JSP and Servlets in Java

Aabha Sharma, Nikhita Upreti & Divya Bali

(aabha6@gmail.com), (nikhita.upreti@gmail.com), (divyabali16@gmail.com)

Abstract:

The paper presents an approach for statically reasoning about the behavior of Web applications that are developed using Java Servlets and JSP.it basically depicts the use of java server pages as well as servlets in building a web application in j2ee.We will be covering variety of topics in this paper including:

- **Introduction to jsp and servlets.**
- **Life cycle and architecture of servlet and jsp.**
- **Working of both**
- **Difference between jsp and servlets.**

Keywords: API , jdbc ,taglib ,CGI, http, EJB.

1. Introduction:

1.1 JSP:

JavaServer Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. JSP is a Java view technology running on the server machine which allows you to write template text in (the client side languages like HTML, CSS, JavaScript and so on). JSP supports taglibs, which are backed by pieces of Java code that let you control the page flow or output dynamically.

1.2 SERVLETS:

The servlet is a Java programming language class used to extend the capabilities of a server. Although servlets can respond to any types of requests, they are commonly used to extend the applications hosted by web servers, so they can be thought of as Java applets that run on servers instead of in web browsers. Servlet is basically a Java application programming interface (API) running on the server machine, which intercepts requests made by the client and generates/sends a response. A well known example is the HttpServlet which provides methods to hook on HTTP requests using the popular HTTP methods such as GET and POST.

2. Advantages of JSP and Servlets over CGI

2.1 WHY JSP?

JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files.

- JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc.

- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

2.2 WHY Servlets?

- Performance is significantly better.

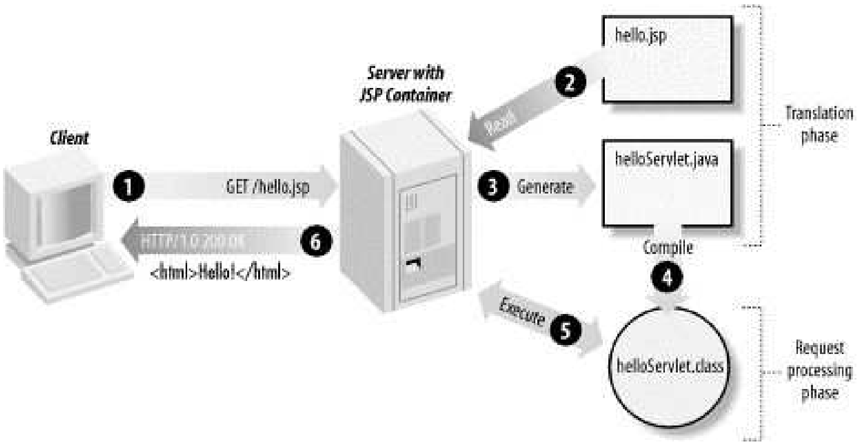
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.

- Servlets are platform-independent because they are written in Java. Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.

- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

3. JSP Architecture and Life Cycle

3.1 JSP Architecture:



The following steps explain how the web server creates the web page using JSP:

- As with a normal page, your browser sends an HTTP request to the web server.

- The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine. This is done by using the URL or JSP page which ends with **.jsp** instead of .html.

- The JSP engine loads the JSP page from disk and converts it into a servlet content. This conversion is very simple in which all template text is converted to println( ) statements and all JSP elements are converted to Java code that implements the corresponding dynamic behavior of the page.

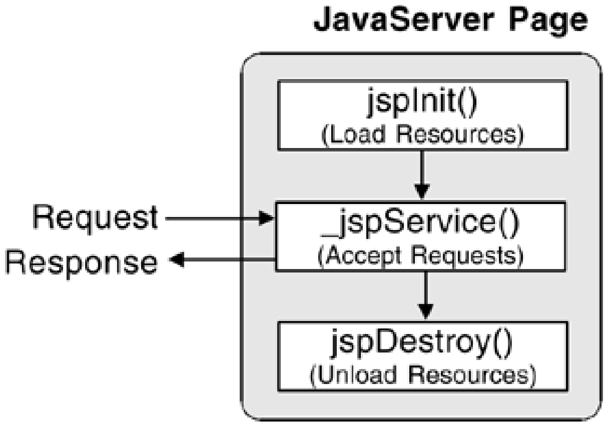
- The JSP engine compiles the servlet into an executable class and forwards the original request to a servlet engine.

