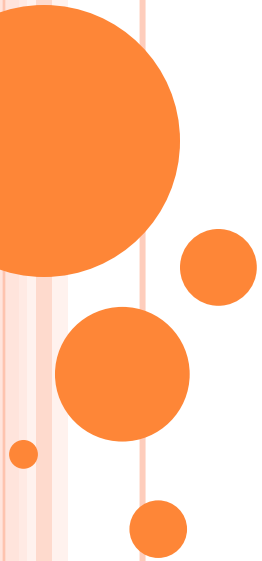

UNIT -5

Java Server Page

INDEX

INTRODUCTION

1. A JSP page is a page created by the web developer that includes JSP technology-specific tags, declarations, and possibly scriptlets, in combination with other static HTML or XML tags.
2. A JSP page has the extension .jsp this signals to the web server that the JSP engine will process elements on this page.
3. Pages built using JSP technology are typically implemented using a translation phase that is performed once, the first time the page is called. The page is compiled into a Java Servlet class and remains in server memory, so subsequent calls to the page have very fast response times.

LIFE CYCLE OF JSP

1. A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.
2. The following are the paths followed by a JSP
   - Compilation
   - Initialization
   - Execution
   - Cleanup

LIFE CYCLE OF SERVLET

1.JSP Compilation

1. When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.
2. The compilation process involves three steps:
   - Parsing the JSP.
   - Turning the JSP into a servlet.
   - Compiling the servlet.

LIFE CYCLE OF JSP

2.JSP Initialization:

1. When a container loads a JSP it invokes the jspInit() method before servicing any requests. If you need to perform JSP-specific initialization, override the jspInit() method:
2. Typically initialization is performed only once and as with the servlet init method, you generally initialize database connections, open files, and create lookup tables in the jspInit method.

```
public void jspInit(){
// Initialization code...
}
```

LIFE CYCLE OF JSP

3.JSP Execution :

- This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.
- Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the _jspService() method in the JSP.
- The _jspService() method takes an HttpServletRequest and an HttpServletResponse as its parameters as follows:
  - void _jspService(HttpServletRequest request, HttpServletResponse response)
  - {
    // Service handling code...
  - }
- The _jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods ie. GET, POST, DELETE etc.

LIFE CYCLE OF JSP

4.JSP Cleanup:

1. The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.
2. The jspDestroy() method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.
3. The jspDestroy() method has the following form:

```
public void jspDestroy()
{
  // Your cleanup code goes here.
}
```

Relation of Applet and Servlets with JSP

Relation of Applet with JSP :

1. An Applet is an special program that can be embedded into a web page.
2. An Applet can be added into JSP page.
3. The jsp : plugin action tag is used to embed applet in the jsp file.

Relation of Applets and Servlets with JSP

Relation of Servlets with JSP :

1. JSP and servlet are different technologies that can be run on web server to send HTTP response to the client.
2. Most of the JSP server are implemented to support the jsp as an extension of the servlet.
3. When a client request for JSP page for the first time it is compiled by JSP engine and converted into servlet code and further .exe file is created.

JSP Scripting Elements

1. Scripting JSP scripting elements allows programmers to insert Java code into JSP page.
2. In JSP Scripting element is started using<% and ended using %>.
3. **JSP scripting components :**
    1. Comments (**<%--** comment **--%>**)
    2. Scriptlets (**<%** scriplets **%>**)
    3. Declarations (**<%!** declaration **%>**)
    4. Directive (<%@ directive%>)
    5. Expressions (**<%= expression %>**)

JSP Tags

1.Comment tag

1. JSP Comment is used when you are creating a JSP page and want to put in comments about what you are doing. JSP comments are only seen in the JSP page. These comments are not included in servlet source code during translation phase, nor they appear in the HTTP response. Syntax of JSP comment is as follow
2. <%-- JSP comment --%>
3. Simple Example of JSP Comment

```
<body>
    <%-- Code to show page count --%>
</body>
```

JSP TAGS

2.Scripting tag

1. Scriptlet Tag allows you to write java code inside JSP page. Syntax of Scriptlet Tag is as follows :
2. <% java code %>
3. Simple example of scripting tag.

```
<html>
<body>
<% out.print("This is java code inside jsp page"); %>
</body>
</html>
```

JSP Tags

3.Declaration tag

1. We know that at the end a JSP page is translated into Servlet class. So when we declare a variable or method in JSP inside Declaration Tag, it means the declaration is made inside the Servlet class but outside the service(or any other) method. You can declare static member,instance variable and methods inside Declaration Tag. Syntax of Declaration Tag
2. <%! declaration %>
3. Example of Declaration Tag

```
<html>
    <body>
     <%! int a=10;int b=20;int c=30; %>
      <p>Sum of a,b,c is <%=a+b+c%></p>
 </body>
</html>
```

JSP Tags

4.Directive tag

1. Directive Tag gives special instruction to Web Container at the time of page translation.
2. Directive tags are of two types.
   - Page directive
   - Include directive
   - taglib directive

JSP Tags

1.JSP page directive

1. page directive provide attributes that gets applied to the entire JSP page. page directive has a lot of attributes that we will look now. We can define multiple attributes in a single page directive or we can have multiple page directives in a single JSP page.

