Lecture 9

Servlets and JSPs
My Servlet Directory
You pressed Page 1 Button
You entered firstName []
You entered lastName []
doPost() invoked on = /servlet/ServletSessionManagementDemo

First Name
Last Name: Poelman

Page 1  Page 2  Page 3  Reset

You pressed Page 1 Button
You entered firstName []
You entered lastName [Poelman]
doPost() invoked on = /servlet/ServletSessionManagementDemo

First Name
Last Name

Page 1  Page 2  Page 3  Reset
first it does a GET
two browsers executing the same servlet
see that the instance variable has been changed but the session variables are indeed local to that user
session

- session is private to a specific user of a page (html, servlet, JSP)
- It is shared across all the pages in a web application.
- It's basically a hashmap like object that is bound to a specific client's requests (usually via a unique id stored in a browser cookie)
Session Management

• How do you save information between HTTP requests that relate to a specific user?
  – hidden HTML fields
  – cookies
  – URL rewriting
  – a database
  – Session object of the servlet container
The GVBooksOnlineWithDB in the Lecture9 directory

• Show a multiple view (page) servlet
• Shows accessing the servlet session object
• Shows read/write the parameters of the POST
You have to ask for a session object to be created by the container.
Filled in the form and pressed SAVE
Saved it to disk, then reloaded it and saved the customer info into the session
Fields are loaded from the customer object in memory
Spawn a new browser and change the info and save
2 DIFFERENT Browsers viewing the same page and sharing the session? (same machine)
But from a different machine the session is different
you pressed Save

Customer to String()
1 Bill Poelman 212-217-2487 2307 beverly road manhattan NY 11111
fghfghfgh@erterte
Created a customer file on the server
Added the fields to the reason
Reloaded the data from disk:
Customer to String()
1 Bill Poelman 212-217-2487 2307 beverly road manhattan NY 11111
fghfghfgh@erterte
Goto display session attributes page
Session attributes are:
creation time = 1036631634946
Session ID = wkdp7b1ll
getMaxInactiveInterval = 15
getLastAccessedTime = 1036631634946

ActionToDo BUTTON = Save
Phone = 212-217-2487
Zipcode = 11111
Email = fghfghfgh@erterte
Address = 2307 beverly road
LastName = Poelman
City = manhattan
FirstName = Bill
State = NY
Then press the Get Session Info button
The same page visited twice
(see lastAccessedTime)
After a period of activity longer than the MaxInactiveInterval

• the session has timed out
• All the attributes I set are gone
• A new session has been created (SessionId is changed)
Session points

• Session is shared across servlets but not across clients of the web application
• Session is not persisted to disk
• Sessions can timeout, write your code to deal with the timeouts appropriately
• You must create the session object, not automatically created for every servlet executed (scalability issue)
3 browsers on the same machine – 2 sharing a session one that does not. Started #1 & #2 by clicking on the ICON, #3 I pressed Ctrl-N
“scopes” of a web application

- **Page** – per servlet instance and class variables are shared by all clients to the servlet – note some engines may instantiate multiple copies of a servlet, not just one.

- **Session** – bound to a client and shared across all the servlets in a web app accessed by that client.

- **Context (Application)** – shared by all clients and shared by all pages in the web application – servlets, JSPs, etc.
Relationship of servlets, clients, sessions and the web application context

Client
Session A
Servlet W

Client B
Session B
Servlet X

Client C
Session C
Servlet Y
Servlet Z

Context
JSPs a better servlet
(YAAWT -yet another actual WWW technology!)
Java Server Pages - JSP

• Intentionally created to mimic the format of Microsoft ASPs
• Translated into Java code that is inserted into a template servlet that is then compiled
• The servlet is then loaded into the servlet engine
• When a JSP is referenced the engine checks to see if the version in memory is older than the JSP file date, if it is it retranslates it, compiles it and loads it (unlike a servlet)
A simple JSP

DateDisplayJSP.jsp

```jsp
<%@page contentType="text/html"%>
<%@page import = "java.util.*"%>

<html>
<head><title>JSP Page</title></head>
<body>

<%-- <jsp:useBean id="beanInstanceName" scope="session" class="package.class" /><!--%>
<%-- <jsp:getProperty name="beanInstanceName" property="propertyName" /> --%>

The current time is <%= (new Date()).toString() %>

</body>
</html>
```
A fragment of the servlet that gets created via the translation process

```java
// HTML // begin [file="C:\Lectures\Lecture9\DateDisplayJSP.jsp";from=(0,33);to=(1,0)]
MappedCharData
    out.write("\r\n");
    out.write("\n");
// end
// HTML // begin [file="C:\Lectures\Lecture9\DateDisplayJSP.jsp";from=(1,32);to=(8,0)]
MappedCharData
    out.write("\r\n");
    out.write("\r\n");
    out.write("<html>
");
    out.write("<head><title>JSP Page</title></head>
");
    out.write("<body>
");
    out.write("</body>
");
    out.write("</html>
");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");
    out.write("The current time is ");
// end
// begin [file="C:\Lectures\Lecture9\DateDisplayJSP.jsp";from=(11,23);to=(11,52)]
Expression
    out.print( (new Date()).toString() );
// end
...
```
JSP “view” of the code
To see the generated servlet

