© 2014 IJIRT | Volume 1 Issue 6 | ISSN : 2349-6002

JAVA SERVLETS

Akshay Raina, Arushi Kohli
DCE,Gurgaon

***Abstact*- The ascent of server-side Java applications is one of the most recent and most energizing patterns in Java programming. Java servlets are a key segment of server-side Java improvement. A servlet is a little, pluggable expansion to a server that improves the server's usefulness. Servlets permit engineers to amplify and tweak any Java-empowered server—a web server, a mail server, an application server, or any custom server—with a heretofore obscure level of versatility, adaptability, and straightforwardness. This paper additionally incorporates the life cycle of servlet. The applet –servlet cooperation is depicted by two systems, Http Connection and Socket Connection. Finally it incorporates different systems for server debugging.**

I. INTRODUCTION

The servlet is a Java programming language class used to Expand the capacities of a server. In spite of the fact that servlets can react to any sorts of requests, they are ordinarily used to broaden the applications hosted by web servers, so they can be considered Java Applets that run on servers rather than in web browsers [1]. These kinds of servlets are the Java counterpart to other dynamic web content Technologies such as PHP and Asp.net. Technically Speaking, a "servlet" is a Java class in Java EE that complies with the Java Servlet API,[2] a standard for Actualizing Java classes which react to demands. Servlets could on a basic level impart over any client– server convention, yet they are frequently utilized with the HTTP convention. In this manner "servlet" is regularly utilized as shorthand for "HTTP servlet".[3] Thus, a product engineer may utilize a servlet to add element substance to a web server utilizing the Java stage. The created substance is usually HTML, at the same time may be other information, for example, XML. Servlets can keep up state in session variables crosswise over numerous server transactions by utilizing HTTP treats, or URL revamping.

Servlets are regularly used

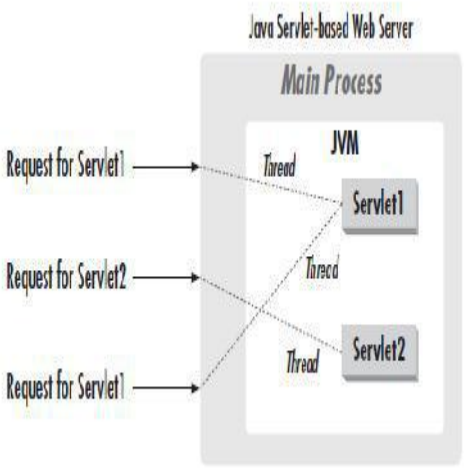
- Process or store data that was submitted from an HTML form
- Provide dynamic content such as the results of a database query
- Manage state information that does not exist in the stateless HTTP protocol, such as filling the articles into the shopping cart of the appropriate customer .

To install and run a servlet, a web compartment must be utilized. A web container (otherwise called a servlet compartment) is basically the segment of a web container that cooperates with the servlets. The web container is in charge of dealing with the lifecycle of servlets, mapping a URL to a specific servlet and guaranteeing that the URL requester has the right get to rights. The Servlet API, contained in the Java package hierarchy of "javax.servlet", characterizes the normal cooperations of the web container and a servlet.[3]

A Servlet is an object that gets a request and produces a Response focused around that request. The fundamental Servlet package characterizes Java objects to represent to servlet requests and reactions, and objects to reflect the servlet's design parameters and execution environment. The package "javax.servlet.http" characterizes HTTP-particular subclasses of the generic servlet components, including session management objects that track different requests and reactions between the web server and a client. Servlets may be packaged in a WAR document as a web application.

Servlets can be generated consequently from Java Server Pages (JSP) by the Javaserver Pages compiler. The contrast in between of servlets and JSP is that servlets regularly embed HTML inside Java code, while JSPs embed Java code in HTML. While the immediate utilization of servlets to produce HTML (as indicated in the case beneath) has got to be uncommon, the more elevated amount MVC web framework in Java EE (JSF) still unequivocally utilizes the servlet engineering for the low level request/reaction taking care of by means of the Facesservlet. As was said prior, a servlet is a generic server extension— a Java class that can be stacked dynamically to grow the usefulness of a server. Servlets are normally utilized with web servers, where they can take the spot of CGI scripts. A servlet is like an exclusive server extension, aside from that it runs inside a Java Virtual Machine (JVM) on the server, so it is sheltered and versatile. Servlets work singularly inside the domain of the server: not at all like applets, they don't oblige help for Java in the web browser.

