DOI: https://doi.org/10.53555/nncse.v1i4.517 **Publication URL: http://nnpub.org/index.php/CSE/article/view/517**

A REVIEW ON JAVA SERVLETS VS. JAVA APPLETS

Rincy Jacob[1]*, Vinayak Bhardwaj[2]
*[*1,2] Student, Information Technology Engineering, Maharishi Dayanand University Gurgaon, Haryana, India*
*[*1]rincy.14478@ggnindia.dronacharya.info, [2]vinayak.14510@ggnindia.dronacharya.info*

***Corresponding Author: -**
Email ID- rincy.14478@ggnindia.dronacharya.info

Abstract: -

Servlet technology is one of the leading web development technologies in the recent years due to its safety, portability, efficiency and elegance in design. They are platform and protocol-independent server side components, which run as part of a network service and dynamically extend Javaenabled servers. An applet on the other hand, is a Java program that can be embedded in an HTML page and executed by a Java enabled browser. It can be used for sharing resource data on enterprise intranets and for web page enhancements. Servlets used along with applets can be used for dynamic web content generation. The paper tries to analyze Java Servlets and Java Applets, their capabilities and limitations.

INTRODUCTION

Applets are Java programs that can be embedded within a HTML file. It is not only used for web page enhancements but they also allow local validation of data entered by the user by using database to perform list of values lookups and data validation, and have complex GUI widgets. Servlets specialize in browsing the web (web documents and various databases) for information extraction, data mining and meaningful presentation inside the web browser. More important, their power is in filtering, online analysis and structuring of the retrieved information.

1. JAVA SERVLETS

Java code that runs as part of a network service, typically an HTTP server and responds to requests from clients is called Java Servlet. The **servlet** is a Java programming language class used to extend the capabilities of a server. For example, a client may need information from a database; a servlet can be written that receives the request, gets and processes the data as needed by the client and then returns the result to the client. Servlets can also be used to provide dynamic web content such as responses to database queries, store and process data that was obtained from an HTML form and manage state information which does not exist in stateless http. Since servlets run inside servers, they require a graphical user interface. They can also be described as the server side counterpart to applets: they are Java applications which are downloaded, on demand, to the part of the system which needs them. To deploy and run a servlet, a web container is used. A web container (also known as a servlet container) is the component of a web server which interacts with the servlets. The web container is responsible for the mapping a URL to a particular servlet, managing lifecycle of servlets, and ensuring that the URL requester has the correct access rights.

Servlets often are provided by organizations which provide customized multi-user services to their customer bases. However, servlets are extensible enough to support standardized services such as proxying services and serving static web pages using HTTP protocols. They may also be used in a plug-in style to support dynamic extensibility and for use in semi-custom applications and search engines.

1.1. SERVLET USAGE

- Servlets play an important role in enterprise networks by connecting with a database using JDBC connectivity
- Servlets provide with protocol support, meaning it allows a file service using NFS protocol to migrate to any other protocol and make the transfer transparent to the user.
- Another common usage of servlet is to accept, store and process data provided by a user in a html form
- A group of servlets can work as active agents for sharing data
- Servlets can handle multiple requests concurrently which can be synchronized together for collaborative applications such as video conferencing
- A single servlet forwards request to all the servers reducing the load on several servers containing the same data.

1.2. SERVLET LIFE CYCLE

Servlets support a familiar programming model of accepting requests and generating responses. This model is used with a various distributed system programming toolsets, ranging from HTTP requests made to web servers to remote procedure calls. Servlets are implemented using the servlet interface, generic or http specific. They encapsulate the data sent by a client, providing parameter access and allowing servlets to report the status.

