IS345-Internet Applications

Lab 4 Session and Cookie

Session

We can use an HttpSession object to hold the conversational state across multiple requests. In other words, for an entire *session* with that client.(Store the client actions).
For each client, there should be an HttpSession object that hold the specific actions that client did.

1) Looking up the HttpSession object associated with the current request.

```
HttpSession session = request.getSession(true);
```

2) Extracting Information from a Session

getId, isNew, getCreationTime, getLastAccessedTime, getMaxInactiveInterval, getValue … etc.

```
HttpSession session = request.getSession(true);
ShoppingCart previousItems = (ShoppingCart)session.getValue("previousItems");
if (previousItems != null) {
doSomethingWith(previousItems);
} else {
previousItems = new ShoppingCart(...);
doSomethingElseWith(previousItems);
}
```

3) Inserting Information in a Session

```
session.putValue("previousItems", previousItems);
```

IS345-Internet Applications

Lab 4 Session and Cookie

Session Example

Create a new web-project:
In the index page, write the following code:

```
<html>
<head>
<title>Welcome to JSP and Servelt</title>
</head>
<body>
<h1>Join our email list</h1>
<p>To join our email list, enter your name and
email address below. <br>
Then, click on the Submit button.</p>
<form action="EmailServlet" method="get">
<table cellspacing="5" border="0">
<!-- The three text boxes represent parameters that will be passed to the JSP when the user
clicks the Submit button -->
<!-- The parameter names are firstName, lastName, and emailAddress, and the parameter
values are the strings that the user enters into the text boxes.-->
<tr>
<td align="right">First name:</td>
<td><input type="text" name="firstName"></td>
</tr>
<tr>
<td align="right">Last name:</td>
<td><input type="text" name="lastName"></td>
</tr>
<tr>
<td align="right">Email address:</td>
<td><input type="text" name="emailAddress"></td>
</tr>
<tr>
<td></td>
<td><br><input type="submit" value="Submit"></td>
</tr>
</table>
</form>
</body>
</html>
```

IS345-Internet Applications

Lab 4 Session and Cookie

Create a Servlet class which will take parameters from the index page and create a new session with user data: (Note, the servlet is created in package named email5)

//EmailServlet.java

```
package email5;
import java.io.*; // The java.io class is required because it contains the IOException class
import java.net.*;
import javax.servlet.*; //The javax.servlet class is required because it contains the ServletException class
import javax.servlet.http.*;
public class EmailServlet extends HttpServlet {
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
String firstName = request.getParameter("firstName");
String lastName = request.getParameter("lastName");
String emailAddress = request.getParameter("emailAddress");
/*
The first line of code generates a new session object, or retrieves an existing one.
The second line sees if the session is new by checking the value from isNew().
A true tells you the session was just created;
A false means this user already had a session and you need to invalidate it.
*/
HttpSession session = request.getSession (true);
if (session.isNew() == false) {
session.invalidate();
session = request.getSession(true);
}
//Fill the session object
session.setAttribute("session_fName", firstName);
session.setAttribute("session_lName", lastName);
session.setAttribute("session_Email", emailAddress);
```

IS345-Internet Applications

Lab 4 Session and Cookie

//redirct to the JSP page
response.sendRedirect("Show_Email_Entry.jsp");
out.close();
}
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}
}

Create a JSP page that take invoke the session created from email servlet and represent it.

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@page import="javax.servlet.http.*" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Check Session</title>
</head>
<body>
<!-- Get the session Variables
you can call the session: either by request.getSession or orignal session name directly -->
<%
String firstName = request.getSession().getAttribute("session_fName").toString();
String lastName = session.getAttribute("session_lName").toString();
String emailAddress = session.getAttribute("session_Email").toString();
%>
<h1>Thanks for joining our email list</h1>
<p>Here is the information that you entered:</p>
<table cellspacing="5" cellpadding="5" border="1">
<tr>
<td align="right">First name:</td>
<td><%= firstName %></td> <!--JSP Expression-->
</tr>
<tr>
<td align="right">Last name:</td>
<td><%= lastName %></td>
</tr>
```

IS345-Internet Applications

Lab 4 Session and Cookie

```
<tr>
<td align="right">Email address:</td>
<td><%= emailAddress %></td>
</tr> </table>
<p>To enter another email address, click on the Back button in your browser <br>
or the Return button shown below.</p>
<form action="Email_List.html" method="post">
<input type="submit" value="Return">
</form>
</body>
    </html>
```

IS345-Internet Applications

Lab 4 Session and Cookie

Cookie

Cookies are small bits of textual information that a Web server sends to a browser and that the browser returns unchanged when visiting the same Web site or domain later

By default, a cookie lives only as long as user opening his browser (which called Session), once the client quits his browser (Session closed), the cookie disappears.

1) Creating Cookies

A Cookie is created by calling the Cookie constructor, which takes two strings: the cookie name and the cookie value. Neither the name nor the value should contain whitespace or any of: [ ] ( ) = , " / ? @ : ;

Cookie cookie = new Cookie(“username”, name);

2) Reading and Specifying Cookie Attributes

getComment/setComment, getDomain/setDomain, getMaxAge/setMaxAge, getName/setName, getPath/setPath … etc**.**

3) Placing Cookies in the Response Headers

```
Cookie userCookie = new Cookie("user", "uid1234");

response.addCookie(userCookie); //sent cookie to the client
```

4) Reading Cookies from the Client

```
Cookie[] cookies = request.getCookies();

for (int i = 0; i < cookies.length; i++) {

Cookie cookie = cookies[i];

if (cookie.getName().equals(“username”)) {

String userName = cookie.getValue();

out.println(“Hello “ + userName);

break;

} }
```