Dissimilar to CGI and FastCGI, which utilize various processes to handle separate programs and/or separate requests, servlets are all taken care of by particular threads inside the web server process. This implies that servlets are likewise productive and adaptable. Since servlets run inside the web server, they can cooperate nearly with the server to do things that are unrealistic with CGI scripts. An alternate preference of servlets is that they are versatile: both crosswise over operating systems as we are utilized to with Java furthermore crosswise over web servers. In spite of the fact that servlets are most generally utilized as a substitution for CGI scripts on a web server, they can extend any kind of server.

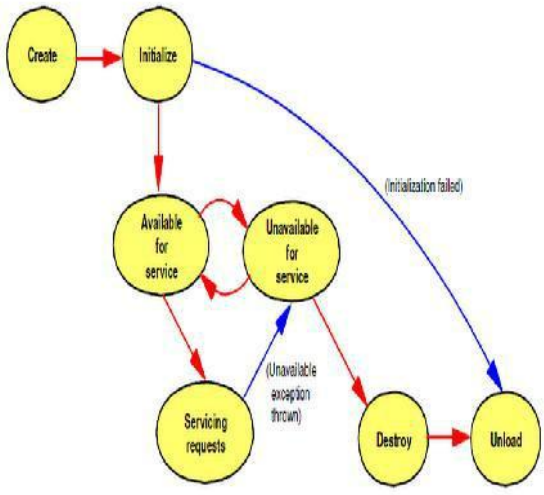


Servlets offer various focal points over different methodologies:

- Portability: Because servlets are composed in Java and comply with a decently characterized and generally acknowledged API, they are profoundly convenient crosswise over operating systems and crosswise over server executions. You can create a servlet on a Windows NT machine running the Java Web Server and later send it easily on a top of the line UNIX server running Apache. With servlets, you can genuinely "compose once, serve all around."
- Power: Servlets can load the full power of the center Java Apis: systems administration and URL access, multithreading, picture control, information clamping, database network, internationalization, remote system conjuring (RMI), CORBA integration, and article serialization, among others. On the off chance that you need to compose a web application that permits workers to question a corporate legacy database, you can exploit the majority of the Java Enterprise Apis in doing so.
- Efficiency and Endurance: Servlet summon is exceedingly effective. When a servlet is stacked, it by and large stays in the server's memory as a solitary article occurrence. From that point, the server summons the servlet to handle a request utilizing a straightforward, lightweight system summon. Servlets, as a rule, are commonly continuing objects. Since a servlet stays in the server's memory as a solitary article case, it naturally keeps up its state and can clutch outer assets.
- Safety: Servlets help safe programming practices on various levels. Since they are written in Java, servlets inherit the solid sort safety of the Java dialect. Moreover, the Servlet API is actualized to be sort safe. Servlets can deal with lapses securely, because of Java's exemption taking care of system.
- Integration: Servlets are hard incorporated with the server. This integration permits a servlet to participate with the server in ways that a CGI program can't.
- Extensibility and Flexibility: The Servlet API is intended to be effectively extensible. The way things are today, the API incorporates classes that are streamlined for HTTP servlets. Servlets are also quite flexible.

II. LIFE CYCLE OF SERVLETS

The servlet life cycle is a standout amongst the most energizing gimmicks of Servlets. The servlet life cycle permits servlet engines to address both the execution and asset issues of CGI and the security concerns of low-level server API programming. A servlet motor may execute all its servlets in a solitary Java virtual machine (JVM) . Since they are in the same JVM, servlets can proficiently impart data to one another, yet they are kept by the Java language from getting to each other's private data. Servlets might likewise be permitted to hold on between requests as item instances, taking up far less memory than undeniable processes. Servlet life cycle is exceedingly adaptable. Servers have huge space by the way they decide to backing servlets. The main resolute principle is that a servlet motor



must comply with the accompanying life cycle contract:

1. Create and initialize the servlet.
2. Handle zero or more service calls from clients.
3. Destroy the servlet and then garbage collects it.