JSP Tags

2.JSP include directive

1. JSP include directive is used to include the contents of another file to the current JSP page. The included file can be HTML, JSP, text files etc. Include directive is very helpful in creating templates for user views and break the pages into header, footer, sidebar sections.
2. We can include any resource in the JSP page like below.
3. <%@ include file="test.html" %>
4. The file attribute value should be the relative URI of the resource from the current JSP page.

JSP Tags

3.JSP taglib directive

1. JSP taglib directive is used to define a tag library with prefix that we can use in JSP, we will look into more details in JSP Custom Tags tutorial.
2. We can define JSP tag libraries in like below;
3. <%@ taglib uri="/WEB-INF/c.tld" prefix="c"%>
4. JSP taglib directive is used in JSP standard tag libraries, please read JSTL Tutorial.
5. Thats all for JSP directives, we will look into Expression Language (EL) and JSP Actions next.

JSP Tags

5.JSP Expression tags:

1. JSP Expression tag is used to display value of variables.
2. Calculate an expression and display its result.
3. Call method and display value returned by method.
4. The expression written between Jsp Expression tag do not end with semicolon.

Ex: <html>

<body>

<% @ page import =“java.util.*” % >

<% =Calender.getInstance().grtTime() %>

</body>

</html>

JSP V/S SERVLETS

JSP	Servlets
JSP is a webpage scripting language that can generate dynamic content.	Servlets are Java programs that are already compiled which also creates dynamic web content.
JSP run slower compared to Servlet as it takes compilation time to convert into Java Servlets.	Servlets run faster compared to JSP.
It's easier to code in JSP than in Java Servlets.	Its little much code to write here.
In MVC, jsp act as a view.	In MVC, servlet act as a controller.
JSP are generally preferred when there is not much processing of data required.	servlets are best for use when there is more processing and manipulation involved.
The advantage of JSP programming over servlets is that we can build custom tags	There is no such custom tag facility in servlets.

EXAMPLE:1

```
<html>
  <head>
   <title>How To Create a String</title>
  </head>
  <body>
      <h1>We are creating a String</h1>
       <%
       String create = "Hello world";
       %>
       <h2><%out.println(create);%></h2>
   </body>
   </html>
```

EXAMPLE:2

- Program to fetch records from database:

```
<html>
<body>
<%@ page import="java.sql.*" %>
<table><tr><td>Rollno</td>
<td>Name</td><td>City</td>
</tr>
<%
Class.forName("com.mysql.jbdc.Driver")
;
Connection
con=java.sql.DriveManager.getConnecti
on("jdbc:mysql://localhost/JAVADEMO",
"root","");
String sql="select * from
StudentMaster";
Statement s=con.createStatement();
ResultSet rs=s.executeQuery(sql);
while(rs.next())
{
%>
<tr><td><%=rs.getInt("Rollno")
%></td>
<td><%=rs.getString("StudentName")
%></td>
<td><%=rs.getString("City"); %></td>
</tr>
<%
}
rs.close();
s.close();
con.close();
%></table></body></html>
```

List of Questions:

1. What is JSP?
2. What is JSP and why do we need it?
3. What are the JSP lifecycle phases?
4. Explain how a JSP is compiled into servlet into container.
5. Explain the categories of JSP tags.
6. What are different types of comments in JSP.
7. What is JSP page?

Few Questions with answers

1.What is JSP?

- JSP is a technology for developing web pages that support dynamic content which helps developer insert java code into HTML page.

2. List the categories of JSP tags:

- JSP scripting tags
  - Scriptlets(<% scriplets %>)
  - Comments(<%-- comment --%>)
  - Expressions<%= expression %>)
  - Declarations (<%! declaration %>)
  - Directive(<%@ directive%>)

Few Question with answers

3.What are the advantage of JSP over ASP

- Advantages are as follow:
- The dynamic part is written in java,not Visual Basic or other MS specific language,so it is more powerful and easier to use.
- It is portable to other O.S and non-Microsoft Web server.

THANK YOU!