JAVA SERVER PAGES – AN OVERVIEW

Nikita Pahuja1, Rashmi Dewan, Shivangi Kukreja
Computer Science&Engineering, Maharshi Dayanand University
Gurgaon , Haryana, India

***Abstract*- The intention of this paper is to provide an overview on the subject of ADVANCED JAVA. The overview includes previous and existing concepts, current technologies. This paper also covers various concept of java server pages (JSP). Through this paper we are creating awareness among the people about this rising field of JAVA. This paper also offers a comprehensive number of references for each concept of JSP.**

***Index Terms*- JAVA, Survey, JAVA Server Pages, JSP**

I. INTRODUCTION

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.

A. Advantage Of JSP Over Servlet

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

- **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.

- **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.

- **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.

- **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.

B. 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).

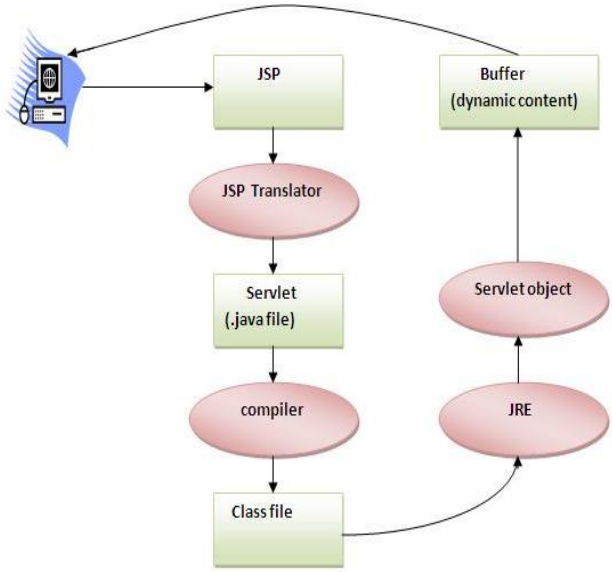

FIG 1

C. *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

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

FIG 2

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

D. *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

E. *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.

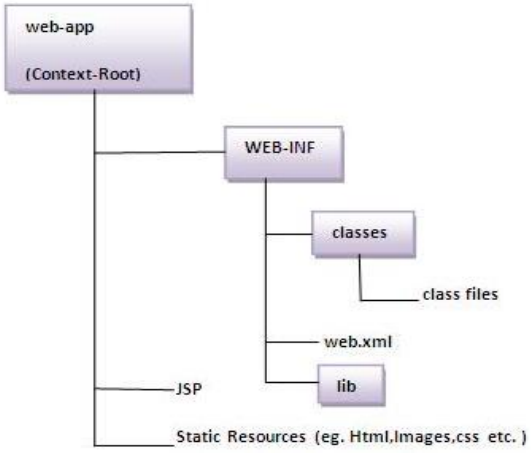

FIG 3

II. THE JSP API

The JSP API consists of two packages:

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

A. *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

B. *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.

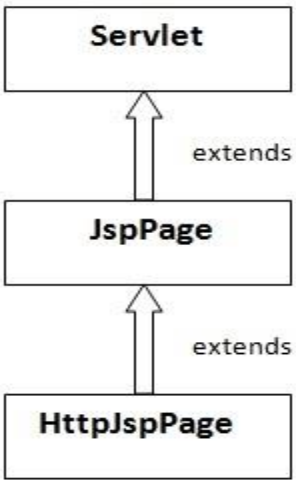

FIG 4

C. *Methods of JspPage interface*

- **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.
- **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.

III. JSP SCRIPTLET TAG (SCRIPTING ELEMENTS)

In JSP, java code can be written inside the jsp page using the scriptlet tag. Let's see what are the scripting elements first.

Scripting elements

The scripting elements provides the ability to insert java code inside the jsp. There are three types of scripting elements:

- scriptlet tag
- expression tag
- declaration tag

A. *Example Of Scriptlet*

In this example, we are displaying a welcome message

```
<html>
<body>
<% out.print("welcome to jsp"); %>
</body>
</html>
```

FIG 5

IV. JSP IMPLICIT OBJECT

There are **9 jsp implicit objects**. These objects are *created by the web container* that are available to all the jsp pages.The available implicit objects are out, request, config, session, application etc.

A list of the 9 implicit objects is given below:

Object	Type
out	JspWriter
request	HttpServletRequest
response	HttpServletResponse
config	ServletConfig
application	ServletContext
session	HttpSession
pageContext	PageContext
page	Object
exception	Throwable

FIG 6

V. EXCEPTION HANDLING IN JSP

The exception is normally an object that is thrown at runtime. Exception Handling is the process to handle the runtime errors. There may occur exception any time in your web application. So handling exceptions is a safer side for the web developer. In JSP, there are two ways to perform exception handling:

- By **errorPage** and **isErrorPage** attributes of page directive
- By **<error-page>** element in web.xml file

Example of exception handling

In this case, you must define and create a page to handle the exceptions, as in the error.jsp page. The pages where may occur exception, define the errorPage attribute of page directive, as in the process.jsp page.

