

# CSCE 156 – Computer Science II

## Lab 9.0 - JDBC in a Webapp I

### Prior to Lab

1. Review this laboratory handout prior to lab.
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/>

### Lab Objectives & Topics

Following the lab, you should be able 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

### Peer Programming Pair-Up

To encourage collaboration and a team environment, labs will be structured in a *pair programming* setup. At the start of each lab, you will be randomly paired up with another student (conflicts such as absences will be dealt with by the lab instructor). One of you will be designated the *driver* and the other the *navigator*.

The navigator will be responsible for reading the instructions and telling the driver what to do next. The driver will be in charge of the keyboard and workstation. Both driver and navigator are responsible for suggesting fixes and solutions together. Neither the navigator nor the driver is “in charge.” Beyond your immediate pairing, you are encouraged to help and interact and with other pairs in the lab.

Each week you should alternate: if you were a driver last week, be a navigator next, etc. Resolve any issues (you were both drivers last week) within your pair. Ask the lab instructor to resolve issues only when you cannot come to a consensus.

Because of the peer programming setup of labs, it is absolutely essential that you complete any pre-lab activities and familiarize yourself with the handouts prior to coming to lab. Failure to do so will negatively impact your ability to collaborate and work with others which may mean that you will not be able to complete the lab.

## JDBC in a Web Application

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). The design of the webapp is simple: it consists of an index page that loads album data via Ajax (Asynchronous JavaScript and XML) and displays it in a table.

It is not necessary to understand the details of the application (the HTML, JavaScript, Servlets, 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.

## Getting Started

**Note:** Unless you are familiar with Java Applications Servers and have one installed on your laptop, you will need to use the lab computers for this lab. In addition, it would be a good idea to reset your Albums database by rerunning the SQL script from a prior lab.

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. To use this version of Eclipse:

1. Restart your lab computer 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)

Once you’ve got Eclipse running, clone the project code for this lab from GitHub using the URL, <https://github.com/cbourne/CSCE156-Lab09>. Refer to Lab 1.0 for instructions on how to clone a project from GitHub.

## Activities

### Modifying Your Application

1. You will first need to make changes to the `unl.cse.albums.DatabaseInfo` source file. In particular, change the login and password information to your MySQL credentials. You can reset these by going to <http://cse.unl.edu/check>.
2. The HTML, JavaScript, etc. has been provided for you. Feel free to make modifications these files, but you should know what you are doing as changes can break functionality in other parts of the application.
3. The application will not display any album data until you have completed the methods in the `Album` class.
  - `public static List<Album> getAlbumSummaries()` – 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 a list which will then be returned. This method will be used to generate the album table, so it doesn’t need all information, just a subset (see the documentation as to what is required). You should optimize your queries to only select the relevant columns.
  - `public static Album getDetailedAlbum(int albumId)` – this method will query the database for the specific album with the given primary key and return an `Album` instance with *all* relevant data (band and its members, songs, etc.) specified.
  - Important: do not forget to close your database resources (especially connections) after you are finished using them.

- A `Test` class has been provided for you to test your `getAlbumSummaries()` method which you can also adapt to test your `getDetailedAlbum()` method. You should use it to debug your methods before deploying your application.

## Deploying Your Application

You have two options for deploying your application as a Web Archive (WAR) file to an application server.

### Linux: Deploying Locally

If you chose to do this lab in the Linux version, then follow these instructions.

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 ~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.

### Windows: Deploying to CSCE

If you chose to do this lab in the Windows version, then follow these instructions.

1. Export your application in a Web Archive File (WAR): right click your project and choose “Export” → “Export...” → “Web” → “WAR file”
2. Click “Browse” and select a directory to export it to, name the file `loginLab09.war` where `login` is replaced by your cse login. 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. SSH (PuTTY) into `csce.unl.edu` (*not* `cse.unl.edu`) using your cse login
5. Copy the war file to the server’s glassfish auto deploy directory with the command  

```
cp loginLab09.war ~glassfish/glassfish/domains/domain1/autodeploy/
```

6. Make sure that the file has proper permissions by executing the following from the command line:

```
chmod 755 ~glassfish/glassfish/domains/domain1/autodeploy/loginLab09.war
```

where, `login` is replaced by your cse login. If necessary, you may need to give glassfish a kick in the butt by executing:

```
touch ~glassfish/glassfish/domains/domain1/autodeploy/loginLab09.war
```

7. Open a web browser and go to the following URL: <http://csce.unl.edu:8080/loginLab09/> 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.

## Completing Your Lab

Complete the worksheet and have your lab instructor sign off on it. Before you leave, make sure you “undeploy” your web app by deleting the WAR file from the glassfish autodeploy directory:

```
rm ~glassfish/glassfish/domains/domain1/autodeploy/loginLab09.war
```

## Advanced Activities (Optional)

1. The album listing page utilizes a web framework called Bootstrap (see <http://getbootstrap.com/>). However, the album detail and band detail pages do not. Take this opportunity to learn about Bootstrap and use it to add stylistic elements to these pages.
2. 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 datatable’s code into your project and add the appropriate JavaScript code to make your Album table more dynamic.