A Java servlet has a lifecycle that characterizes how the servlet is stacked and initialized, how it gets and reacts to

requests, and how it is taken out of administration. In code, the servlet lifecycle is characterized by the javax.servlet.servlet interface. All Java servlets must, either specifically or in a roundabout way, implement the javax.servlet.servlet interface so they can run in a servlet motor. The servlet motor is a redone extension to a Web server for preparing servlets, implicit conformance with the Java Servlet API by the Web server vendor. The servlet motor gives system administrations, comprehends MIME requests, and runs servlet containers. The javax.servlet.servlet interface characterizes strategies that are called at particular time and in a particular request amid the servlet lifecycle.

2.1 Understanding the Life Cycle

This section describes in detail some of the important servlet life-cycle methods of the Java Servlet API.

2.1.1 Servlet Initialization: init method

Servlets can be dynamically loaded and instantiated when their administrations are initially requested, or the Web server can be arranged so that particular servlets are loaded and instantiated when the Web server initializes. In either case, the init method of the servlet performs any essential servlet initialization, and is ensured to be called once for every servlet occurrence, before any requests to the servlet are taken care of. A case of an errand which may be performed in the init method is the stacking of default data parameters or database connections. The most widely recognized type of the init method of the servlet acknowledges a Servletconfig object parameter. This interface article permits the servlet to get to name/worth sets of initialization parameters that are particular to that servlet. The Servletconfig protest additionally provides for us get to the Sevletcontext question that depicts data about our servlet surroundings. Each of these objects will be examined in more detail in the servlet cases segments.

2.1.2 Servlet Request Handling

Once the servlet has been legitimately initialized, it may handle requests (despite the fact that it is conceivable that a loaded servlet may get no requests). Each one request is spoken to by a Servletrequest article and the relating reaction by a Servletresponse question in the Java Servlet API. Since we will be managing Httpservlets, we will bargain only with the more specific Httpservletrequest and Httpservletresponse objects.

The Httpservletrequest object encapsulates information about the client request, including information about the client's surroundings and any data that may have been sent from the client to the servlet. The Httpservletrequest class contains methods for concentrating this information from the request item.

The Httpservletresponse is frequently the dynamically produced reaction, case in point, a HTML page which is sent again to the client. It is frequently assembled with data from the Httpservletrequest item. Notwithstanding a HTML page, a reaction article might likewise be a HTTP mistake reaction, or a redirection to an alternate URL, servlet, or Javaserver Page. The redirection procedures will be examined in more detail in the servlet cooperation segment of this section.

Each one time a client request is made, another servlet thread is produced which benefits the request. Thusly, the server can deal with different simultaneous requests to the same servlet. For each one request, normally the administration, doget, or dopost methods will be called. These methods are passed the Httpservletrequest and Httpservletresponse parameter objects.

- dopost: Invoked at whatever point a HTTP POST request is issued through a HTML structure. The parameters connected with the POST request are imparted from the program to the server as a different HTTP request. The dopost method ought to be utilized at whatever point changes on the server will happen.
- doget: Invoked at whatever point a HTTP GET method from a URL request is issued, or a HTML structure. A HTTP GET method is the default when a URL is defined in a Web program. Rather than the dopost method, doget ought to be utilized when no changes will be made on the server, or when the parameters are not susceptible data

2.1.2 Destroy Method

The destroy method is called when the Web server empties the servlet. A subclass of Httpservlet just needs to implement this method in the event that it needs to perform cleanup operations, for example, discharging database connections or shutting documents. The server calls a servlet's destroy() method when the servlet is going to be unloaded. In the destroy() method, a servlet ought to free any assets it has obtained that won't be refuse gathered. The destroy() method additionally gives a servlet an opportunity to work out its unsaved stored information or any constant information that ought to be perused amid the following call to init(). Different methods include:

- getservletconfig: The getservletconfig method gives back a Servletconfig occasion that can be utilized to furnish a proportional payback parameters and the Servletcontext object.
- getservletinfo: The getservletinfo method is a method that can give information about the servlet, for example, its creator, form, and copyright. This method is by and large overwritten to have it give back a significant quality for your application. By default, it returns an empty string.

