

***In a previous post we had implemented Spring Boot Security for a Form Application. It made use of the default Spring Login Page.***

***In this Use case we will be adding our own custom login web page. On log out we will be directed to this login page with some logout message.***

1. Continue with ***“boot-security-loginwith-users”***

The Custom login page login.jsp is as follows-

```

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

<c:set var="contextPath" value="" />

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- The above 3 meta tags *must* come first in the head; any other head content must
come *after* these tags -->
<meta name="description" content="">
<meta name="author" content="">

<title>Log in with your credentials</title>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>

<body>

    <div class="container">

        <form method="POST" action="/login" class="form-signin">
            <h2 class="form-heading">Log in</h2>

            <div class="form-group ">
                <span></span> <input name="username" type="text"
                    class="form-control" placeholder="Username"
autofocus="true" /> <input
                    name="password" type="password" class="form-control"
                    placeholder="Password" /> <span></span>

                <button class="btn btn-lg btn-primary btn-block"
type="submit">Log
                    In</button>

            </div>

        </form>

    </div>
<!-- /container -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/
jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/
bootstrap.min.js"></script>
</body>
</html>

```

2. Next in the **EmployeeController**, add a GET method which returns the custom login page with RequestMapping of /login.

```
@RequestMapping(value = "/login", method = RequestMethod.GET)
public String login(Model model, String error, String
logout) {
    if (error != null)
        model.addAttribute("errorMsg", "Your username and
password are invalid.");

    if (logout != null)
        model.addAttribute("msg", "You have been logged out
successfully.");

    return "login";
}
```

3. Finally modify the Spring Security configuration where we specify the custom Login Page to be called during login.

### **EmployeeSecurityConfiguration.java**

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

Rewrite the configure() method as shown:

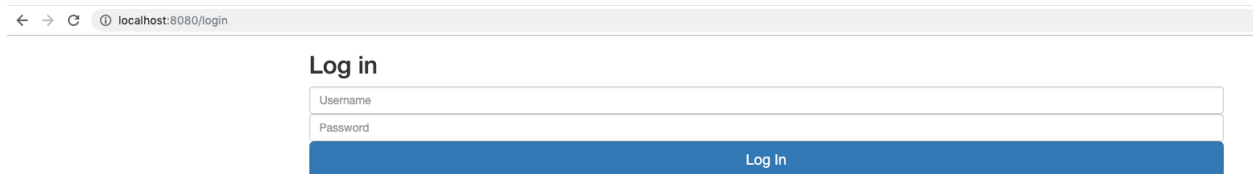
```
@Override
protected void configure(HttpSecurity http) throws Exception
{
    http.authorizeRequests().antMatchers("/").permitAll().antMatchers(
"/welcome")
        .hasAnyRole("USER", "ADMIN").antMatchers("/
getEmployees").hasAnyRole("USER", "ADMIN")
        .antMatchers("/
addNewEmployee").hasAnyRole("ADMIN").anyRequest().authenticated(
)

        .and().formLogin().loginPage("/login")
        .permitAll().and().logout().permitAll();
}
```

```
    http.csrf().disable();  
}
```

#### 4. RUN the APP

Must be able to see custom login page

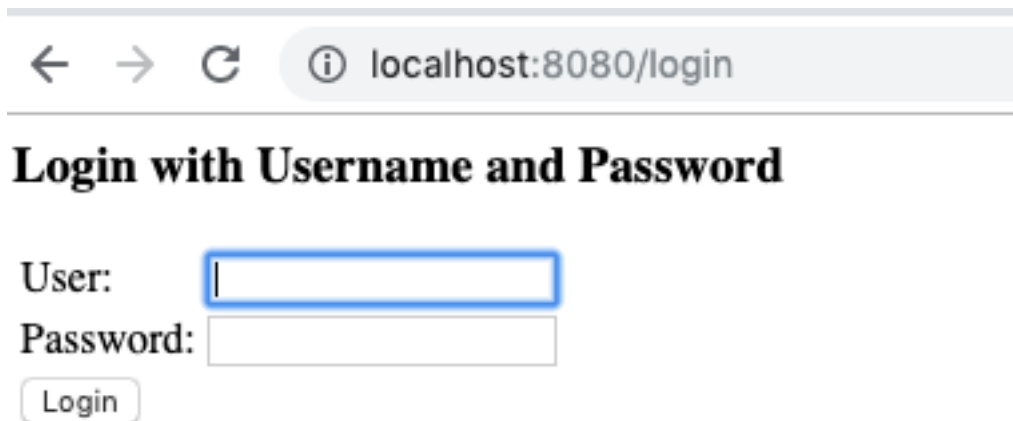


The screenshot shows a web browser window with the address bar displaying "localhost:8080/login". The page content includes a "Log in" heading, a "Username" input field, a "Password" input field, and a blue "Log In" button.

Log in	
Username	
Password	
Log In	

#### 8. RUN the App

9. Visit ***http://localhost:8080/welcome***



← → ↻ ⓘ localhost:8080/login

## Login with Username and Password

User:

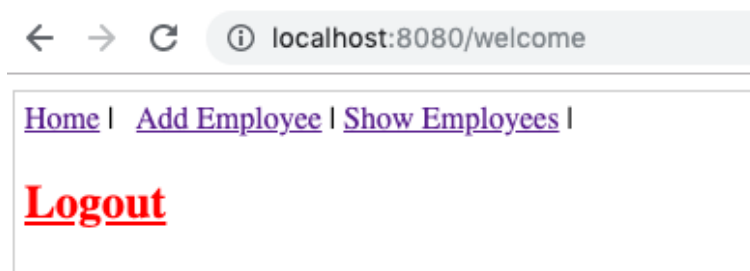
Password:

Login

10.

Observe **EmployeeSecurityConfiguration.java** for the security in-memory credentials

11. If you login with a user of ROLE USER, cannot access addEmployee API



← → ↻ ⓘ localhost:8080/welcome

[Home](#) | [Add Employee](#) | [Show Employees](#) |

**Logout**

**Hello!**

12. Click on **Add Employee** link

# Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun Oct 27 23:33:04 EDT 2019

There was an unexpected error (type=Forbidden, status=403).

Access is denied

13. addEmployee is available only for ADMIN ROLE user

13.1 Click on **Logout** link

13.2 Change the URL to ***http://localhost:8080/welcome***

13.3 Login as admin/admin

13.4 Click on Add Employee

[Home](#) | [Add Employee](#) | [Show Employees](#) |

**[Logout](#)**

## Add New Employee

Enter Employee Id

Enter Name

14. Enter form data and click on Submit

←

→

↻

localhost:8080/addNewEmployee

Home | [Add Employee](#) | [Show Employees](#) |

**Logout**

### Show All Employees

- Employee [empId=110, empName=John]
- Employee [empId=190, empName=Tom]
- Employee [empId=160, empName=jane]

