Java Server Pages (JSP)

Presented By: Sagar Giri

Topics Covered:

JSP Implicit Object

Object Scope

Synchronization Issue

Exception Handling

Session Management

Creating and Processing Forms.

JSP Implicit Object

Created by JSP Engine during translation phase (while translating JSP to Servlet).

They are being created inside service method so we can directly use them within Scriptlet without initializing and declaring them.

There are total 9 implicit objects available in JSP.

JSP Implicit Objects are the Java objects that the JSP Container makes available to developers in each page and developer can call them directly without being explicitly declared. JSP Implicit Objects are also called pre-defined variables.

9 - JSP Implicit Object

out - javax.servlet.jsp.JspWriter

request - javax.servlet.http.HttpServletRequest

response - javax.servlet.http.HttpServletResponse

session - javax.servlet.http.HttpSession

application - javax.servlet.ServletContext

exception - javax.servlet.jsp.JspException

page - java.lang.Object

pageContext - javax.servlet.jsp.PageContext

Config - javax.servlet.ServletConfig

Object Scope

Page Scope -

Objects with page scope are accessible only within the page in which they're created.

The data is valid only during the processing of the current response; once the response is sent back to the browser, the data is no longer valid.

If the request is forwarded to another page or the browser makes another request as a result of a redirect, the data is also lost.

Example:

```
<jsp:useBean id="employee" class="EmployeeBean" scope="page" />
```

Object Scope

Request Scope -

Objects with request scope are accessible from pages processing the same request in which they were created.

Once the container has processed the request, the data is released.

Even if the request is forwarded to another page, the data is still available .

Example:

```
<jsp:useBean id="employee" class="EmployeeBean" scope="request" />
```

Object Scope

Session Scope -

Objects with session scope are accessible from pages processing requests that are in the same session as the one in which they were created.

A session is the time users spend using the application, which ends when they close their browser, when they go to another Web site, or when the application designer wants (after a logout, for instance).

So, for example, when users log in, their username could be stored in the session and displayed on every page they access.

This data lasts until they leave the Web site or log out

Example:

```
<jsp:useBean id="employee" class="EmployeeBean" scope="session" />
```

Object Scope

Application Scope -

Objects with application scope are accessible from JSP pages that reside in the same application.

This creates a global object that's available to all pages.

Application scope uses a single namespace, which means all your pages should be careful not to duplicate the names of application scope objects or change the values when they're likely to be read by another page (this is called thread safety).

Application scope variables are typically created and populated when an application starts and then used as read-only for the rest of the application.

Example:

```
<jsp:useBean id="employee" class="EmployeeBean" scope="application" />
```

Synchronization Issue

```
<%! private int idNum = 0; %>

<%

String userID = "userID" + idNum;

out.println("Your ID is " + userID + ".");

idNum = idNum + 1;

%>
```

```
<%! private int idNum = 0; %>

<%

synchronized(this) {

String userID = "userID" + idNum;

out.println("Your ID is " + userID + ".");

idNum = idNum + 1;

}

%>
```

Session Management

Session in JSP can be managed in following ways:

- Cookies
- Hidden Form Fields
- URL Rewriting
- Session Object

Creating and processing forms

HTML Element:

<FORM ACTION="..." ...> ... </FORM>

Attributes:

ACTION

METHOD : GET/POST

ENCTYPE : default is application/x-www-form-urlencoded

Example: “Larry (Java Hacker?)” is sent as “Larry+%28Java+Hacker%3F%29”

Creating and processing forms

<FORM ACTION="http://localhost:8088/SomeProgram" ENCTYPE="multipart/form-data" METHOD="POST">

<!-- transmits each field as separate part of MIME compatible document -->

TARGET - where to redirect

ONSUBMIT - what to do while submitting (calling a JS function)

ONRESET - while resetting the form elements (calling a JS function)

ACCEPT - which file type do it accept

Thank You !