Customize Login Page and Error Messages.

Steps for Task C:

- 7) Design the `login.jsp`
- 8) Update `HttpSecurity` with login configuration.

```
.formLogin()  
.loginPage("/login")  
.failureUrl("/login?error")  
.usernameParameter("myusername")  
.passwordParameter("mypassword")
```

- 9) Add the method in the controller to handle request for `/login`

```
@GetMapping("/login")  
public String loginPage(...) {  
    .....  
    //Get the Cart Data if any and store in Session  
    //Get the Last Accessed Time from DB.  
  
    return "login";  
}
```

D) Customize Logout Page

Steps for Task D:

- 10) Update `HttpSecurity` with logout configuration.

```
.logout()  
.invalidateHttpSession(true)  
.logoutSuccessUrl("/login?logout")
```

- 11) Add the method in the controller to handle request for `/logout`

```
@GetMapping("/logout")  
public String logoutPage(.....) {  
    //Store the Cart Data if any into the database  
    //Store the Logout time in DB.  
}
```



E) Display the Links according to Role.

F) Steps for Task E:

```
<sec:authorize access="hasRole('ADMIN') or hasRole('STOREKEEPER')">
<br/><br/>
<a href='<c:url value="addBook"/>'> Add Book</a>
<br/><br/>
<a href='<c:url value="editBook"/>'> Edit Book</a>
</sec:authorize>
```

Spring Security

Lab81: Files required

1. index.jsp	Updated in Lab81
2. mymenu.jsp	Updated in Lab81
3. viewBooks.jsp	Same as Lab80
4. addBook.jsp	Same as Lab80
5. editBook.jsp	Same as Lab80
6. deleteBook.jsp	Same as Lab80
7. placeOrder.jsp	Same as Lab80
8. invalidAccess.jsp	Same as Lab80
9. login.jsp	Newly added in Lab81
10. UserController.java	Updated in Lab81
11. JLCWebConfig.java	Same as Lab80
12. JLCSecurityConfig.java	Updated in Lab81
13. JLCWebInitializer.java	Same as Lab80
14. SpringSecurityInitializer.java	Same as Lab80

1. index.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>

<html>
<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<div align="right">
<c:import url="mymenu.jsp"/>
</div>
<div align="center">
```



```
<br/>
<h2>
<a href='<c:url value="viewBooks"/>'> View Books</a>

<sec:authorize access="hasRole('ADMIN') or hasRole('STOREKEEPER')">
<br/><br/>
<a href='<c:url value="addBook"/>'> Add Book</a>
<br/><br/>
<a href='<c:url value="editBook"/>'> Edit Book</a>
</sec:authorize>

<sec:authorize access="hasRole('ADMIN')">
<br/><br/>
<a href='<c:url value="deleteBook"/>'> Delete Book</a>
</sec:authorize>

<sec:authorize access="hasRole('CUSTOMER')">
<br/><br/>
<a href='<c:url value="placeOrder"/>'> Place Order</a>
</sec:authorize>

</h2>
</div>
</body>
</html>
```

2.mymenu.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>

<h2>
<a href='<c:url value="/" />'>Home</a>
| Welcome : ${pageContext.request.userPrincipal.name} ${MyRoles}
| <sec:authorize access="!isAuthenticated()">
<a href='<c:url value="login"/>'> Login </a>
| <a href='<c:url value="showRegister"/>'> Register </a>
</sec:authorize>
<sec:authorize access="isAuthenticated()">
<a href="<c:url value='logout'/>"> Logout</a>
</sec:authorize>
</h2>
```



9.login.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<head>
<style type="text/css">
input,table{
font-size:25px;
margin:10px;
}
</style>
</head>

<body>
<div align="left">
<h1 align="center">Welcome to JLC BookStore</h1>
</div>
<hr/>
<div align="center">
<h2 align="center">
<c:if test="${param.error != null}">
<font color="red" size=5> Invalid username and password. </font>
</c:if>

<c:if test="${param.logout != null}">
<font color="red" size=5>You have logged out successfully.</font>
</c:if>
</h2>
</div>

<div align="center">
<form name='loginForm' action="<c:url value='/login' />" method='POST'>
<table>
<tr>
<td align="center" colspan="2">
<h2>User Login Form</h2>
</td>
</tr>
<tr>
<td>Username</td>
<td><input type='text' name='username' value=""></td>
</tr>

<tr>
<td>Password</td>
```



```
<td><input type='password' name='password' /></td>
</tr>
<tr>
<td align="center" colspan="2">
<input type="hidden"      name="{_csrf.parameterName}" value="{_csrf.token}" />
<input type="submit" value="LOGIN">
</td>
</tr>
</table>
</form>
</div>
<div align="center">
<h2>  <b> <a href='<c:url value="showRegister"/>'> New User Register Here </a>
</b></h2>
</div>

</body>
</html>
```

10.UserController.java

```
package com.coursecube.spring;

import java.util.*;
import javax.servlet.http.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.security.web.authentication.logout.*;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.SessionAttributes;
/*
 * @Author : Srinivas Dande
 * @company : Java Learning Center
 */

@Controller
@SessionAttributes("MyRoles")
```



```
public class UserController {

    @Autowired
    UserService userService;

