Introduction to JSP

Introduction

- It stands for **Java Server Pages**.
- It is a server side technology.
- It is used for creating web application.
- It is used to create dynamic web content.
- In this JSP tags are used to insert JAVA code into HTML pages.
- It is an advanced version of Servlet Technology.
- It is a Web based technology that helps us to create dynamic and platform-independent web pages.
- In this, Java code can be inserted in HTML/ XML pages or both.
- JSP is first converted into servlet by JSP container before processing the client's request.

JSP pages are more advantageous than Servlet:

- They are easy to maintain.
- No recompilation or redeployment is required.
- JSP has access to entire API of JAVA .
- JSP are extended version of Servlet.

Features of JSP

- **Coding in JSP is easy**: As it is just adding JAVA code to HTML/XML.
- **Reduction in the length of Code**: In JSP we use action tags, custom tags etc.
- **Connection to Database is easier**: It is easier to connect website to database and allows to read or write data easily to the database.
- **Make Interactive websites**: In this we can create dynamic web pages which helps user to interact in real time environment.
- **Portable, Powerful, flexible and easy to maintain**: as these are browser and server independent.
- **No Redeployment and No Re-Compilation**: It is dynamic, secure and platform independent so no need to re-compilation.
- **Extension to Servlet**: as it has all features of servlets, implicit objects and custom tags
    1. **Declaration Tag:** It is used to declare variables.
    2. **Java Scriplets:** It allows us to add any number of JAVA code, variables and expressions.
    3. **JSP Expression:** It evaluates and converts the expression to a string.
    4. **JAVA Comments:** It contains the text that is added for information which has to be ignored.
        - Create html page from where request will be sent to server eg try.html.
        - To handle to request of user next is to create .jsp file Eg. new.jsp

- Create project folder structure.
- Create XML file eg my.xml.
- Create WAR file.
- Start Tomcat Run Application

5. It does not require advanced knowledge of JAVA
6. It is capable of handling exceptions
7. Easy to use and learn
8. It contains tags which are easy to use and understand
9. Implicit objects are there which reduces the length of code
10. It is suitable for both JAVA and non JAVA programmer
11. Difficult to debug for errors.
12. First time access leads to wastage of time
13. Its output is HTML which lacks features.

First JSP:
JSP simply puts Java inside HTML pages. You can take any existing HTML page and change its extension to ".jsp" instead of ".html". In fact, this is the perfect exercise for your first JSP.

Take the HTML file you used in the previous exercise. change its extension from ".html" to "jsp". Now load the new file, with the ".jsp" extension, in your browser.

You will see the same output, but it will take longer! But only the first time. If you reload it again, it will load normally.

What is happening behind the scenes is that your JSP is being turned into a Java file, compiled, and loaded. This compilation only happens once, so after the first load, the file doesn't take long to load anymore. (But every time you change the JSP file, it will be recompiled again.)

Of course, it is not very useful to just write HTML pages with a .jsp extension! We now proceed to see what makes JSP so useful.

Adding dynamic content via expressions:
As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp . Of course , what makes JSP useful is the ability to embed Java. Put the following text in a file. jsp extension (let us call it hello.jsp) , place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello! The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Notice that each time you reload the page in the browser, it comes up with the current time. The character sequence.

<%= and %> enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dynamic HTML pages that change in response to user actions or vary from user to user.

Explain JSP Elements:

We will learn about the various elements available in JSP with suitable examples. In JSP elements can be divided into 4 different types.

These are:

- Expression
- Scriptlets
- Directives
- Declarations

1- Expression:

We can use this tag to output any data on the generated page. These data are automatically converted to string and printed on the output stream.

Syntax: JSP Expressions are : <%=”Anything” %>

JSP Expressions start with Syntax of JSP Scriptles are with <%=and ends with %>. Between these, you can put anything that will convert to the String and that will be displayed.

Example:

```
<%="HelloWorld!" %> Above code will display 'HelloWorld!'
```

2 – Scriplets:

In this tag we can insert any amount of valid java code and these codes are placed in the _jsp Service method by the JSP engine.

Syntax of JSP Scriptlets are:

```
<%//java codes%>
```

JSP Scriptlets begins with <% and ends %> . We can embed any amount of Java code in the JSP Scriptlets. JSP Engine places these codes in the _jspService() method.

Variables available to the JSP Scriptlets are:

- Request
- Response
- Session
- Out

3- Directives:

A JSP “directive” starts with <%@ characters. In the directives, we can import packages , and define error-handling pages or the session information of the JSP page.

Syntax
JSP directives is:

```
<%@directive attribute="value"% >
```

- page
- include
- taglib

4 – Declarations:

This tag is used for defining the functions and variables to be used in the JSP.

Syntax: JSP Declaratives are:

```
<%!
//java codes
%
```

JSP Declaratives begins with <%! and ends %> with We can embed any amount of java code in the JSP Declaratives. Variables and functions defined in the declaratives are classlevel and can be used anywhere on the JSP page.

Example:

```
<%@page import="java.util.*"%>
<HTML>
<BODY>
<%!
Date the Date=new Date(); Date getDate()
```

```
{
System.out.println("In getDate() method"); return theDate;
}
%>
Hello! The time is now<%=getDate()%>
</BODY>
</HTML
```

Explain about Jsp programs.

```
A Web Page with JSP code
<HTML>
<HEAD>
<TITLE>A Web Page</TITLE>
</HEAD>
<BODY>
<%out.println("Hello there!");%>
</BODY>
</HTML>
```

Last Updated : 20 Apr, 2023

This text is presented here in case of changed or broken links, so that the learner may still have access to the information.
https://www.geeksforgeeks.org/introduction-to-jsp/