

# Homework 5: Sessions and Unit Tests

For this homework, you will practice (1) using sessions to implement conversational state and (2) using unit tests to test your model classes.

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

## Step 1. Add conversational state

Each page must keep a count of the number of times that the user has visited that page during the current session. To implement these features, create a controller class that has a counter for each servlet. When a user first requests a servlet (i.e, at the beginning of a session), create an instance of the counters class, increment the appropriate counter, and store the instance in a session object. When the user requests subsequent servlets, each servlet should increment the appropriate counter. At the top of each page returned by a servlet, print the message “Servlet hit #: 99”, where 99 is the current count.

Each team member should implement the above for the servlets/JSPs that they created in **hw3**.

Don't forget to synchronize accesses to the session.

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

<https://utopia.cs.memphis.edu/course/comp4081-2013fall/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!).

## Step 2. Create unit tests

Add black-box unit tests for each of the following model classes, dividing the work among your team by task:

- Task 1: Test class **User**
- Task 2: Test class **Movie**
- Task 3: Test class **PopularityCalculator**
- Task 4: Test class **MovieScore**
- Task 5: Test class **PlayList**
- Task 6: Test class **Review**

Thoroughly exercise each classes features. For a class with only getter/setter methods, this might involve loading an instance of the class up with “typical” values, and testing to see that the getter methods return the appropriate values. For a class that computes something, test typical and boundary cases.

### 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 **hw5** (case sensitive, no spaces).

To tag the current revision of your trunk as **hw5**, 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 **hw5** 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/comp4081-2013fall/teams/YOUR\\_TEAM/movieclub/tags/](https://utopia.cs.memphis.edu/course/comp4081-2013fall/teams/YOUR_TEAM/movieclub/tags/)