- A part of the web server called the servlet engine loads the Servlet class and executes it. During execution, the servlet produces an output in HTML format, which the servlet engine passes to the web server inside an HTTP response.

- The web server forwards the HTTP response to your browser in terms of static HTML content.

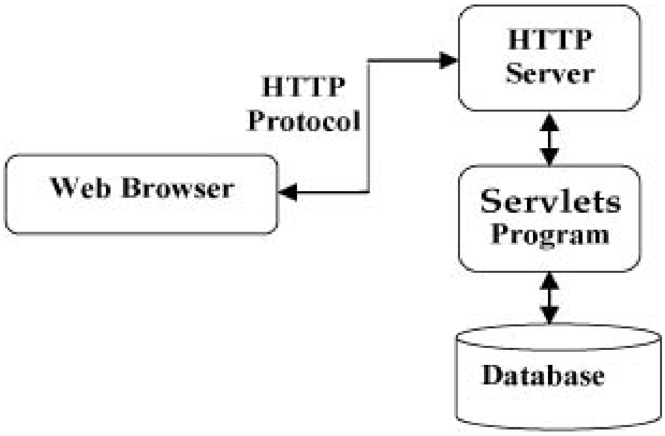
- Finally web browser handles the dynamically generated HTML page inside the HTTP response exactly as if it were a static page.

3.2 JSP Life Cycle: JSP lifecycle is consists of jspInit(), jspService() and jspDestroy(), and these methods are called life cycle methods of the JSP.



4. Servlet Architecture and lifecycle

4.1 Servlet Architecture:

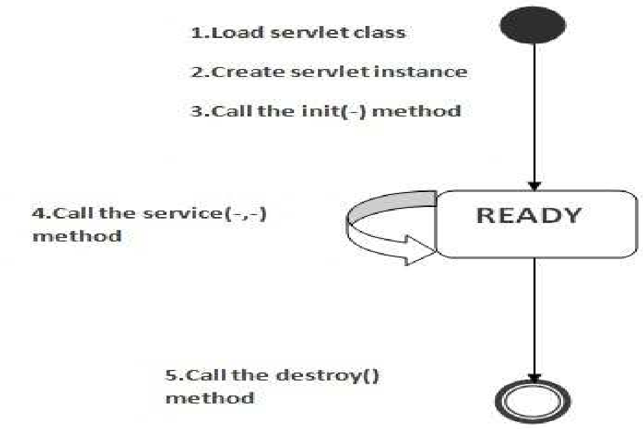


Servlets perform the following major tasks:

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing invoking a Web service, or computing the response directly.
- Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
- Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

4.2 Life Cycle of Servlet: The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.
4. service method is invoked.
5. destroy method is invoked.


1.Load servlet class
2.Create servlet instance
3.Call the init(-) method
READY
4.Call the service(-,-) method
5.Call the destroy() method

5.JSP vs Servlet

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 which can directly call Java beans.	There is no such custom tag facility in servlets.
We can achieve functionality of JSP at client side by running JavaScript at client side.	There are no such methods for servlets.

Conclusion:

In this paper we have taken into consideration all aspects of java server pages as well as servlets, discussed about their architectures and life cycles and their working. Comparison between the two technologies has also been made towards the end of the paper. On comparing and contrasting the two we find out that main difference between them is, In servlets both the presentation and business logic are place it together. Where as in jsp both are separated by defining by java beans .In jsp's the overall code is modulated so the developer who doesn't know about java can write jsp pages by simply knowing the additional tags and class names. One more important to be considered is servlet take less time to compile. Jsp is a tool to simplify the process and make the process automate. Both are web applications used to produce web content that mean dynamic web pages. Both are used to take the requests and service the clients.

References:

[1] http://www.tutorialspoint.com/jsp/

[2] http://users.polytech.unice.fr/~buffa/cours/internet/POLYS/servlets/Servlet-Tutorial-Overview.html

[3] http://www.tutorialspoint.com/servlets/

[4] http://en.wikipedia.org/wiki/Java_Servlet

[5] http://stackoverflow.com/questions/2095397/what-is-the-difference-between-jsf-servlet-and-jsp

[6] http://www.javatpoint.com/life-cycle-of-a-servlet

[7]http://www.allinterview.com/showanswers/10953.html