JSP Implicit Objects and Forms

Content By: Bairon Vasquez

1

Week 10 - Day 2

Questions?

Agenda

❑ What are JSP implicit objects?

❑ How to handle form using JSP?

What are implicit objects?

❑ Implicit objects in JSP are the objects that are created by the web container automatically and the container makes them available to the JSP to access it

❑ Implicit objects are available only inside the _jspService() method hence cannot be accessed anywhere outside

Object	Class	Description
request	javax.servlet.ServletRequest	Reference to the current request
response	javax.servlet.ServletResponse	HttpServlet response object for the current request.
session	javax.servlet.http.HttpSession	Session associated with the current request
application	javax.servlet.ServletContext	Servlet context to which a page belongs
page	java.lang.Object	An instance of the JSP page's servlet class that processes the current request.

Object	Class	Description
pagecontext	javax.servlet.jsp.PageContext	Object to access request, response, session and application associated with a page.
out	javax.servlet.jsp.JspWriter	Object that writes to the response output stream.
config	javax.servlet.ServletConfig	Servlet configuration for the page.
exception	java.lang.Throwable	This objects is available in JSP pages where the page directive **isErrorpage is set as true** stating that it is an error page for some other JSP page. The error message can be displayed from this object.

Process Values in Servlet Context

❑ Values can be stored in servlet context for making it available for all servlets in the application by setting it as an attribute to the JSP implicit object named application

    ❑ application.setAttribute("visitorCount", count) – Sets attribute to the application context

    ❑ application.getAttribute("visitorCount") – Reads attribute from the application context

    ❑ application.removeAttribute("visitorCount") – Removes attribute from the application context

❑ JSP provides both *request* and *response* as implicit objects. All the HttpServletRequest and HttpServletResponse methods can be called on these objects

❑ Example:

```
request.getParameter("name") ;

response.sendRedirect("home.jsp");
```

❑ We will learn more on using these objects during assignments

platform
By Per Scholas

❑ Rendering Forms:

  ❑ Forms can be rendered as normal HTML forms inside JSP pages

  ❑ Forms can also be created dynamically by printing using the *out.print()* method if required

❑ Example:

```
<%
    out.print("<form>");

    out.print("<input type='text' name='userName'/>");

    out.print("</form>");
%>
```

Form Processing JSP (Cont.)

❑ Handling Forms Actions:

    ❑ Form actions can be handled by the same JSP or another servlet/JSP

        ❑ Example – When user clicks the login button in Login.jsp it can be handled by *Login.jsp* or *HomePage.jsp* or *LoginValidator.java* (Servlet)

    ❑ If the same JSP is handling the form action then the form action attribute need not be specified

        ❑ Example:

```
<form name="LoginForm" method="get">   // No action attribute specified.
```

Session Handling in JSP

❑ The session management techniques used in servlets are also applicable to handle session in JSP also

❑ The following are techniques used to manage session in JSP:

   ❑ URL rewriting

   ❑ Hidden Field

   ❑ Cookie

   ❑ Session Object

❑ We will look more in depth using the JSP implicit object *session* to manage sessions

❑ In JSP the HttpServlet object named session is automatically created by the container and passed to the service method for use

❑ By setting attributes to the implicit session object, session can be managed across pages in JSP

