JSP

By Rahul Barve

Objectives

- Understand: What is JSP, Why JSP
- Explain: Life Cycle of JSP
- Working with JSP Tags
- Working with Implicit Objects

What is JSP

What is JSP

- JSP stands for Java Server Pages.
- JSP is a server side component that is used to extend the functionality of web server.
- Used to generate dynamic web content.

Why JSP

- Allows developers to concentrate on Presentation rather than Processing.
- Designers without knowing Java, still can develop elegant web pages.

JSP Life Cycle

- There are 3 life cycle methods:
  - `jspInit()`
  - `_jspService()`
  - `jspDestroy()`

JSP Tags

JSP Tags

- JSP Specification supports 3 types of tags:
  - Directives
  - Scripting Elements
  - Standard Actions

Directives

- Directives are divided into 3 categories:
  - page
  - include
  - taglib

Page Directive

- Used to specify some information about the page.
- Syntax:

```
<%@page attr=“value”...%>
```

Page Directive

- Attributes:
  - `language`
  - `extends`
  - `import`
  - `session`
  - `isThreadSafe`
  - `errorPage`
  - `isErrorPage`

Page Directive

- `language`
  - Specifies scripting language of the JSP page.
  - Possible Values: `java`
  - Default Value: `java`

Page Directive

- `extends`
  - Specifies fully qualified name of the class from which JSP specific Servlet implementation class will inherit from.

Page Directive

- `import`
  - Specifies fully qualified names of the classes or interfaces to be imported from the package other than `java.lang`.
  - E.g.

```
<%@ page import =
  “java.util.Vector, java.io.File” %>
```

Page Directive

- session
  - Specifies whether the page participates in a session or not and therefore whether the implicit object session will be available or not.
  - Possible Values: true / false
  - Default Value : true

Page Directive

- `isThreadSafe`
  - Tells container whether to make the JSP page thread safe or not.
  - It is in contrast to the thread safety concept of multithreading.
  - Possible Values: `true / false`
  - Default Value: `true`

Page Directive

- `errorPage`
  - Specifies the name of the JSP page to which control is to be diverted if the current JSP page contains some Java code that fires an exception.

Page Directive

- `isErrorPage`
  - Indicates whether the JSP page is an error page or not. If so, an implicit object exception becomes available in the page.
  - Possible Values: `true / false`
  - Default Value: `false`

Include Directive

- Used to include resources like HTML, JSP or Text files in JSP.
- Syntax:

```
<%@include file = “<filename>”%>
```

Scripting Elements

- Scripting Elements are divided into 3 categories:
  - Declaration
  - Scriptlet
  - Expression

Declaration

- Used to declare variables and define methods.
- E.g.

```
<%!
    int x = 100;
    public void myMethod(){
         //Some Code
    }
%>
```

Scriptlet

- Used to write any valid Java code.
- E.g.

```
<%
     //Java Statements
%>
```

- Statements written inside scriptlet execute inside the service method of the servlet.
- Not possible to define methods inside scriptlet.

Expressions

- Used to extract value of the variable.
- Expressions are directly processed on the browser window.
- E.g.

  <%=<expr>%>
- Methods returning ‘void’ cannot be invoked using expressions.

Implicit Objects

Implicit Objects

- request
- response
- out
- session
- config
- application
- page
- pageContext
- exception

Implicit Objects

- request
  - An object of type javax.servlet.http.HttpServletRequest that can be used to handle request specific parameters.
  - E.g.

```
<%
        String str =
        request.getParameter(“name”);
%>
```

Implicit Objects

- `response`
  - An object of type `javax.servlet.http.HttpServletResponse` that can be used to handle response.
  - E.g.

```
<%
        String url =
        response.encodeURL("hello");
%>
```

Implicit Objects

- `out`
  - An object of type `javax.servlet.jsp.JspWriter` that can be used to generate response through scriptlets.
  - E.g.

```
<%
        String str = "Welcome";
  out.println("<h1>" +str+ "</h1>");
%>
```

Implicit Objects

- session
  - An object of type `javax.servlet.http.HttpSession` that can be used to handle the information at the session level.
  - E.g.

```
<%
            session.setAttribute
            (“name”, “James”);
%>
```

Implicit Objects

- `config`
  - An object of type `javax.servlet.ServletConfig`

Implicit Objects

- application
  - An object of type `javax.servlet.ServletContext` that can be used to handle application level information.
  - E.g.

```
<%
        application.setAttribute
        ("message", "Welcome");
%>
```

Implicit Objects

- `page`
  - It is a reference that refers to an object of JSP specific servlet implementation class generated by the web container.

Implicit Objects

- pageContext
  - An object of type

    javax.servlet.jsp.PageContext that acts as an SPOC through which it is possible to obtain other resources..

Implicit Objects

- pageContext
  - E.g.

```
  <%
pageContext.
setAttribute(“a”,“Hello”,
  pageContext.REQUEST_SCOPE);

pageContext.
setAttribute(“b”,“Welcome”,
  pageContext.SESSION_SCOPE);
  %>

  %>
```

Implicit Objects

- exception
  - An object of type Java.lang.Throwable that can be used to handle exceptions occurred during the execution.
  - It is available only when isErrorPage attribute of page directive is set to true.

Let’s Summarize

- What is JSP, Why JSP
- Life Cycle of JSP
- JSP Tags
- Implicit Objects