



JSP

Introduction

Table of Contents

- JSP Introduction
- Life Cycle of JSP
- JSP API

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to servlet because it provides more functionality than servlet such as expression language, jstl etc.

A JSP page consists of HTML tags and JSP tags. The jsp pages are easier to maintain than servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tag etc.

Advantage of JSP over Servlet

There are many advantages of JSP over servlet. They are as follows:

1) Extension to Servlet

JSP technology is the extension to servlet technology. We can use all the features of servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.

2) Easy to maintain

JSP can be easily managed because we can easily separate our business logic with presentation logic. In servlet technology, we mix our business logic with the presentation logic.

3) Fast Development: No need to recompile and redeploy

If JSP page is modified, we don't need to recompile and redeploy the project. The servlet code needs to be updated and recompiled if we have to change the look and feel of the application.

4) Less code than Servlet

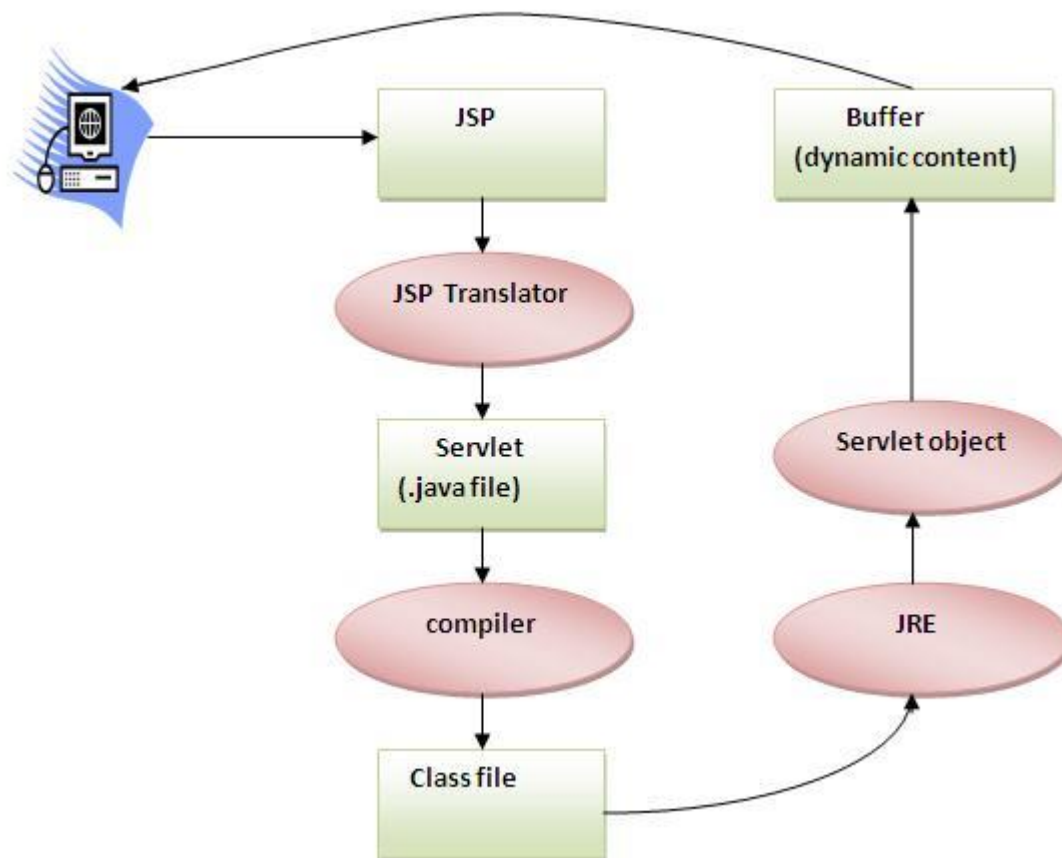
In JSP, we can use a lot of tags such as action tags, jstl, custom tags etc. that reduces the code. Moreover, we can use EL, implicit objects etc.

Life cycle of a JSP Page

The JSP pages follows these phases:

- Translation of JSP Page
- Compilation of JSP Page
- Classloading (class file is loaded by the classloader)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (`jspInit()` method is invoked by the container).
- Request processing (`_jspService()` method is invoked by the container).
- Destroy (`jspDestroy()` method is invoked by the container).

Note: `jspInit()`, `_jspService()` and `jspDestroy()` are the life cycle methods of JSP.



