COMP S380F Web Applications: Design and Development

Lab 7: EL, JSTL – Multi-value request parameter, Session activity tracking & Shopping cart

We will revisit some previous lab exercises and use EL and JSTL. The following topics are covered:

- Accessing EL implicit objects and scoped variables
- Using the JSTL core tag library (with the namespace "c")
- Using the JSTL formatting tag library (with the namespace "fmt")

Task 1: Setting up for EL and JSTL

Download the branch "**lab07**" of the GitHub repository "**https://github.com/cskeith/380_2024.git**".

1. Add the following dependency for EL. The JSTL dependencies were already included. Reload the project in the Gradle window (on the right-side menu).

```
implementation 'jakarta.el:jakarta.el-api:5.0.1'
```

2. In web.xml, we have included the JSP prelude **base.jspf**, which has the JSP taglib directive for the JSTL core tag library (c). Add the following the JSTL formatting tag library for fmt:

```
<%@ taglib prefix="fmt" uri="jakarta.tags.fmt" %>
```

Task 2: Revisiting the exercise on multi-value request parameter

- Servlet: MultiValueParameterServlet (URL pattern: /checkboxes)
- JSP pages:
  - /WEB-INF/jsp/MultiValueForm.jsp
  - /WEB-INF/jsp/MultiValueResult.jsp

Your task is to update the JSP page **MultiValueResult.jsp** to remove all JSP scripting elements.

1. Use the EL ${paramValues.fruit} to get the string array of the request parameter "fruit".
2. You need to use <c:choose>, <c:when>, <c:otherwise>, and <c:forEach>.

Task 3: Revisiting the exercise on tracking session activity

- JavaBean: PageVisit
- Servlet: ActivityServlet (URL pattern: /do/*)
- JSP page: /WEB-INF/jsp/view/viewSessionActivity.jsp

Your task is to update the JSP page **viewSessionActivity.jsp** to remove all JSP scripting elements.

1. Add a getter `getTimeString()` for a virtual property `timeString` of the JavaBean **PageVisit**:

```
public String getTimeString() {
    if (this.leftTimestamp == null) {
        return "";
    }
    long timeInterval = this.leftTimestamp - this.enteredTimestamp;
    if (timeInterval < 1_000) {
        return "less than one second";
    }
    if (timeInterval < 60_000) {
        return (timeInterval / 1_000) + " seconds";
    }
    return "about " + (timeInterval / 60_000) + " minutes";
}
```

2. We first ignore the date formatting and replaces all the scripting elements with EL and JSTL <c:if> and <c:forEach>. Note that
    - **${pageContext.session.id}** gives you the session ID from the session object.
    - Use **${pageContext.session["new"]}** instead of ${pageContext.session.new}. The latter does not work as "new" is a Java keyword.
    - We can get the time string for a PageVisit object "visit" using EL: **${visit.timeString}**

3. After successfully running the web app, we will add date formatting using **<fmt:formatDate>**, which has the following attributes:
    - **value** must be an instance of `java.util.Date`
    - **type** = {date, time, both}: Display only date or only time, or both date and time.
    - **dateStyle / timeStyle** = {default, medium, long, full}: How much detail to show.
    - We can use **pattern** to define the format (replacing type, dateStyle & timeStyle).

    Use the following code to display the creation time of the session (and display the enteredTimestamp of a PageVisit object, similarly):

```
<jsp:useBean id="timeValues" class="java.util.Date"/>
<c:set target="${timeValues}" property="time"
       value="${pageContext.session.creationTime}"/>
Session created: <fmt:formatDate value="${timeValues}"
                                 pattern="EEE, d MMM yyyy HH:mm:ss Z"/><br/>
```

    - As the `value` of `fmt:formatDate` only accepts a `Date` object, we need to use `jsp:useBean` to define a `Date` object which we call `timeValues`.
    - The `jsp:useBean` statement is only required one time before all `fmt:formatDate` statements.
    - Use `c:set` to set the `time` property of the `Date` object `timeValues`.

4. If your browser's locale is not in English, the displayed time would not be in English. You can use the following `fmt:setLocale` before using `fmt:formatDate` to set the locale to English:

```
<fmt:setLocale value="en" scope="session"/>
```

Task 4: Revisiting the exercise on shopping cart

- Servlet: StoreServlet (URL pattern: /shop)
- JSP page:
    - /WEB-INF/jsp/view/browse.jsp
    - /WEB-INF/jsp/view/viewCart.jsp

Your task is to update **browse.jsp** and **viewCart.jsp** to remove all JSP scripting elements.

1. You need to use <c:choose>, <c:when>, <c:otherwise>, and <c:forEach>.
2. If you use `<c:forEach var="product" items="${products}">` to loop through each product in the map "products", the key and value of the variable "product" can be accessed using **product.key** and **product.value**.