Servlets are dynamically loaded, although servers provide with an administrative option to force loading and initializing the servlet when the server starts up. This allows for greater flexibility in system architecture with easier distribution of network services. Three methods central to the life cycle of a servlet are init(), service(), and destroy(). They are implemented by every servlet and are invoked at specific times by the server. These are:

- **Init () or initialization of servlet**: When a server loads a servlet, the web container0 runs the servlet's init method allowing it to access the name-value parameters from the web application.
- **Servicing of client request**: An HTTP servlet handles client requests using its service method. Each request is serviced in its own separate thread. The service method dispatches each request to a method designed to handle that request. If a request is made for a method that is not implemented by the servlet, the method of the parent class is called, typically resulting in an error being returned to the requester. This allows servlets to handle multiple clients simultaneously.
- **Destroying the servlet**: When the web container destroys a servlet, the server runs the servlet's destroy method. The method is run once in the lifetime of a servlet. The server won’t run the servlet again until it is reloaded and reinitialized.

1.3. COMPARISON WITH CGI

CGI has existed for a long time and they are universally accepted and stable. The major advantages of servlets over CGI applications are:

- Platform independence: CGI programs are platform-dependent. Servlets are based on Java classes which need to written only once and can be implemented anywhere. Therefore, they are truly platforms independent.
- Network Programming: Servlets have full access to Java's networking features. The servlets connect with other networked computers by using socket or Remote Method Invocation (RMI). Servlets can easily connect to a database using the Java Database Connection (JDBC). They are also used to develop middleware.

- Performance: Servlets perform substantially better over CGI counterpart. Each CGI request results in the creation of a new process. The overhead associated with it dominates the workload especially when fast operating speeds are required. On the other hand, a servlet continues to run in the background after servicing a request. Also, CGI programs are not threaded. Servlets can use multi-threading to process multiple requests simultaneously provided the JVM supports threading.
- State Information: CGI programs are stateless since they result in the creation of a new process each time a request is serviced. A servlet keeps the memory of its state once it has been loaded by the server. The JVM loads the servlet when it is called. The servlet isn't reloaded until it changes. Also, the servlet on being modified may be reloaded without restarting the server. Maintaining such information allows servlets to share information.
- Restrictive Environment: A servlet is run within servlet container in a restrictive environment, called a sandbox. Similar to an applet which runs in the sandbox of a web browser. This prevents any harm to the user system due to potentially harmful servlets.

2. JAVA APPLETS

An applet is a java program that can be embedded in a HTML file. When you use Java enabled browser to view an applet containing webpage, the applet's code is send to your system and executed by a browser.

A Java applet can appear in the frame of the web page, Sun's AppletViewer, a new application window, or a stand-alone tool for testing applets. Java applets were first introduced in 1995. They can be written in any programming language that can compile Java bytecode. Usually written in Java, but other languages such as Pascal, JRuby, Scala etc. Web browsers implement strict restrictions on what the applet can do in order to prevent access violations of the user's system. A Java applet extends the class java.applet.Applet, or in the case of a Swing applet, javax.swing.JApplet.

2.1 LIFE CYCLE OF AN APPLET

When you load a web page that contains an applet, it goes through several stages. During each of these stages, the applet performs several tasks, although most of these tasks are invisible to the enduser. One of the three steps in processing an applet is first initialized then it is run and finally it is exited.

During the initialization, the applet is loads the sound clips, images and other resources that it needs to run. The initialization stage is over, when the applet has all the resources it needs to run. Thus the applet is ready to run.

While running it performs all the tasks it was designed to do. On the other hand, the applet remains idle when the web page containing it has not been accessed. Since the applets work on when the web page has been opened, they have two different stages of stop and start which have their own cycles. The starting and stopping stages of the applet can also be manipulated.

During it final exiting stage, the JVM performs some garbage collection ensuring that all the memory used by the applet has been freed and the applet has been destroyed. This is an important stage as the applet is downloaded to the user's system and uses its memory and CPU.

The applet package used has methods corresponding to each of the stages on an applet. Mostly these methods are automatically invoked in an applet and the user does not need to particularly define them everytime unless something particular needs to be performed.

The first method invoked when an applet is loaded is init(), this allows the applet to load everything necessary for its working. For example, an applet that plays an audio clip at the click of a button. In such an applet, the audio clip can be pre-loaded and the button can be setup before the applet began to run.

