JSP Demo

Create a simple servlet that uses JSP files for HTML

1. Create a project named JSPIntroDemo and set index.jsp to

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Test</title>
    </head>
    <body>
      <a href='JSPIntroServlet'>JSPIntroDemo</a>
    </body>
</html>
```

2. In Web Pages, create a JSP page named Hello.jsp and initialize it to

```
<html>
  <head>
    <title>Introduction to JSP demo - Hello page</title>
  </head>
  <body>
    <form method='post' action=''JSPIntroServlet'>
      <input type='submit' name='btnSubmit' value='Click me'/>
    </form>
  </body>
</html>
```

3. In Web Pages, create a JSP page named PostMessage.jsp and initialize it to

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title> Introduction to JSP demo - postback page </title>
    </head>
    <body>
      <p>This is the postback message</p>
    </body>
</html>
```

4. Create a servlet named JSPIntroServlet. When creating it look for the “Configure Servlet Deployment” screen in the New Servlet wizard. You will need to check the “Add information to deployment descriptor” box. Then add the following code:

```
package edu.mum.cs545;

import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class JSPIntroServlet extends HttpServlet
{
  protected void doGet(HttpServletRequest request, HttpServletResponse
response)
  throws ServletException, IOException
  {
    RequestDispatcher dispatcher = request.getRequestDispatcher("Hello.jsp");
    dispatcher.forward(request, response);
  }

  protected void doPost(HttpServletRequest request, HttpServletResponse
response)
  throws ServletException, IOException
  {
    RequestDispatcher dispatcher =
request.getRequestDispatcher("PostMessages.jsp");
    dispatcher.forward(request, response);
  }
}
```

5. F6 (there might be a bug to resolve)

6. After running a JSP page, you can view the servlet that was created by right-clicking the the JSP page in the Project Window and choosing "View Servlet".

Insert JSP scripting elements into JSP demo.

Insert the following scripting elements after the postback message in PostMessage.jsp, then run and inspect View Servlet.

```
      <%! int count = 0; %>
      The count is now:
      <%= ++count %>

      <%-- This is a jsp scriptlet that increments the count --%>
      <!-- This is an html comment inserted by the increment comment -->
      <% count = count * 10; %>
      The count is now:
      <%= count %>
```

Create a JSP page that demonstrates maintaining state info in HTML controls between requests.

(Add the following to your JSPIntroDemo project you created above.)

1. Create a JSP page named Choose.jsp and initialize it to:

```
<html>
  <head>
    <title>Choose</title>
  </head>
  <body>
    <form method='post' action='ChooseServlet'>
      <p>Is JSP cool?</p>
      <input type='radio' value='1' name='radioJSPCool'><span>Yes</span><br/>
      <input type='radio' value='0' name='radioJSPCool'><span>No</span><br/>
      <input type='submit' name='btnSubmit' value='Submit'/>
      <p>Is JSF way cool?</p>
      <input type='radio' value='1' name='JSFwayCool'><span>Yes</span><br/>
      <input type='radio' value='0' name='JSFwayCool'><span>No</span><br/>
      <input type='submit' name='btnSubmit' value='Submit'/>
      <p>Is the moon made of cheese?</p>
      <input type='radio' value='1' name='moonCheese'><span>Yes</span><br/>
      <input type='radio' value='0' name='moonCheese'><span>No</span><br/>
      <input type='submit' name='btnSubmit' value='Submit'/>

    </form>
  </body>
</html>
```

2. Create a servlet named ChooseServlet and initialize it to

```
package edu.mum.cs545;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ChooseServlet extends HttpServlet
{

  protected void doGet(HttpServletRequest request, HttpServletResponse 
response)
  throws ServletException, IOException
  {
    RequestDispatcher dispatch = request.getRequestDispatcher("Choose.jsp");
    dispatch.forward(request, response);
  }

  protected void doPost(HttpServletRequest request, HttpServletResponse 
response)
  throws ServletException, IOException
  {
    String choice = request.getParameter("radioJSPCool");
    if (choice != null)
      System.out.println ("choice= "+ choice);
    else
      System.out.println("No choice made");

    RequestDispatcher dispatch = request.getRequestDispatcher("Choose.jsp");
    dispatch.forward(request, response);
  }
}
```

3. Add <a href="ChooseServlet">Choose</a> to index.jsp

4. F6

5. Click the Choose link

6. Select the Yes button and click Submit

7. Observe that the doPost method of Choose is called, that choice=1 is written to the web container (Glassfish) window. However, in the redisplayed page the Yes radio button is no longer selected. What we want is that the user of this form can enter some responses, then click on a Submit button to save that work, and then be able to continue working on the rest of the form. They should be able to still see all of their previous answers after clicking Submit.

8. Use your understanding of state management in servlets to fix this.

```
HINT:
<input type='radio' value='1' name='radioJSPCool' checked>
<span>Yes</span><br/>
```

8a. Try using an EL expression to generate the "checked" flag.
8b. Try using an object in your EL expression, e.g.,

class name: RadioState
fields:
String yesCheck
String noCheck