CSCE 156 – Lab: Using JDBC in a Web Application I

Handout

0. Prior to the Laboratory

1. Review the laboratory handout
2. Make sure that the Albums database is installed and available in your mysql instance on CSE
3. Review the SQL and JDBC lecture notes
4. Review a JDBC tutorial from Oracle: http://download.oracle.com/javase/tutorial/jdbc/

1. Lab Objectives & Topics

Upon completion of this lab you should be able to understand how to:

- Write SQL queries for use in JDBC
- Make a JDBC connection, query and process a result set from a database
- Have some exposure to a multi-tiered application and a web application server

2. Problem Statement

In this lab you will familiarize yourself with the Java Database Connectivity API (JDBC) by finishing a simple, nearly-complete retrieve-and-display web application and deploying it to an application server (Glassfish) on CSCE.

The design of the webapp is simple: it consists of an index page that has a single link to a Java Server Page (JSP) that displays a list of albums queried from the album database you have worked with in prior labs.

It is not necessary to understand the details of the application (the HTML, JSP, JavaBeans, or application server). The main goal of this lab is to give you some familiarity with JDBC and exposure to a multi-tiered application and web application server environment.

3. Instructions

For this lab, you will need to use the JEE (Java Enterprise Edition) version of Eclipse, *not* the JSE (Java Standard Edition). In Windows, click the start menu and enter “Eclipse”, the “Java EE Eclipse” should show up, select this version. You may use the same workspace as with the JSE (Java Standard Edition) version of Eclipse.

Alternatively (and recommended) you can use the Eclipse version in the linux partition which will enable you to deploy locally to your lab machine instead of the csce server.

Running Eclipse in Linux

1. The glassfish server on your lab machine is only installed in the linux image. If you are not already in the linux image, restart your machine and choose openSUSE. Login with your usual CSE credentials.
2. Once logged into linux, go to Applications > Search and open up a Terminal
3. Launch eclipse by executing `/usr/local/bin/eclipse &`
   (the ampersand launches eclipse in the background so you can still use the terminal session)

Importing Your Project

1. Download the zip archive file from blackboard and import it into Eclipse:
   a. File -> Import -> General -> Existing Projects into Workspace
   b. Select "Select archive file" and select the zip file you downloaded
2. The java code is located under the "Java Resources -> src" directory
3. The JSP and web files are located under the "WebContent" directory

Modifying Your Application

1. You will first need to make changes to the `unl.cse.music.DatabaseInfo` java source file:
   a. Change instances of `YOURLOGIN` to your cse login
   b. Change instances of `YOURSQLPASSWORD` to your mysql password (if you have misplaced it, it can be reset by going to http://ponca.unl.edu)
2. All the JSP and most of the Java code is complete except for two methods in the `AlbumBean` class. You can make any modifications to the other classes, JSP or HTML files that you feel will help you, but modifying code in one part may break functionality in another. The two methods that you will need to implement are:
   - `public Album getDetailedAlbum(int albumId)` – this method will query the database for the specific album with the given primary key (`Albums.AlbumID` in the database) and return an Album object with *all* of its fields specified.
   - `public List<Album> getAlbums()` – This method will query the database and get a complete list of all Albums in the database. It will create and populate `Album` objects and put them in an `ArrayList` which will then be returned.

Hints

1. Refer to the course lecture slides on how to make a JDBC connection and query
2. Refer to the prior lab(s) for the mysql database schema
3. Refer to the `BandBean` class for another JDBC example
4. **Important**: do not forget to close any database resources (especially connections) after you are finished using them.
5. A main method has been included in the `AlbumBean` class that you can use to test your method implementations before you deploy your application. Make use of this method and

modify it as needed to ensure your code works (the glassfish server's log configuration will not make it convenient to debug your code once deployed).

Deploying Your Application

You have two options for deploying your application as a Web Archive file (WAR) to an application server. You may deploy locally to your lab machine (recommended) or you may deploy to the application server hosted on CSCE.

Running Eclipse in Linux

4. The glassfish server on your lab machine is only installed in the linux image. If you are not already in the linux image, restart your machine and choose openSUSE. Login with your usual CSE credentials.
5. Once logged into linux, go to Applications > Search and open up a Terminal
6. Launch eclipse by executing `/usr/local/bin/eclipse &`
   (the ampersand launches eclipse in the background so you can still use the terminal session)

Deploying Locally

1. Right click your project and select "Export…", select Web, WAR file, Next
2. Click Browse and select a directory (your home directory is recommended) and file to export to; name the file `loginLab09.war` where "login" is replaced by your cse login.
3. Return to your terminal and copy the WAR file to glassfish's autodeploy directory:
   `cp loginLab09.war /var/glassfish3/glassfish/domains/domain1/autodeploy`
4. Open a web browser (Applications > Firefox) and go to the following url:
   http://localhost:8080/loginLab09
   where login is replaced with your CSE login.

Deploying to CSCE

1. Export your application in a Web Archive File (WAR)
   a. Right click your project and choose Export -> Export… -> Web -> WAR file
   b. Click Browse and select a directory to export it to, name the file `loginLab09.war` where "login" is replaced by your cse login.
2. The file should now be in the directory you selected.
3. Copy the WAR file to the root of your CSE directory (copy it to your Z: drive)
4. Make sure that the file has proper permissions by executing the following command:
   `chmod 755 loginLab08.war`
   where, "login" is replaced by your cse login.
5. SSH (putty) into `csce.unl.edu` (*not* `cse.unl.edu`) using your cse login
6. Copy the war file to the server's glassfish auto deploy directory:
   `/usr/local/glassfishv3/glassfish/domains/domain1/autodeploy`
   Example:
   `cp loginLab08.war`
   `/usr/local/glassfishv3/glassfish/domains/domain1/autodeploy`
7. Open a web browser and go to the following URL:
   http://csce.unl.edu:8080/loginLab08/

where “login” is replaced by your CSE login.

Unless there were problems with your code, the webapp should now work, you can click on album titles or bands to be taken to another page that gives further details.

4. Completing Your Lab

1. Complete the worksheet and have your lab instructor sign off on it.
2. Before you leave, make sure you “undeploy” your web app by deleting the WAR file from the glassfish autodeploy directory:
   `rm /usr/local/glassfishv3/glassfish/domains/domain1/autodeploy/loginLab08.war`

Advanced Activities (Optional)

1. Look at the `albumList.jsp` page and observe how links were created to take you to a detailed album page. Modify the `albumDetail.jsp` page to make it so that you can also click the Band’s name and be taken to the `bandDetail.jsp` page.
2. The album list page (`albumList.jsp`) produces a table which is difficult to read. One common way to make tables more readable is to alternate slightly the background color on table cells. Modify the JSP page’s for-loop and add some CSS (Cascading Style Sheets) to alternate the color from one row to the next.
3. Many javascript plugins are available to add additional functionality to a plain HTML table (the ability to sort, pagination, column rearrangement, searching/filtering, etc.). One of the best plugins is datatables, a jQuery plugin available at: http://datatables.net/. Download and incorporate the datatable’s code into your project and add the appropriate javascript code to make your Album table more dynamic.