

Homework 3: Simple Servlets

For this homework, you will create a simple servlet that works with the forms you created last homework to print out the form data entered by the user.

You will do this homework as a team; however, each member of your team will be responsible for the completion of a particular task.

Step 1. Update your GameDen form

Last homework, you created a web page for the **GameDen** project. This homework, you will continue to work with your **GameDen** web page.

When you submit an HTML form, the form's target receives a set of name/value pairs that correspond to the user's input. Thus, you must make sure that your form properly specifies all those name/value pairs. To do so, you must set **name** and **value** attributes for your form elements.

Specifically, you must make sure that each **input** element (except submit buttons) and **option** element has a **name** attribute and the name defined is unique (with respect to the form). Your servlet will use these *names* to look up what *values* the user entered. Note that radio-button **input** elements should share the same name. For example:

```
<input type="radio" name="sex" value="male">Male<br>
<input type="radio" name="sex" value="female">Female
```

You must also specify **value** attributes for each non-text input element. (Text-base elements send the user-entered text as their value.)

Also, you must fill in the **target** attribute of your **form** element as follows. If your HTML file is named **taskX.html**, then enter the target as **taskX.do**. Finally, make sure that your **form** element's **method** attribute is set to "post".

Step 2. Create a servlet

You must create a servlet that corresponds to your HTML form. If your HTML file is named **taskX.html**, then name your servlet class **TaskXServlet**, and set its URL pattern to **/taskX.do**. Recall that URL patterns are set with the Java tag **WebServlet**. For example:

```
@WebServlet("/task7.do")
public class Task7Servlet ...
```

Note that your form's **target** should match your URL pattern, except the **target** should not have the leading slash.

Your servlet must respond to HTTP POST requests with a proper HTML5 web page that prints out each of the *names* that your form submitted along with the user-entered *values* that correspond to that name. Hardcode the names that you're interested in into your servlet. (Do *not* make a general-purpose servlet that prints all name/value pairs associated with the request no

matter what the names are.) Although I'm leaving the style of this page up to you, make it neat and easy to read. Also, provide a link at the end of the servlet-generated web page that takes you back to the form.

Step 3. Submit (by tagging) your team's submission

The following instructions are essentially the same as last time; only the tag name has changed.

Attention! Before performing this step, you must make sure that all team members have committed their edits to the **trunk** in the repository.

Only one team member (the leader) performs the following.

First, you must fill out the **README.txt** file in your project's **trunk**. The file should list which team member performed each task (one team member per task).

To submit work in this course, you must tag it. Then, I will checkout the revision that you tagged and grade it. By tagging, you tell me that you are done, and this is the version you want me to grade.

The tag you must use for this homework is **hw3** (case sensitive, no spaces).

To tag the current revision of your trunk as **hw3**, do as follows:

1. Go to the **SVN Repository Exploring** perspective in Eclipse.
2. In the **SVN Repositories** view, find the **trunk** folder that you want to tag.
3. Right-click on the **trunk** folder, and click **Show History**. This should open the **History** view with a table listing the past commits to the **trunk**.
4. In the History table, right-click the newest revision (i.e., the one with the greatest revision number), and click **Tag from...** This should open a **Create Tag** dialog.
5. Enter **hw3** into the **Tag** field and optionally enter a log comment, then click **OK**. This should create the tag!

To verify that tagging was successful, open the following URL in a web browser (replacing *YOUR_TEAM* with the appropriate name):

https://utopia.cs.memphis.edu/course/comp7012-2013spring/teams/YOUR_TEAM/GameDen/tags/

You should see an **hw3** folder, and within that folder should be **src** and **WebContent** folders along with the **README.txt** file. Everyone's HTML files should be in the **WebContent** folder.