There are 3 files:

- index.jsp for input values
- process.jsp for dividing the two numbers and displaying the result
- error.jsp for handling the exception

INDEX.JSP

```
<form action="process.jsp">
No1:<input type="text" name="n1" /><br/><br/>
No1:<input type="text" name="n2" /><br/><br/>
<input type="submit" value="divide"/>
</form>
```

FIG 7

PROCESS.JSP

```
<%@ page errorPage="error.jsp" %>
<%

String num1=request.getParameter("n1");
String num2=request.getParameter("n2");

int a=Integer.parseInt(num1);
int b=Integer.parseInt(num2);
int c=a/b;
out.print("division of numbers is: "+c);

%>
```

FIG 8

ERROR.JSP

```
<%@ page isErrorPage="true" %>

<h3>Sorry an exception occured!</h3>

Exception is: <%= exception %>
```

FIG 9

VI. JSP ACTION TAGS

There are many JSP action tags or elements. Each tag is used to perform some specific tasks. The action tags basically are used to control the flow between pages and to use Java Bean. Jsp action tags are as follows:

- jsp:forward
- jsp:include
- jsp:useBean
- jsp:setProperty
- jsp:getProperty
- jsp:plugin
- jsp:param
- jsp:fallback

A. jsp:forward action tag

The jsp:forward action tag is used to forward the request to another resource it may be jsp, html or another resource.

B. jsp:include action tag

The **jsp:include action tag** is used to include the content of another resource it may be jsp, html or servlet.
The jsp include action tag includes the resource at request time so it is **better for dynamic pages** because there might be changes in future.

C. jsp:useBean action tag

The jsp:useBean action tag is used to locate or instantiate a bean class. If bean object of the Bean class is already created, it doesn't create the bean depending on the scope. But if object of bean is not created, it instantiates the bean.

D. jsp:setProperty and jsp:getProperty action tags

The setProperty and getProperty action tags are used for developing web application with Java Bean. In web devlopment, bean class is mostly used because it is a reusable software component that represents data.
The jsp:setProperty action tag sets a property value or values in a bean using the setter method

VII. MVC IN JSP

MVC stands for Model View and Controller. It is a **design pattern** that separates the business logic, presentation logic and data.
Controller acts as an interface between View and Model. Controller intercepts all the incoming requests.
Model represents the state of the application i.e. data. It can also have business logic.
View represents the presentaion i.e. UI(User Interface).

A. Advantage Of MVC (Model 2) Architecture

- Navigation Control is centralized
- Easy to maintain the large application

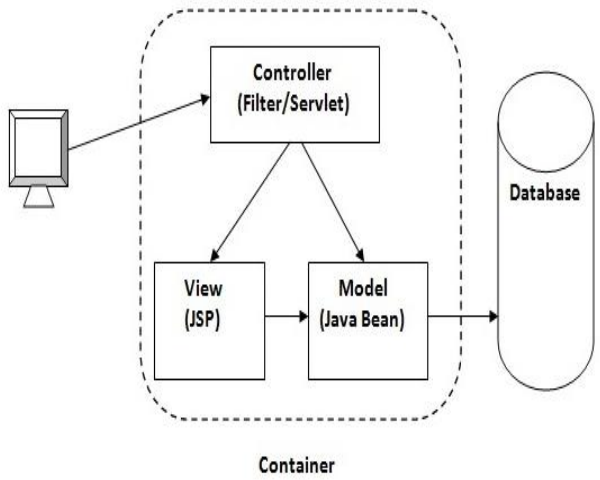

FIG 10

SUMMARY

JavaServer Pages (**JSP**) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP, but it uses the Java programming language.

To deploy and run JavaServer Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified.

JSP can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.

JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, use Java bytecode rather than a native software format. Like any other Java program, they must be executed within a Java virtual machine (JVM) that integrates with the server's host operating system to provide an abstract platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well.

The Web container creates JSP implicit objects like pageContext, servletContext, session, request & response.

DISCLOSURE STATEMENT

There is no financial support for this research work from the funding agency.

ACKNOWLEDGMENTS

We thank our guide for his timely help, giving outstanding ideas and encouragement to finish this research work successfully.

SIDE BAR

Comparison: it is an act of assessment or evaluation of things side by side in order to see to what extent they are similar or different. It is used to bring out similarities or differences between two things of same type mostly to discover essential features or meaning either scientifically or otherwise.

Content: The amount of things contained in something. Things written or spoken in a book, an article, a programme, a speech, etc.

DEFINTION

- COMPONENT - a part or element of a larger whole, especially a part of a machine or vehicle.
- CONTAINER - an object for holding or transporting something.
- ARCHITECTURE - the art or practice of designing and constructing buildings.
- DEVELOPING - grow or cause to grow and become more mature, advanced, or elaborate.

REFERENCES

[1] Mailing list archive: "Sun JSP 1.0 *not* available"

[2] The Life Cycle of a JSP Page (Sun documentation)

[3] Understanding JavaServer Pages Model 2 architecture (JavaWorld)

[4] Forum thread (JavaRanch): OutputStream already obtained

[5] JSP 1.1 Syntax Reference

[6] The Unified Expression Language (Sun Developer Network)

RELATED REFERNCES

[1] Tag Libraries Tutorial - What is a Tag Library? (Sun)

[2] JSTL documentation (Oracle)

[3] IBM WebSphere 6.0.2 documentation

[4] Sybase EAServer 5.0 documentation

[5] The Problems with JSP (January 25, 2000)