

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: For this lab, you will need to use the JEE (Java Enterprise Edition) version of Eclipse, not the JSE (Java Standard Edition). If you have this installed you should be able to use your own machine, but if not you will need to use one of the lab computers. 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.

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.

Running Your Application

When working with a Java Web Application you generally build a Web Archive (WAR) file and *deploy* it to an application server. For this lab, we'll "deploy" locally using Eclipse. To start your application follow these instructions.

1. Change to the "Java EE" perspective in Eclipse

2. Right click your project and select “Run As” → “Run on Server”
3. Select “J2EE Preview as localhost”
4. After a brief startup, a preview browser window should appear and you can interact with your application.
5. To stop the server, hit the stop button in the terminal
6. You may need to restart the server if you make any changes to your code.

Completing Your Lab

Complete the worksheet and have your lab instructor sign off on it.

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.