    ❑ session.setAttribute(attributeName, value) – Sets attribute to session

    ❑ session.getAttribute(attributeName) – Reads the attribute value

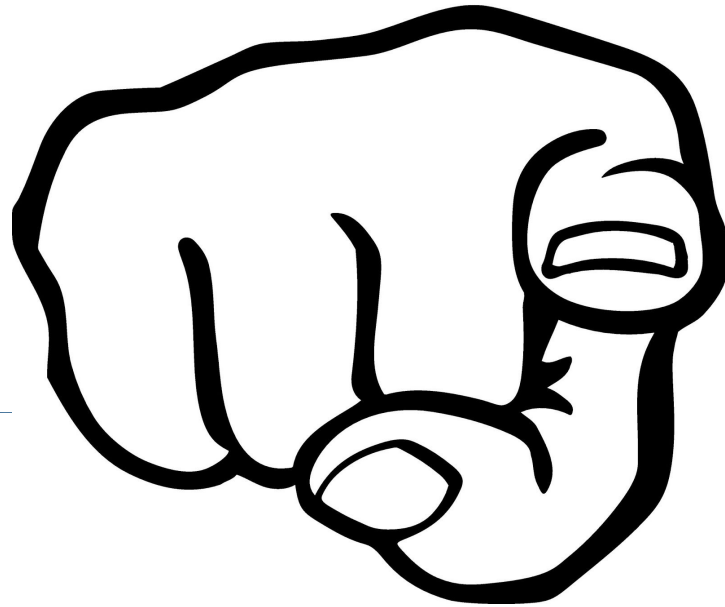   ❑ session.removeAttribute(attributeName) – Removes attribute from the session

Questions?

Your turn

Your Turn – JSP Form Processing

- ❑ In this demo we familiarize how forms can be developed and handled using JSP

- ❑ Here the form action will be handled by the same JSP itself

- ❑ JSPs to be developed:

  - ❑ login.jsp – Page which captures user name, password for login. The same page should be designed to process the login request and validated

    - ❑ If user name is "tom" and password is "jerry" the login is considered successful

  - ❑ success.jsp – In case of success the control should be redirected to the success page

Your Turn (Code Assistance)

❏ Sample Code (login.jsp):

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login</title>
</head>
<%
    String message = null;
    if (request.getParameter("login") != null) {
        String userName = request.getParameter("userName");
        String password = request.getParameter("password");
        if ("tom".equals(userName) && "jerry".equals(password)) {
            response.sendRedirect("success.jsp");
        } else {
            message = "Password and Username does not match";
        }
    }
%>
<body>
<%
    if (message != null)
        out.print("<span style='color:red'>" + message + "</span>");
%>
<form method="post">
UserName<input type="text" name="userName" />
<br />
Password<input type="text" name="password" /><br />
<input type="submit" name="login" value="Login" />
</form>
</body>
</html>
```

Performs the form processing as below,
1. If the request parameter value for the submit button is not null it means form is submitted else the form is accessed using URL.
2. The other parameter values are read using the get parameter method validate user name and password as mentioned.

Prints the error message

❑ Sample Code (success.jsp):

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!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=ISO-8859-1">
<title>Success Page</title>
</head>
<body>
<h1>You have Successfully Logged in</h1>
</body>
</html>
```
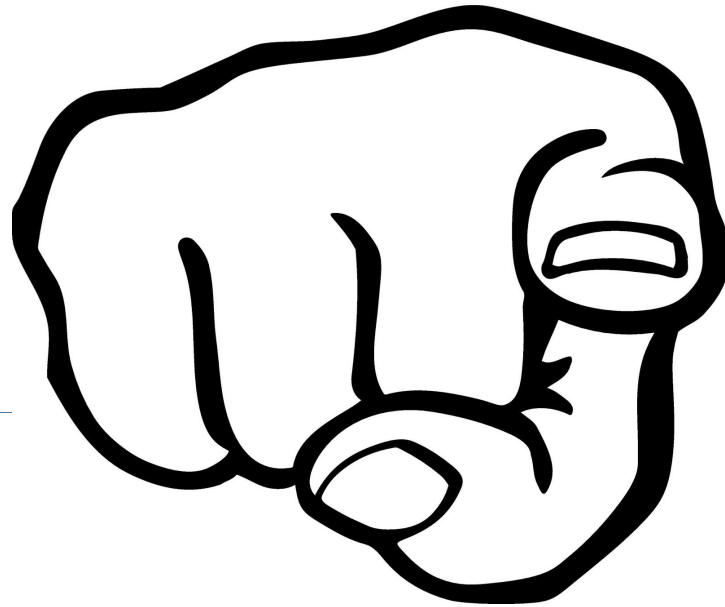
Your Turn – Registration Application

