CS544 EA

Applications

Spring Security: Taglib

Taglib

```
<%@ taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>
```

Tag	Use
<sec:authorize access=”expression”>	Show content if expression is true
<sec:authorize url=”/location”>	Show content if allowed to access location
<sec:authentication>	Access properties of logged in person
<sec:csrfInput />	Add csrf tags into a form
<sec:csrfMetaTags />	Add csrf meta tags (to access with JS)

CSRF

- Cross Site Request Forgery
  - Tricks user into making a request
  - Accidentally submitting a form to a site they’re (hopefully) already logged into

- Mitigated by having a token on the real form
  - If random token value is present, request is real

Spring Security CSRF

- Spring Security includes CSRF protection
  - Always on by default
  - Spring Tags <form:form> automatically includes token

- Any <form> submitted without csrf_token not accepted
  - Common mistake made by beginners
  - They forget to include token on their <form>
    - Spring Security give 403 Forbidden on submit

Examples

```
<sec:authorize access="hasRole('ADMIN')" >
    <a href="addContact"> Add a Contact</a>
</sec:authorize>
```

```
<sec:authorize url="/addContact" >
    <a href="addContact"> Add a Contact</a>
</sec:authorize>
```

```
<sec:authorize access="!isAuthenticated()">
  <p><a href="login">Login</a></p>
</sec:authorize>
<sec:authorize access="isAuthenticated()">
  <p>Welcome Back, <sec:authentication property="name"/></p>
  <p><a href="logout">Logout</a></p>
</sec:authorize>
```

CSRF

```
<%@taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>
<%@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>
        <sec:csrfMetaTags />
        <script type="text/javascript" language="javascript">
          var csrfParameter = $("meta[name='_csrf_parameter']").attr("content");
          var csrfHeader = $("meta[name='_csrf_header']").attr("content");
          var csrfToken = $("meta[name='_csrf']").attr("content");
        </script>
    </head>
    <body>
        <h1>Login Page!</h1>
        <c:if test="${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}"/>
            <sec:csrfInput />
            <input type="submit" />
        </form>
    </body>
</html>
```

If CSRF needed in JS

Clean replacement