As depicted in the above diagram, JSP page is translated into servlet by the help of JSP translator. The JSP translator is a part of webserver that is responsible to translate the JSP page into servlet. After that Servlet page is compiled by the compiler and gets converted into the class file. Moreover, all the processes that happens in servlet is performed on JSP later like initialization, committing response to the browser and destroy.

Creating a simple JSP Page

To create the first jsp page, write some html code as given below, and save it by .jsp extension. We have save this file as index.jsp. Put it in a folder and paste the folder in the web-apps directory in apache tomcat to run the jsp page.

index.jsp

Let's see the simple example of JSP, here we are using the scriptlet tag to put java code in the JSP page. We will learn scriptlet tag later.

1. `<html>`
2. `<body>`
3. `<% out.print(2*5); %>`
4. `</body>`
5. `</html>`

It will print **10** on the browser.

How to run a simple JSP Page ?

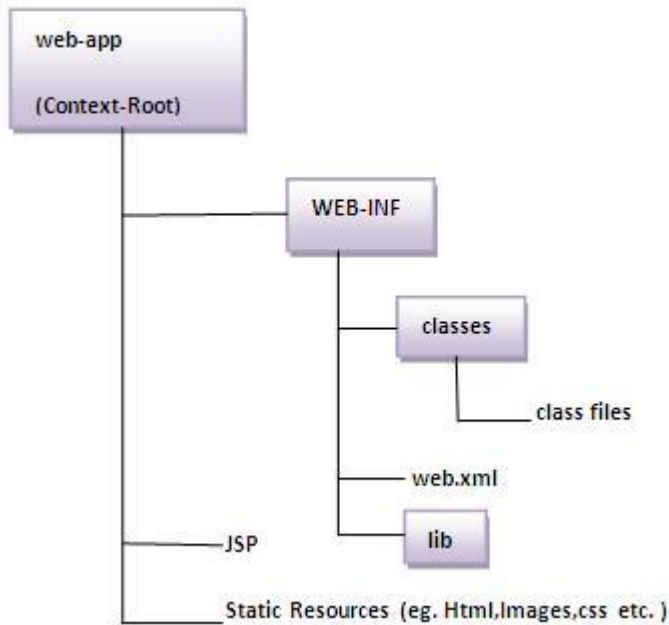
Follow the following steps to execute this JSP page:

- Start the server
- put the jsp file in a folder and deploy on the server

- visit the browser by the url `http://localhost:portno/contextRoot/jspfile` e.g.
`http://localhost:8888/myapplication/index.jsp`
-

Directory structure of JSP

The directory structure of JSP page is same as servlet. We contains the jsp page outside the WEB-INF folder or in any directory.



JSP API

The JSP API consists of two packages:

1. `javax.servlet.jsp`
2. `javax.servlet.jsp.tagext`

javax.servlet.jsp package

The `javax.servlet.jsp` package has two interfaces and classes. The two interfaces are as follows:

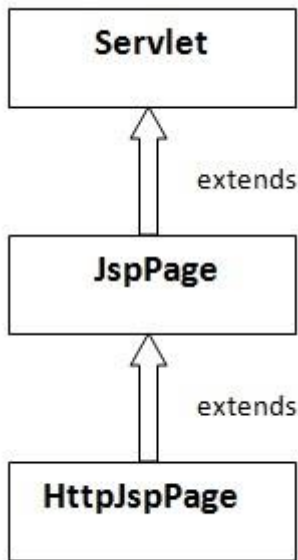
1. `JspPage`
2. `HttpJspPage`

The classes are as follows:

- `JspWriter`
 - `PageContext`
 - `JspFactory`
 - `JspEngineInfo`
 - `JspException`
 - `JspError`
-

The JspPage interface

According to the JSP specification, all the generated servlet classes must implement the JspPage interface. It extends the Servlet interface. It provides two life cycle methods.



Methods of JspPage interface

1. **public void jspInit():** It is invoked only once during the life cycle of the JSP when JSP page is requested firstly. It is used to perform initialization. It is same as the `init()` method of Servlet interface.
2. **public void jspDestroy():** It is invoked only once during the life cycle of the JSP before the JSP page is destroyed. It can be used to perform some clean up operation.

The HttpJspPage interface

The HttpJspPage interface provides the one life cycle method of JSP. It extends the JspPage interface.

Method of HttpJspPage interface:

1. **public void _jspService():** It is invoked each time when request for the JSP page comes to the container. It is used to process the request. The underscore `_` signifies that you cannot override this method.

We will learn all other classes and interfaces later.