SENG2050 - Lab 03

February 26, 2018

Note: You may need to set your CLASSPATH and JAVA_HOME on the lab computer each time you log in.
Note: Create a webapps/lab03/ directory to complete these tasks. These tasks build on last week's, you way want to copy last week's lab content into this week's lab folder.
Note: When working in a UON lab, you will not be able to access any pages with Chrome or Firefox. On Edge you can access pages via http://localhost:8080. Do not forget to add http://.
Note: Remember to **NEVER** copy text straight out of this lab sheet.

1 Java Server Pages

The main objective of this tutorial part is to introduce you to Java Server Pages (JSP).

- JSPs should be saved in:

  ```
  apache-tomcat/webapps/[WEBAPP_NAME]/
  ```

  along with your static (HTML, CSS, JavaScript, etc.) content. i.e. For this lab they will go in:

  ```
  apache-tomcat/webapps/lab03
  ```

- JSPs should be accessed from:

  ```
  http://localhost:8080/lab03/[DOCUMENT_NAME].jsp
  ```

- JSP's are automatically compiled and changes are automatically updated (if you have a problem with the content not updating, delete everything under Tomcat's **work/** directory, this is where Tomcat caches compiled JSPs).

Introduction to JSP Tags

- `<%= java expression %>`
  - Java code within these tags is evaluated and the result is inserted into the servlet's output.
  - i.e. $<\%=3+3\%>$ and $<\%=$"6"$\%>$ will both output 6

- `<% java code %>`
  - Standard Java code goes between these tags.
  - i.e.

```
<%
  List myList = someList();
  for(int i=0; i<myList.size(); i++)
  out.println(myList.get(i).toString());
 %>
```

  - **Note:** you have access to the **JspWriter out** object by default (along with the **HttpServletRequest request** and **HttpServletResponse response** objects).

- `<%! Java declaration or method %>`
  - A declaration is a block of Java code that is used to define class-wide variables and methods in the generated Servlet class.

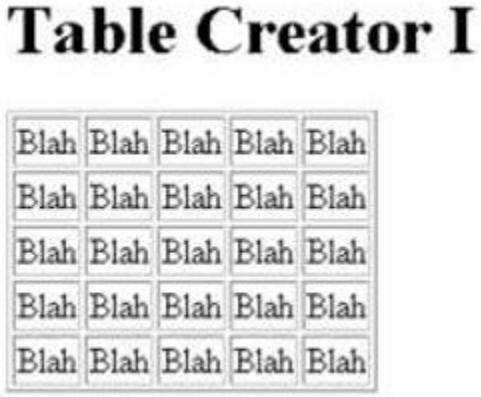
- `<%@page %>`
  - Page directives.
  - Used to provide instructions to the container in relation to the current JSP page.
  - Useful for importing other classes and libraries.
  - These generally go up the top of the JSP.
  - i.e.

```
<%@page import="java.io.IOException"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

2 Generating a Table

1. You should create a JSP which outputs something like this:

2. Now, create a page which generates an HTML table (see screenshot below) using **for** loops such that it is easy to modify the contents of the cells and the number of columns and rows.



3. Using the declaration tag, create a method:

```
void createTable(int numRows, int numCols, String cellContents, JspWriter out) throws IOException
```

which takes the four parameters and generates the table as in the previous exercise. You should re-use your code (the code shouldn't require much alteration at all).

3 Passing Values into a JSP

In this exercise you will make use of the request object's **getParameter()** method. Use a simple form to enter the number of rows, number of columns and the cell contents which then submits these data to a .jsp. The JSP will need to have code that extracts the given values and re-uses the method you wrote earlier (just copy and paste in this instance) to display a table to the user.

```
<form method=POST action="[DOCUMENT_NAME].jsp">
Rows: <input type="text" name="rows" size=5>
Columns: <input type="text" name="cols" size=5>
Cell Text: <input type="text" name="cellText" size=10>
<input type="submit" value="SEND">
</form>
```

4 Intro to JavaBeans

Modify the code from the previous exercise to use a JavaBean. To do this you will need to create and compile a JavaBean class and store it in a Java package in the directory:

```
webapps/lab03/WEB-INF/classes/[PACKAGE_NAME]
```

To include the bean in your JSP use:

```
<jsp:useBean id="[INSTANCE_NAME]" class="[PACKAGE_NAME].[CLASS_NAME]" scope="page">
```

The JSP should set all the bean's values using tag(s), i.e.

```
<jsp:setProperty ... />
```

You can access the bean's methods in your Java code by using the reference created by the jsp:useBean statement (the value of the **id** attribute).

Lastly, think about how you could make this more maintainable and discuss with the demonstrator.