- NetBeans will pop up the servlet code if your JSP has a translation/compilation error.
- You can’t directly edit that servlet!
Servlet “view” of the code
The browser view of the code

<table>
<thead>
<tr>
<th></th>
<th>hello</th>
<th>yellow</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>1</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>2</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>3</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>4</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>5</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>6</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>7</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>8</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>9</td>
<td>hello</td>
<td>yellow</td>
</tr>
<tr>
<td>Counting 0</td>
<td>Counting 1</td>
<td>Counting 2</td>
</tr>
</tbody>
</table>
JSP

• Directives – affect the overall structure of the servlet generated
  – <%! //directive goes here %>  
  – Static variables, methods, etc.

• Scripting Elements
  – <%= //java code goes here %>  
  – Code that is compiled into the service method

• Actions – special tags that affect the runtime – standard and custom
  – <jsp:useBean>
Directives

- `<%@ directivename attribute="value" attribute="value"%>`
  - page
  - include
  - taglib
Creating a JSP based page with a table

```jsp
<%@page contentType="text/html"%>
<html>
<head><title>JSP Page</title></head>
<body>

<p>Create a table in HTML</p>
<%-- <jsp:useBean id="beanInstanceName" scope="session" class="package.class" /> --%>
<%-- <jsp:getProperty name="beanInstanceName" property="propertyName" /> --%>
<div align="left">
  <table border="2" style="border-collapse: collapse" bordercolor="#111111" width="90%" id="AutoNumber1">
    <tr>    
      <td width="33%">This is a cell</td>
      <td width="33%">2</td>
      <td width="34%">3</td>
    </tr>
    <tr>    
      <td width="33%">4</td>
      <td width="33%">So is this</td>
      <td width="34%">6</td>
    </tr>
    <tr>    
      <td width="33%">7</td>
      <td width="33%">8</td>
      <td width="34%">And this</td>
    </tr>
    <tr>    
      <td width="33%">10</td>
      <td width="33%">11</td>
      <td width="34%">12</td>
    </tr>
  </table>
</div>
</body>
</html>
```
Request object

- A RequestDispatcher allows one JSP/Servlet to forward you to another JSP/Servlet/HTML page in the same web application via forward().
- Or you can include the output from a different JSP/Servlet via include().
- The URL should be relative without the HTTP://localhost:8080.
JSP code fragment showing a RequestDispatcher

```<
if (request.getMethod().equalsIgnoreCase("POST"))
{
    String buttonUserPushed =   (String)request.getParameter("ActionToDoBUTTON");

    if (buttonUserPushed.equalsIgnoreCase("Add or Update Customer"))
    {
        String URLToReference = "/Lecture9/GVBookstoreCustomerForm.jsp";
        RequestDispatcher currentDispatcher = request.getRequestDispatcher(URLToReference);
        currentDispatcher.forward(request, response);
        return;
    }

    if (buttonUserPushed.equalsIgnoreCase("Browse Customers"))
    {
        String URLToReference = "/Lecture9/GVBookstoreCustomerManager.jsp";
        RequestDispatcher currentDispatcher = request.getRequestDispatcher(URLToReference);
        currentDispatcher.forward(request, response);
        return;
    }

    if (buttonUserPushed.equalsIgnoreCase("Browse Orders"))
```
Summary

• Servlets have page, session, and context scope variables
• JSP are a markup friendly servlet format
• You can intermingle HTML and Java code via `<% %>` scriptlets
• RequestDispatcher class can be used to invoke other servlets
HW #4 grading criteria

1) Create 4 JSPs or 3 JSPs and 1 servlet (the CustomerInfo from HW#3)
2) The main page should use a POST and SUBMIT button that uses if statements to use a RequestDispatcher and the forward() method.
4) The main page should invoke 1 of three other JSPs - CustomerInfo, CustomerBrowser and OrderBrowser.
5) CustomerInfo should be a working JSP (or Servlet if you wish) that saves and loads a customer object from disk. You can hard code the file name that the object gets saved/loaded from
6) The CustomerBrowser and OrderBrowser JSPs can be dummy JSPs with only the list and buttons but no code to actually load the stuff from disk or display anything.
7) The Customer class must use serialization to save out individual customer objects and load them back in.
8) You MUST use localhost:8080 in all your URLs in your code!! Not your specific machine name or IP address or anything else.
Homework #4
Homework #4 main page

GV Books Online Main Administrator Menu

- Add or Update Customer
- Browse Customers
- Browse Orders
Homework #4 Customer Info JSP

This is a JSP based form

First Name
Last Name
Address
City
State
Zipcode
Phone
Email

Load  Save  Create Order  Reset
Homework #4  Customer Manager (Browser)
Homework #4 Order Manager (Browser)