Java Server Pages (JSP)

Java Server Pages

- JSP is a server side programming technology that enables the creation of dynamic, platform independent method for building web-based applications just like Servlet technology.
- JSP can be thought of as an extension to Servlet as it provides more functionality than Servlet.
- JSP helps in building web pages that supports dynamic content.
- JSP inserts Java code into HTML pages (JSP = HTML+JAVA)
- This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.
- Using JSP, we can collect input from users through Webpage forms, present records from a database or another source, and create Webpages dynamically.

JSP Vs Servlet

- JSP can be seen as an extension of Servlet which supports all the functionality of Servlet.
- Servlet needs to be redeployed after a modification while if JSP page is modified, there is no need to recompile and redeploy the web project.
- JSP needs less code to be written as compared to Servlet. Different types of tags such as JSTL, action, custom tag, implicit objects etc. are used.

JSP Architecture

- The JSP container (provided by Apache) is responsible for intercepting requests for JSP pages.
- A JSP container works with the Web server to provide the runtime environment and other services a JSP needs.

JSP Processing: Steps through which the JSP webpage is created by web server

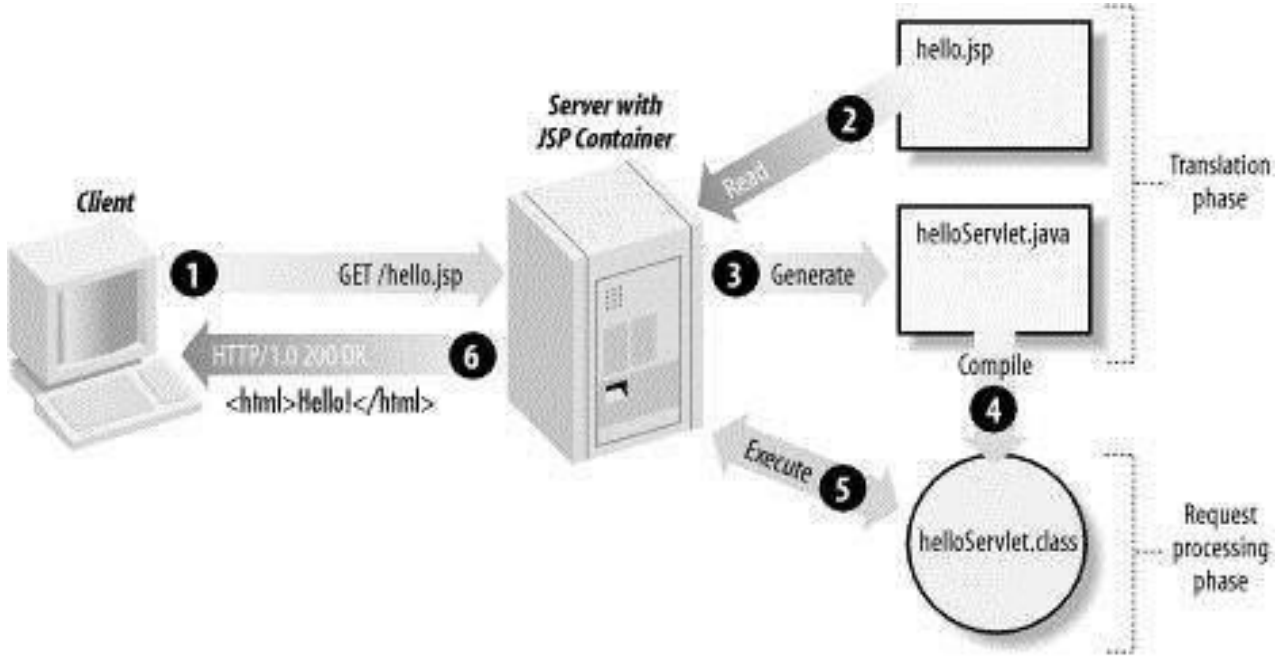

Figure: adapted from [1]

JSP Architecture cont..

- Typically, the JSP engine checks to see whether a servlet for a JSP file already exists and whether the modification date on the JSP is older than the servlet.
- If the JSP is older than its generated servlet, the JSP container assumes that the JSP hasn't changed and that the generated servlet still matches the JSP's contents.
- This makes the process more efficient than with the other scripting languages (such as PHP) and therefore faster.

