

# Homework 4: Add a Database Backend

For this homework, you will practice working with databases by adding a database backend to the web app that you created in the last homework.

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 0. Switch to new trunk

In your team's SVN repository, look in the **branches** subfolder of the **xleague** project folder. You will find a folder named **hw4-trunk**, which contains a complete implementation of the webapp from Homework 3. You must rearrange your repository, so that

- your old **trunk** is moved into your branches folder as **hw3-trunk**
- the **hw4-trunk** replaces your old **trunk**

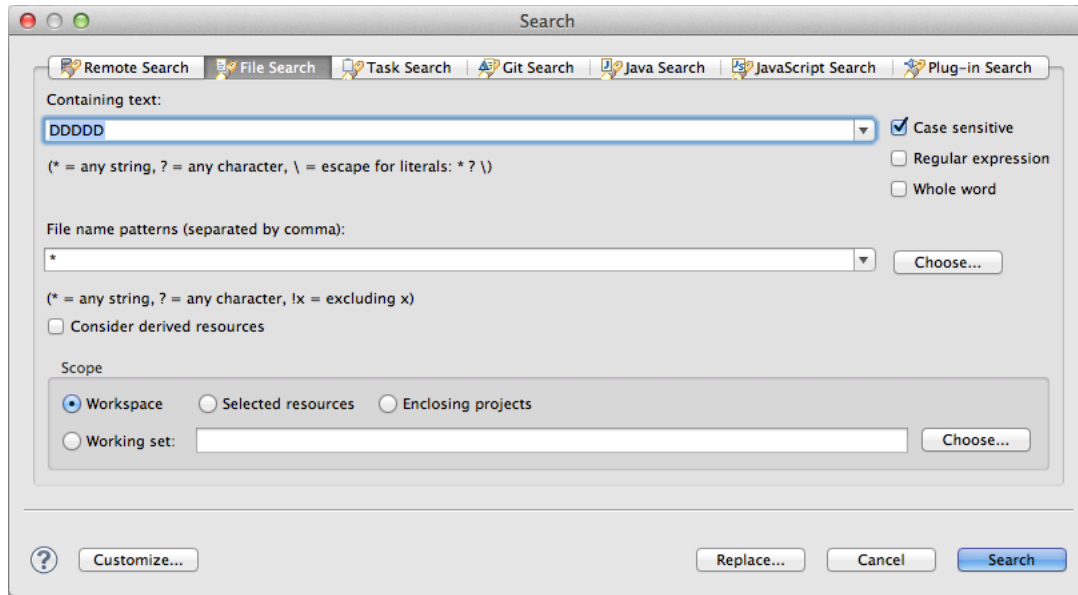
It's probably best if one person does the above, commits the changes, and then has everyone else update their working copies. Note that in this case, you may need to discard your old working copies and do a fresh checkout of the new **trunk**. Also, make sure to move things around in such a way that *everything* in the **hw4-trunk** is moved over – I made many changes!

## Step ½. Find/replace some names

For this assignment, each team will have their own database; however, that means that the database configuration will not be exactly the same for each team. I have put inserted placeholders throughout your project files that you must replace with the proper values. Here the placeholders you must replace:

- DDDDD
  - Replace with the name of your database, which should be *yourTeamName\_dbx* (e.g., **calypso\_dbx**).
- UUUUU
  - Replace with your team's DB username, which should be *yourTeamName\_user* (e.g., **calypso\_user**).
- PPPPP
  - Replace with your team's DB password, which I will provide to you.

To do the find/replace, you can use Eclipse's File Search tool. To access the tool, do **Search > File...** Here's what it looks like:



Note that the placeholders may not be only in Java files, so you must put a glob (\*) in the **File name patterns** field. Also note that there is a **Replace...** button at the bottom of the window.

## Step 1. Add database backend to xleague

Your main job this homework will be to convert the **xleague** project to use a database backend. To do this conversion, you must do two things:

- Implement JDBC DAO classes. I have provided code skeletons for each of them, so your job will actually be to fill in their methods.
- Replaces instantiations of the “Fake” DAO classes in the code with instantiations of the JDBC DAO classes.

The work involved with the first part will be divided up and shared among the team members. Similar to past homeworks, I have created five tasks, one for each team member:

- Task 1: Implement the DAO methods
  - `PlayerDaoJdbc.selectPlayerWithId()`
  - `TeamDaoJdbc.updateTeam()`
  - `RosterDaoJdbc.dropPlayerFromTeam()`
- Task 2: Implement the DAO methods
  - `TeamDaoJdbc.createTeam()`
  - `TeamDaoJdbc.selectAllTeams()`
- Task 3: Implement the DAO methods
  - `TeamDaoJdbc.selectTeamWithId()`
  - `TeamDaoJdbc.deleteTeam()`
  - `RosterDaoJdbc.addPlayerToTeam()`
- Task 4: Implement the DAO methods
  - `PlayerDaoJdbc.selectAllPlayers()`
  - `PlayerDaoJdbc.deletePlayer()`
- Task 5: Implement the DAO methods

- `PlayerDaoJdbc.updatePlayer()`
- `RosterDaoJdbc.selectAllPlayersOnTeam()`

Note that I have implemented `PlayerDaoJdbc.createPlayer()` for you.

Also, I have created the tables in your database and initialized them with test data. I have included the following SQL scripts in the new trunk:

- **src/main/db/create\_tables.sql**
  - Use this script to create the database tables (empty).
- **src/main/db/drop\_tables.sql**
  - Use this script to delete all the database tables and the data they contain. I included this script just in case you want to clear out your database and start from scratch.
- **src/test/db/insert\_test\_data.sql**
  - Use this script to load up the tables with test data.

For an example of a simple Java EE web app with a database backend, checkout this example:

<https://svn.cs.memphis.edu/svn/comp7012-2014spring/examples/booksrus-db/trunk/>

## 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 **hw4** (case sensitive, no spaces).

To tag the current revision of your trunk as **hw4**, 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 **hw4** 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 your team's repository URL in a web browser, and inspect what's in the repository.