2.1.3 Service Method

This method is pronounced Abstract in the essential Genericservlet class, along these lines subclasses, for example, Httpservlet, must override it. In our subclass of Httpservlet, when utilizing this method, we must implement this method as per the mark characterized in Httpservlet, to be specific, that it acknowledges Httpservletrequest and Httpservletresponse contentions. We do some treatment of the reaction object, which is in charge of sending our reaction over to the client. Our reaction here is an arranged HTML. page, so we first set the reaction substance sort to content/html by coding res.setcontenttype("text/html"). Next, we request a Printwriter item to compose content to the reaction by coding Printwriter out = res.getwriter() . We could likewise have utilized a Servletoutputstream item to

work out our reaction, however getwriter provides for us more adaptability with Internationalization. In either case, the substance kind of the reaction must be set before references to these objects can be made.

2.1.5 How the Servlet gets invoked

We could invoke this servlet with either a GET or POST form activity method; the administration method will execute for either. On the off chance that we knew something about how this servlet was eventually to be called, for example, what the HTML structure method was going to be, we could have implemented the above usefulness through particular doget or dopost methods. The result would be the same. The most straightforward approach to invoke the servlet would be by defining a URL in the Web program. A URL strengths the Web program to send the request utilizing GET, like the way a standard HTML page is requested. The above servlet could be invoked from the Web program with the URL: http://host/servlet/itso.servjsp.servletapi.simplehttpservlet http://host/itsoservjsp/servlet/itso.servjsp.servletapi.simpleh ttpservlet.

III. APPLET-SERVLET INTERACTION

An applet is a component that exists and works in the client level; the servlet goes about as an extension to the Web application or a standalone segment offering administrations. Applets permit an alternate type of segment embodiment and there are numerous applets accessible for business. Whether you utilize applets, Javabeans, or Servlets rely on upon their capacity and what level you need these parts to work through. Case in point, there are applets that situated up menus and toolbars, offer an extensive variety of monetary number crunchers and show business sector costs. how about we contemplate applets that need to correspond with the server. There are various great cases. Examine the organization applet that deals with the Java Web Server. Ponder how it functions -it executes on the client, yet it designs the server. To do this, the applet and the server need to be in close steady correspondence. As an alternate sample, examine one of the well known visit applets. One client says something, and all the rest see it. How does that work? They positively don't impart applet to applet. Rather, every applet presents its messages on a focal server, and the server deals with upgrading alternate clients. At last, envision an applet that tracks the cost of a set of stocks and offers consistent upgrades. How does the applet know the current stock costs, and, all the more critically, how can it know when they change?

3.1 Communication Choices

There are numerous methods for communication, and the two most basic are Internet and intranet. An interior network has particular communication channels that it must utilization and these channels are frequently exclusive to the network operating framework. The Internet offers an extensive variety of communication protocols. the most widely recognized being FTP and HTTP. The method of communication for applets and Servlets are either HTTP or attachment.

3.1.1 HTTP Communication

Having an applet make a HTTP connection with a CGI system functions admirably thus:

- It's not difficult to compose.
- It works actually for applets running behind a firewall. Most firewalls permit HTTP connections yet forbid crude attachment connections.
- It permits a Java applet to communicate with a system written in any language. The CGI program doesn't need to be composed in Java. It can be in Perl, C, C++, or some other language.
- It meets expectations with applets composed utilizing JDK 1.0, so it meets expectations with all Java-enabled programs.
- It permits secure communication. An applet can communicate with a secure server utilizing the encoded HTTPS (HTTP + SSL) convention.
- The CGI project can be utilized by programs and additionally applets. On account of our stock tracker sample, the CGI system can do twofold obligation, additionally going about as the back-end for a HTML structure based stock quote administration.

Anyhow the HTTP connection to a CGI program likewise has a few issues:

- It's moderate. Due to the HTTP request/reaction ideal model, the applet and the CGI program can't communicate intelligently. They need to restore another communication channel for each one request and reaction. In addition, there is the standard deferral while the CGI program dispatches and initializes itself to handle a request.
- It normally obliges requests to be framed as an unbalanced show of name/quality sets.
- It compels all reactions to be formatted utilizing some discretionary, long ago settled upon standard.
- Only the applet can initiate communication. The CGI program needs to hold up inactively for the applet to request something before it can react. On the off chance that a stock value changes, the applet can discover just when it asks the right question.