❑ A registration application should be implemented based on the requirement depicted below:

> ❑ The user invokes registration page from the browser – registration page rendered to the user
>
> ❑ User fills the registration form and submits the form – If all fields are not null the entered details should be displayed in a details page in a tabular form with name and value as the column headers
>
> ❑ User enters null value to any of the fields – The user is redirected to the same registration page with the appropriate error message printed

❑ The following JSP pages needs to be developed

   ❑ Registration.jsp – Page which captures registration details

   ❑ Details.jsp – This will process the request from registration.jsp, if any of the mandatory values is null the control should be redirected to the registration page

      ❑ If the entered details are proper display the details entered in a tabular format

   ❑ Error Message Format – "Password cannot be null"

Hints :
1. Make details.jsp the action page of registration form.
2. To pass the error message from details.jsp to registration.jsp set it as query string parameter to the URL
   Example :*response.sendRedirect("registration.jsp?message='User Name , Password cannot be null' ");*
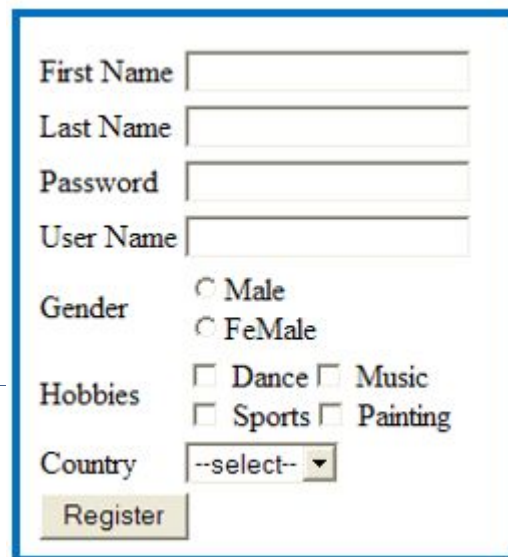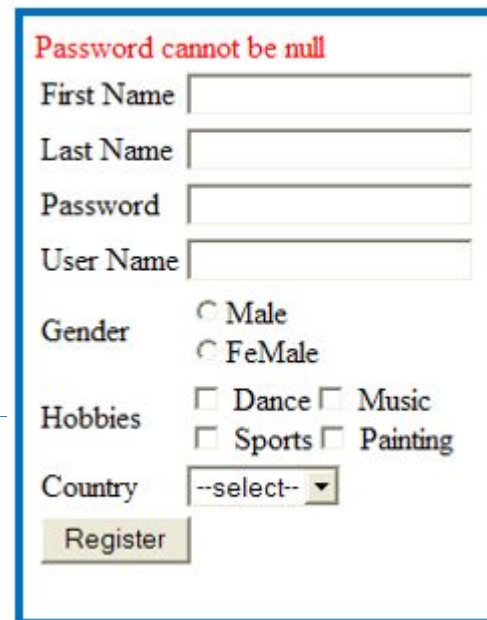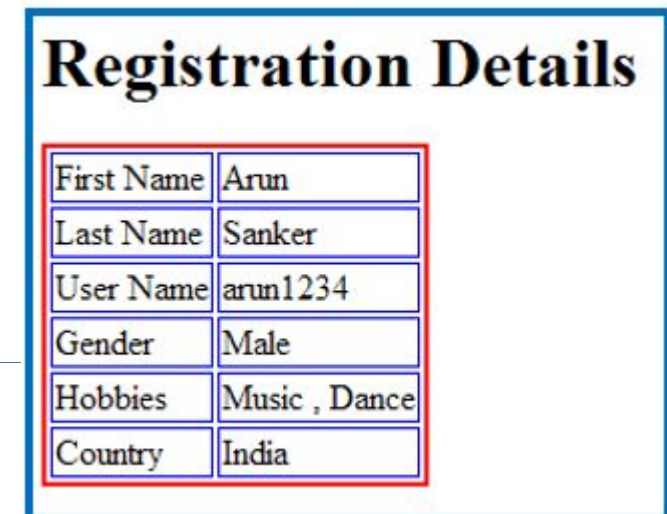
❑ Expected Displays on Webpages:

