



CS544 EA

# Applications

Spring Security: Login / Logout

# Customize Login

- You can create a custom login page
  - Instead of using the generated one
  - Spring Sec lets you configure every detail

```
http.authorizeHttpRequests(auth -> auth
    .formLogin(form -> form.loginPage("/login")
        .loginProcessingUrl("/login")
        .usernameParameter("username")
        .passwordParameter("password")
        .failureUrl("/index")
        .defaultSuccessUrl("/contact")
        .permitAll()))
```

You can also provide a login-handler method (not shown)

`.permitAll()` makes sure users have access to all login related URLs

# XML Config

```
<sec:http use-expressions="false">
  <sec:intercept-url pattern="/login" access="permitAll" />
  <sec:intercept-url pattern="/index" access="permitAll" />
  <sec:intercept-url pattern="/addContact" method="GET" access="hasRole('ADMIN')" />
  <sec:form-login login-page="/login" login-processing-url="/login"
    username-parameter="username" password-parameter="password"
    authentication-failure-url="/index"
    default-target-url="/contacts" />
  <sec:logout />
</sec:http>
```

No permitAll option on the tag

# Custom Login Form

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML>

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Login Page!</h1>
    <c:if test="${error eq true}">
      <p>${sessionScope["SPRING_SECURITY_LAST_EXCEPTION"].message}</p>
    </c:if>
    <form method="post" action="<c:url value='/login' />">
      User: <input name="username"
value= '<c:if test="${not empty param.login_error}"><c:out value="${SPRING_SECURITY_LAST_USERNAME}" /></c:if>' />
      <br />
      Pass: <input type="password" name='password' /> <br />
      <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
      <input type="submit" />
    </form>
  </body>
</html>
```

# Customize Logout

- You can similarly configure the logout

```
http.authorizeHttpRequests(auth -> auth
    .logout(logout -> logout
        .logoutUrl("/logout")
        .logoutSuccessUrl("/login?logout")
        .invalidateHttpSession(true)
        .deleteCookies("cookie-names")
        .permitAll()
    )
)
```

Options also exist to provide a  
logoutHandler and a  
logoutSuccessHandler

```
<sec:logout logout-url="/logout" logout-success-url="/index"
    delete-cookies="cookie-names" invalidate-session="true" />
```