JSP Lifecycle

Phases of JSP lifecycle:

1. **JSP Compilation** 2. JSP Initialization 3. JSP Execution 4. JSP Cleanup

1. JSP Compilation:

- 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.
- The compilation process involves three steps −
  - → Parsing the JSP.
  - → Turning the JSP into a servlet.
  - → Compiling the servlet

JSP Lifecycle cont..

Phases of JSP lifecycle:

1. JSP Compilation **2. JSP Initialization** 3. JSP Execution 4. JSP Cleanup

2. JSP Initialization:

- When a container loads a JSP it invokes the `jspInit()` method before servicing any requests.
- In order to perform JSP-specific initialization, override the `jspInit()` method as follows:

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

- Initialization is performed only once and as with the servlet `init` method.
- We generally initialize database connections, open files, and create lookup tables in the `jspInit` method.

JSP Lifecycle cont..

Phases of JSP lifecycle:

1. JSP Compilation 2. JSP Initialization **3. JSP Execution** 4. JSP Cleanup

3. JSP Execution:

- All interactions with the requests are represented in this phase until the JSP is destroyed.
- JSP engine invokes the `_jspService()` method after loading and initializing the JSP based on the client request.
- The `_jspService()` method takes an **HttpServletRequest** and an **HttpServletResponse** as its parameters as follows:

```
void _jspService(HttpServletRequest request, HttpServletResponse
      response) {        // Service handling code...   }
```

- This 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, i.e, GET, POST, DELETE, etc.

JSP Lifecycle cont..

Phases of JSP lifecycle:

1. JSP Compilation 2. JSP Initialization 3. JSP Execution **4. JSP Cleanup**

4. JSP Cleanup:

- The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.
- 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 as follows:

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

Scripting Elements of JSP

The Scriptlet:

- A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.

  Syntax of Scriptlet – `<% code fragment %>`

- Any text, HTML tags, or JSP elements you write must be outside the scriptlet as follows:

- The declaration of scriptlet tag is placed inside the _jspService() method.

```
<html>
        <head><title>Hello World</title></head>
        <body>
        Hello World!<br/>
        <%
        out.println("Your IP address is " +
                request.getRemoteAddr());
        %>
        </body>
</html>
```

Scripting Elements of JSP

JSP Declarations

- A declaration declares one or more variables or methods that you can use in Java code later in the JSP file. You must declare the variable or method before you use it in the JSP file.

```
<%! declaration; [ declaration; ]+ ... %>
```

- The declaration of jsp declaration tag is placed outside the `_jspService()` method.
- Following is an example for JSP Declarations :

```
<%! int i = 0; %>
<%! int a, b, c; %>
<%! Circle a = new Circle(2.0); %>
```

Scripting Elements of JSP

JSP Expression

- A JSP expression element contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file.
- Because the value of an expression is converted to a String, you can use an expression within a line of text.
- The expression element can contain any expression that is valid according to the Java Language Specification but you cannot use a semicolon to end an expression.

```
<%= expression %>
```

JSP Expression Example:

```
<html>
   <head><title>A Comment Test</title></head>
      <body>
      <p>Today's date: <%= (new java.util.Date()).toLocaleString()%></p>
   </body>
</html>
```

Today's date: 11-Sep-2010 21:24:25

Scripting Elements of JSP

JSP Comments

- JSP comment marks text or statements that the JSP container should ignore.
- A JSP comment is useful when we want to hide or "comment out", a part of your JSP page.

```
<%-- This is JSP comment --%>
```

Example:

```
<html>
   <head><title>A Comment Test</title></head>
      <body>
      <h2>A Test of Comments</h2>
      <%-- This comment will not be visible in the page source --%>
   </body>
</html>
```

```
A Test of Comments
```

Scripting Elements of JSP

There are a small number of special constructs we can use in various cases to insert comments or characters that would otherwise be treated specially shown as follows:

S.No.	Syntax & Purpose
1	**<%-- comment --%>** A JSP comment. Ignored by the JSP engine.
2	**<!-- comment -->** An HTML comment. Ignored by the browser.
3	**<\%** Represents static <% literal.
4	**%\>** Represents static %> literal.
5	\' A single quote in an attribute that uses single quotes.
6	\" A double quote in an attribute that uses double quotes.

References

- https://docs.oracle.com/cd/E17802_01/products/products/jsp/2.1/docs/jsp-2_1-pfd2/
- https://www.tutorialspoint.com/jsp/index.htm