Initial   Error   Success

❑ This is a demo to showcase how session can be used in a JSP application by using the JSP implicit session obect

❑ Scenario – In a web based mail application user logins with his username and password. The subsequent pages inbox/sent items he visits after login should have a welcome message with his username printed. We will see how this can be achieved using session objects

❑ Components to be developed:

❑ login.jsp – Login page, on successful login the inbox page to be displayed. If username and password does not match "tom" & "jerry" redirect the control to the same login.jsp with the error message displayed

❑ Inbox.jsp – The inbox of the user, should have a link pointing to sent items.

Your Turn (Code Assistance)

❏ Sample Code (login.jsp):

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
<title>Login</title>
</head>
<%
    String message = null;
    if (request.getParameter("login") != null) {
        String userName = request.getParameter("userName");
        String password = request.getParameter("password");
        if ("tom".equals(userName) && "jerry".equals(password)) {
            session.setAttribute("userName", userName);
            response.sendRedirect("inbox.jsp");
        } else {
            message = "Password and Username does not match";
        }
    }
%>
<body>
<%
    if (message != null)
        out.print("<span style='color:red'>" + message + "</span>");
%>
<form method="post">UserName<input type="text" name="userName" />
<br />
Password<input type="text" name="password" /><br />
<input type="submit" name="login" value="Login" /></form>
</body>
</html>
```
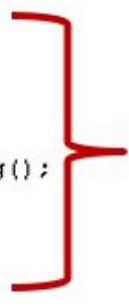
Performs the form processing as below,
1. If the request parameter value for the submit button is not null it means form is submitted else the form is accessed using URL.
2. The other parameter values are read using the get parameter method validate user name and password as mentioned.
3. On successful login the user name is set in session and redirected to inbox.jsp.

❑ **Sample Code (inbox.jsp):**

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Inbox</title>
</head>
<%
    String userName = null;
    if (session.getAttribute("userName") == null) {
        response.sendRedirect("login.jsp");
    } else {
        userName = session.getAttribute("userName").toString();
    }
%>
<body>
<h1 style="margin-left: 40%;">Inbox</h1>
<h3>Welcome  <%=userName%></h3>
<a href="sentItems.jsp">Sent Items</a>
<br/>
<a href="logout.jsp"> Log Out</a>
</body>
</html>
```

1. If user name is not set redirect to login page.
2. If user name is correct print the welcome message and display the sent item and logout link

❑ Sample Code (sentItems.jsp):

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Sent Items</title>
</head>
<%
    String userName = null;
    if (session.getAttribute("userName") == null) {
        response.sendRedirect("login.jsp");
    } else {

        userName = session.getAttribute("userName").toString();
    }
%>
<body>
<h1 style="margin-left: 40%;">Sent Items
</h1>
<h3>Welcome  <%=userName%></h3>
<a href="inbox.jsp">Inbox</a><br/>
<a href="logout.jsp"> Log Out</a>
</body>
</html>
```

1. If user name is not set redirect to login page.
2. If user name is correct print the welcome message and display the inbox and logout link

❑ Sample Code (logout.jsp):

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!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=ISO-8859-1">
<title>Insert title here</title>
</head>
<%
    session.invalidate();
    response.sendRedirect("login.jsp");
%>
</html>
```

On clicking logout links from "inbox" or "Sent Items" page the session is to be invalidated.

Time for
a Break

❑ JSP Implicit Objects:
  ➢ request, response, session, etc.

❑ Using form tags/session management
  ➢ request, response, session, etc.
  ➢ Setting and getting attributes