After the applet has been initialized, the start () method is automatically invoked. This method allows the applet to perform all its tasks. You can make the applet to play music, video, draw images or accept input from user.

The stop () method is called automatically when an applet should stop execution orwhen you leave an applet's web page. You don't need to redefine need to redefine the method everytime you use it. If you have a simple application you can let the applet invoke this method itself but if thread are involved then its good practice to implement the stop method according to your desire.

Finally, the destroy () method is invoked. This method clears the memory of any traces of the applet and literally destroys it.

3. JAVA SERVLETS vs. JAVA APPLETS

Servlets have been named after applets but run using a JVM or HTML browser. Applets and servlets allow the server and client to be extended in a modular way by dynamically loading code which communicates with the main program via a standard programming interface. A servlet can also be thought of as a server side applet. Servlets run the same way on a server as an applets run on a browser. Servers can directly access servlets as applets can be directly accessed by a browser.

- Servlets do not require a graphical user interface since it runs on the server and isn't directly viewed by the user. They are free of the security restrictions implemented on applets. They are trusted programs since they are used at the server side Java application components which are downloaded on demand by the part of the system which needs it. To determine which is one is apt for which situation largely depends on the client-server architecture which cater to thin clients and have fat servers these days. Some situations where servlets are better:
- If most of the computation for the web page can be done on the server side, it is pointless to load the part of the code that does the computation to the client. The computation can be done on the server and passed back to the client.
- Applet from the Internet to the client system and executed in a JVM on the client. If large amount of code is used with slow internet speeds, applets are not the appropriate choice.

- If the process cannot be performed within the restrictions of an applet, then servlets are better Scenarios where applets are more appropriate:
- Applications that require a graphical user interface should be implemented on an applets
- If internet speeds are good then applets can be easily implemented and the overhead involved can be ignored
- Processing can also be split between the two, applets and servlets

CONCLUSION

Servlet technology is robust and scalable as it uses the java language. It allows the user to create complex sever side applications. It can be used to replace CGI applications, communicate with remote computers and access databases. Applet technology on the other hand is promising but developers lack in controlling the final output using it. Both are highly powerful methods in Java and gives the developers various operations. The universal client today is the HTML browser. Tomorrow it will be the HTML/JavaScript browser and servlets and applets play a major role.

REFERENCES

[1].ELE548 Research Essay: Topic 3 Java Applets vs. Java Servlets by Zhengqiang Shan, April 12, 1999 at http://www.ele.uri.edu/Courses/ele548/sp99/shan_top3.pdf
[2].http://www.sys-con.com/java/feature/3-1/servletsfriends/
[3].http://www.sys-con.com/java/feature/3-1/cgi-scripts/index.html
[4].http://www.servletcentral.com/common/articlelist.dchtml
[5].http://webreview.com/wr/pub/97/10/10/feature/colton.html
[6].http://www.developer.com/news/techfocus/021698 servlet.html
[7].http://www.sys-con.com/java/feature/3-2/3-tier/index.html
[8].http://www.servletcentral.com/
[9].http://javaboutique.internet.com/
[10]. http://www.javasoft.com/applets/index.html
[11]. http://www.javasoft.com/features/1997/oct/applets.html
[12]. http://www.javasoft.com/features/1997/dec/applets2.html
[13]. http://www.javasoft.com/features/1998/07/applet.power.iii.html
[14]. http://en.wikipedia.org/wiki/Java_Servlet
[15]. http://en.wikipedia.org/wiki/Java_applet
[16]. chipkidz.wordpress.com/2009/08/05/servlet-api
[17]. www.itu.dk/~sestoft/javaservlets/servlets/api.html
[18]. http://dictionary.reference.com/browse/java+servlet
[19]. http://stackoverflow.com/questions/6632712/how-httpservlet-can-create-threads-withoutimplementing-runnable-or-extending-th