    @GetMapping("/")
    public String showIndexPage(Model model) {
        System.out.println("** showIndexPage **");

        List<String> myroles = getUserRoles();
        System.out.println(myroles);
        model.addAttribute("MyRoles", myroles);
        return "index";
    }

    @GetMapping("/viewBooks")
    public String showViewBooksPage(Model model) {
        System.out.println("** showViewBooksPage **");

        List<String> myroles = getUserRoles();
        System.out.println(myroles);
        model.addAttribute("MyRoles", myroles);

        return "viewBooks";
    }

    @GetMapping("/addBook")
    public String showAddBookPage(Model model) {
        System.out.println("** showAddBookPage **");

        List<String> myroles = getUserRoles();
        System.out.println(myroles);
        model.addAttribute("MyRoles", myroles);

        return "addBook";
    }

    @GetMapping("/editBook")
    public String showEditBookPage(Model model) {
        System.out.println("** showEditBookPage **");

        List<String> myroles = getUserRoles();
        System.out.println(myroles);
        model.addAttribute("MyRoles", myroles);
    }
}
```




```
return "editBook";
}
@GetMapping("/deleteBook")
public String showDeleteBookPage(Model model) {
    System.out.println("** showDeleteBookPage **");

    List<String> myroles = getUserRoles();
    System.out.println(myroles);
    model.addAttribute("MyRoles", myroles);

    return "deleteBook";
}

@GetMapping("/placeOrder")
public String showPlaceOrderPage(Model model) {
    System.out.println("** showPlaceOrderPage **");

    List<String> myroles = getUserRoles();
    System.out.println(myroles);
    model.addAttribute("MyRoles", myroles);

    return "placeOrder";
}

@GetMapping("/showRegister")
public String showRegisterPage(Model model) {
    System.out.println("** showRegisterPage **");
    User myuser=new User();
    model.addAttribute("myuser",myuser);
    return "register";
}
@PostMapping("/registerUser")
public String registerUser(@ModelAttribute("myuser") User user, BindingResult
bindingResult,Model model) {
    System.out.println("** registerUser **");

    userService.registerUser(user);
    return "index";
}

// Spring Security see this :
@GetMapping("/login")
public String loginPage(
@RequestParam(value = "error", required = false) String error,
```



```
@RequestParam(value = "logout", required = false) String logout, Model model) {
    System.out.println("*** loginPage ***");

    List<String> myroles = getUserRoles();
    System.out.println(myroles);
    model.addAttribute("MyRoles", myroles);

    if (error != null) {
        model.addAttribute("error", "Invalid username and password!");
    }

    if (logout != null) {
        model.addAttribute("msg", "You've been logged out successfully.");
    }
    getUserRoles();
    return "login";
}

@GetMapping("/logout")
public String logoutPage(HttpServletRequest request, HttpServletResponse response) {
    System.out.println("*** logoutPage ***");

    List<String> myroles = getUserRoles();
    System.out.println(myroles);

    Authentication auth = SecurityContextHolder.getContext().getAuthentication();
    if (auth != null) {
        new SecurityContextLogoutHandler().logout(request, response, auth);
    }
    return "redirect:/login?logout";
}

private List<String> getUserRoles() {
    Authentication auth = SecurityContextHolder.getContext().getAuthentication();
    List<String> myroles = new ArrayList<String>();
    if (auth != null) {
        Collection<GrantedAuthority> col = (Collection<GrantedAuthority>) auth.getAuthorities();
        for (GrantedAuthority gauth : col)
            myroles.add(gauth.getAuthority());
    }
    return myroles;
}
}
```



12.JLCSecurityConfig.java

```
package com.coursecube.spring;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManag
erBuilder;
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.WebSecurityConfigurerAd
apter;

@Configuration
@EnableWebSecurity
public class JLCSecurityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
        auth.inMemoryAuthentication().withUser("srinivas").password("{noop}srinivas").roles("AD
MIN");
        auth.inMemoryAuthentication().withUser("vas").password("{noop}vas").roles("STOREKEEPE
R");
        auth.inMemoryAuthentication().withUser("sd").password("{noop}sd").roles("CUSTOMER");
        auth.inMemoryAuthentication().withUser("ds").password("{noop}ds").roles("CUSTOMER");
        auth.inMemoryAuthentication().withUser("sri").password("{noop}sri").roles("CUSTOMER","
ADMIN");
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
        .antMatchers("/deleteBook**").access("hasRole('ROLE_ADMIN')")
        .antMatchers("/addBook**").access("hasAnyRole('ROLE_ADMIN','ROLE_STOREKEEPER')")
        .antMatchers("/editBook**").access("hasAnyRole('ROLE_ADMIN','ROLE_STOREKEEPER')")
        .antMatchers("/placeOrder**").access("hasAnyRole('ROLE_CUSTOMER')")
        .and()
        .formLogin()
        .loginPage("/login")
        .failureUrl("/login?error")
        .usernameParameter("myusername")
        .passwordParameter("mypassword")
    }
}
```



```
.and()  
.logout()  
.invalidateHttpSession(true)  
.logoutSuccessUrl("/login?logout")  
.and()  
.exceptionHandling()  
.accessDeniedPage("/WEB-INF/myjsps/invalidAccess.jsp");  
}  
}
```