

# Homework 6: Sessions

For this homework, you will practice using conversational state in your web app by using sessions in your ongoing **GameDen** project.

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. Use sessions in your dynamic web page

For each game a user starts, you must store the state of the game in a session object. Thus, a player should be able to play two separate instances of the same game using two separate browsers. (If you have been storing game state using hidden input fields in the form, you must change that.) The games should still store final results and other stats in the database.

All web pages must try to retrieve user info (e.g., name) from the session first. If the session contains no such info, then the next step is to try to retrieve the info from the database. If user info is found in the database, then that info should be stored in the session.

Submitting the user info form must store the entered info in a session as well as writing it to the database.

Note that the above instructions set up the potential for a TOCTTOU (Time Of Check To Time Of Use) error in the servlet because the database can be updated after the servlet loads the session. You need not fix the error for this homework; however, you should be aware of it.

Don't forget to synchronize accesses to the session.

Note that for this homework, the system will still support only one user.

For an example of that uses sessions, checkout this project:

<https://utopia.cs.memphis.edu/course/comp7012-2013spring/examples/PingPong-Sessions/trunk/>

IMPORTANT! Session caching may cause you problems. Both Tomcat and your web browser cache session-related information. Clearing your web browser's session info should take care of most problems. Most browsers clear session info when quit/restarted. For Eclipse's browser, it seems that you must restart Eclipse to clear its session info (Annoying!). For good measure, I also like to ensure that Tomcat clears its session info on restarts by including the **context.xml** file from the above example in the **META-INF/** folder.

## Step 2. 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 **hw6** (case sensitive, no spaces).

To tag the current revision of your trunk as **hw6**, 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 **hw6** 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/](https://utopia.cs.memphis.edu/course/comp7012-2013spring/teams/YOUR_TEAM/GameDen/tags/)

You should see an **hw6** 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.