3.1.1 Socket Communication

An applet and server can likewise communicate by having the applet build an attachment connection to a non-HTTP server process. This gives the accompanying favorable circumstances over the HTTP-based methodology:

- it permits bidirectional, managed communication. The applet and servlet can utilize the same attachment (or even a few attachments) to communicate intelligently, sending messages here and there and then here again. For security reasons, the applet should dependably initiate the connection by uniting with a server attachment on the server machine, however after an attachment connection has been built, either gathering can keep

in touch with the attachment whenever. This permits our stock tracker to get stock value upgrades when they are accessible.

- it permits a more effective project to run on the server side. The non-HTTP server can be composed to handle a request instantly without propelling an outside CGI project to do the work.

In any case an attachment connection likewise has inconveniences versus the HTTP-based methodology:

- It comes up short for applets running behind firewalls. Most firewalls don't permit crude attachment connections, and along these lines they forbid this kind of applet-server communication. Thusly, this instrument ought to be utilized just when an applet is ensured to never run on the furthest side of a firewall, for example, for an intranet application.
- It can be genuinely convoluted to compose the code that runs on the server. There must dependably be a few procedure, (for example, a stock quote server) listening on a well-known port on the server machine. Creating such an application in Java is less demanding than in C++, however it is still nontrivial.
- It may require the improvement of a custom convention. The applet and server need to characterize the convention they use for the communication. While this convention may be more straightforward and more productive than HTTP, it frequently must be extraordinarily created.
- The non-HTTP server can't be helpfully joined with by a web browser.

IV. SERVLET DEBUGGING

Debugging servlets can be dubious in light of the fact that you don't execute them straightforwardly. Rather, you trigger their execution by method for a HTTP request, and they are executed by the Web server. This remote execution makes it hard to embed break indicates or read debugging messages and stack follows. Along these lines, methodologies to servlet debugging contrast to some degree from those utilized as a part of general advancement. Here are some general procedures.

- Use print statements: Insert several print statements to discover which line of code generated the error and which question on that line was invalid.

- Use an integrated debugger in your IDE: Many integrated improvement situations (Ides) have sophisticated debugging devices that can be integrated with your servlet and JSP compartment.

- Use the log file: The Httpservlet class has a method called log that gives you a chance to write information into a logging file on the server. Perusing debugging messages from the log file is a bit less helpful than watching them specifically from a window as with the two past methodologies, yet utilizing the log file is an alternative actually when running on a remote server; in such a situation, print statements are seldom valuable and just the progressed Ides support remote debugging.

- Write separate classes: One of the essential standards of great programming outline is to put normally utilized code into a separate capacity or class so you don't have to continue reworking it. That guideline is much more important when you are composing servlets, since these separate classes can regularly be tried autonomously of the server.

- Plan ahead for missing or malformed data: Every time you handle data that comes straightforwardly or in a roundabout way from a client, make certain to consider the likelihood that it was entered incorrectly or overlooked altogether.

- Look at the HTML source: If the result you see in the browser looks odd, pick View Source from the browser's menu.

- Look at the request data separately: Servlets read data from the HTTP request, develop a reaction, and send it once more to the client. On the off chance that something the whole time happens, you need to find if the reason is that the client is sending the wrong data or that the servlet is handling it incorrectly.

- Stop and restart the server: Servers should keep servlets in memory between requests, not reload them each one time they are executed. In any case, most servers support an advancement mode in which servlets should be automatically reloaded at whatever point their associated class file changes. Now and again, be that as it may, a few servers can get confounded, particularly when your just change is to a lower-level class, not to the top-level servlet class. Along these lines, in the event that it creates the impression that progressions you make to your servlets are not reflected in the servlet's behavior, have a go at restarting the server.

V. CONCLUSION

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server. Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically. Servlet's 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.

REFERENCES

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

[2] Java Servlet Programming, Jason Hunter with William Crawford, published by O'Reilly, ISBN 1-56592-391-X.

[3] http://pdf.moreservlets.com/

[4] http://courses.coreservlets.com/

[5] http://www.redbooks.ibm.com/

[6] http://java.sun.com/products/servlet

[7] Dimitrova, M. (2003). Cognitive modelling and Web search: Some heuristics and insights, Journal "Cognition, Brain, Behaviour" Vol. VII, No 3, Pp. 251-258.