

Waukesha County Technical College

152-198 Distributed Java

Class 7 Plan and Assignments

Discussion Activities:

- **Due Today and Announcements:**
 1. Calculator labs due today on GitHub
 2. JSTL and EL reading assignments should be completed by today
 3. “JstlExperiments” practice project was due a while ago but will be reviewed today.
- **Q&A**
- **Solutions to the “Calculators” MVC project**
 - ✓ Student solutions
 - ✓ Instructor solution
 - Lessons learned: never pass the HttpServletRequest or HttpServletResponse object to your model. If you do your model will be rigidly dependent on those controller objects, making your code less portable and less flexible.
- **Review of Student Work on JstlExperiments project**

Introduction to JSTL (Java Standard Tag Library) and EL (Expression Language)

- Syntax and basic usage (see reference material below)
- Advantages:
 - Reduce coupling to server side objects
 - Reduce complexity by removing Java code from view and replacing with what are essentially macros that do things simply and with less code
 - Reduce labor
 - Fails gracefully – no exceptions to worry about!
- Disadvantages:
 - Two more “languages” to learn
 - Hard to debug
 - Not designer friendly
- Important To Dos
 - **SQL Functions**: Do not use! Violates Single Responsibility Principle in VIEW pages.
 - **i18n**: for internationalization. Don’t use now. We’ll discuss later in semester.
 - **EL**: you do not need anything special in your JSP pages to use EL, however, you must be in a JSP page and you must be aware of the version of EL you are using (we are using the EL version that works with JEE 7)

- **JSTL:** you must be in a JSP page and you must add taglib statements at the top of the page. Here are the most common taglibs which you can place at the top of every JSP even if you don't use them (no performance penalty):

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@taglib prefix="sec" uri="http://www.springframework.org/security/tags" %>
```

Making CSS Easier, Making Your Apps Look More Professional:

Beginner's Introduction to Twitter Bootstrap

- You can download it and read the documentation at: <http://getbootstrap.com/>
- However, it's better to use a CDN (Content Delivery Network). Sample code for doing this is shown at the bottom of this class plan.
- See the Bootstrap documentation for CSS, Components and JavaScript
- Instructor demo

Bootstrap Sample Code for using CDN

1. Put this in the head section of your page, before your custom.css:

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
```

2. Put this at the end of your html code, just before the closing `</body>` tag, and before any custom ".js" files of your own:

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
```

3. Best practices for CSS and JS tag placement: always put your CSS link tags in the `<head>` section of your web page, and always put your `<script>` tags just above the closing `</body>` tag.

Lab:

- For this lab we are going to start working with a practice project that we will use for the rest of the semester. This project will manage authors and, eventually, the books associated with those authors. This data will eventually come from a database, but for now we will just hard code the data into a class. And for now, we will just work with authors. No books yet.
- Create a new Maven-based web project called "bookWebApp". Git-enable this project, perform an add and commit, and push to GitHub after creating your empty GitHub project. Finally, create and checkout a new Git branch named "NoDb". We will work with this branch today.
- Using the MVC pattern create a home page with links to various administrative actions. For now just create a link to view all authors. Clicking this link should send a request to an "AuthorController" for a list of author objects (entities), which it should retrieve from an "AuthService" model object that hard codes author objects into a list. This list should then be sent back to a `authorList.jsp` page. Use the provided Author class to instantiate your author objects.

- In the authorList.jsp page use JSTL and EL Expressions to display a list of authors in a table. Use columns for id, name and date added. Use a JSTL “<c:forEach>” tag to loop over the list. Also use JSTL choose-when-otherwise to check the “rowCount.count” value of the “varStatus” attribute of the JSTL forEach tag. Use the modulus operator to see if this is an even number, and then set the background-color style of the row to white, or if the value is odd, set it to some other light color, providing a greenbar effect to your report.
- Use EL expressions to extract author information in your table columns. For example, given a JSTL forEach tag with a “var” attribute that has a value of “a” (an author object) and a “items” attribute with a value of “authors” you can retrieve the author name like this: `${a.authorName}`
- Use Bootstrap in your web pages.

Textbook Chapters (and other resources) covered:

- JSTL and EL Quick Reference PDF on Blackboard
- Official EL Reference: <https://docs.oracle.com/javaee/7/tutorial/jsf-el.htm#GJDDD>
- Official JSTL Reference (outdated JEE 5):
<http://docs.oracle.com/javaee/5/tutorial/doc/bnake.html>
- EL and JSTL tutorials and references listed in Class 4 Plan
- Titter Bootstrap: <http://getbootstrap.com/>
- CSS Tutorials: <http://w3schools.com/css>
- Java EE v1.7 tutorial: <http://docs.oracle.com/javaee/7/tutorial/doc/home.htm>
- Java SE API (v1.8): <http://docs.oracle.com/javase/8/docs/api/>
- Java EE API (v1.7): <http://docs.oracle.com/javaee/7/api/>
- Online tutorials for client-side: <http://w2schools.com>
- Netbeans web development tutorials: <https://netbeans.org/kb/trails/java-ee.html>
- Netbeans Git User Guide: <http://netbeans.org/kb/docs/ide/git.html>
(don’t use SSH – we’ll be using the modern HTTPS approach)

Preparation Work for Next Class – 2 points for completion on time by next class:

1. Complete any unfinished lab challenges
2. Continue research and practice with Bootstrap.
3. Continue research and practice with the use of JSTL (Java Standard Tag Libraries) and EL (Java Expression Language) to simplify your code
4. **Review the IntroJDBC sample project** originally provided in Advanced Java and now provided again for your use (see Blackboard). Before reviewing the sample code, read this online tutorial: <http://docs.oracle.com/javase/tutorial/jdbc/> . Then, examine the sample code in IntroJDBC, looking specifically at the “SimpleDB_MySql_Demo.java” file in the “introjdbc” package, and then all of the files in the “lab1” package.