Table of Contents:

1. Java Servlets - Overview
2. Java Servlets
3. Servlet Life Cycle
4. Environment Setup
5. Java Servlet Example
6. HTTP Servlet
7. Interfaces of Servlet

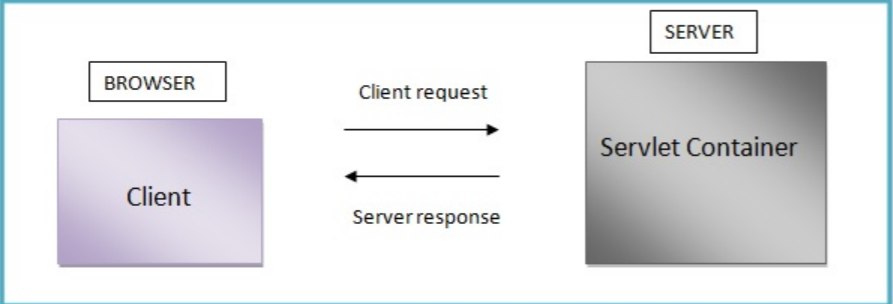
Servlets Overview

Servlet is a Java programming language class, part of Java Enterprise Edition (Java EE). Sun Microsystems developed its first version 1.0 in the year 1997. Its current Version is Servlet 3.1. Servlets are used for creating dynamic web applications in java by extending the capability of a server. It can run on any web server integrated with a Servlet container. In fact, servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. (Exelixis Media (P.C., 2015)

Before Servlets, CGI (Common Gateway Interface) programming was used to create web applications. They come up with certain disadvantages such as having a high response time because CGI programs execute in their own OS shell, not scalable not always secure or object-oriented. So they need to find a solution.By this, Sun Microsystems developed **Servlet** which serves as their solution over traditional CGI technology.

Servlets Process

When client is requesting, the request is sent to a servlet container. Servlets are run inside the servlet container. No matter how many requests is send by the client. The time the request is arrived, the web server searches for the servlet and initiates it. The request of the client will be processed by the servlet and forwarded the response back to the server. Then the Server response is then forwarded to the client.



Servlets advantages according to Exelixis Media (P.C., 2015):

-Servlets are platform independent as they can run on any platform.

- The Servlet API inherits all the features of the Java platform. It means that, Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases

- It builds and modifies the security logic for server-side extensions.

- Servlets inherit the security provided by the Web Server.

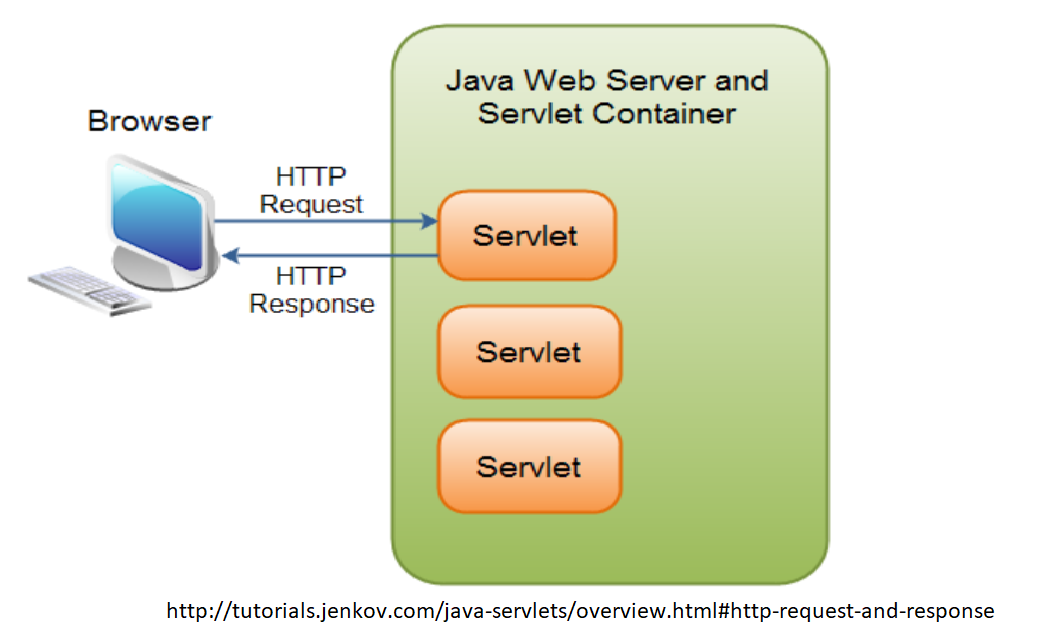
- In Servlet, only a single instance of the requests runs concurrently. It does not run in a separate process. So, it saves the memory by removing the overhead of creating a new process for each request.

- It collects input from users through web page forms, present records from a database or another source, and create web pages dynamically. It means that Servlet container may run multiple and execute web applications at the same time, each having multiple servlets running inside.

JAVA SERVLETS

Java Servlets are java object that are intended to play the role server components in client-server communications and a java class extending HTTPServlet class. It handles client requests by generating responses to such requests. A java servlets class needs to be compiled prior to using it; it must use servley-api.jar. Java Servlets are part of the Java Enterprise Edition (Java EE). It is run inside a Servlet compatible "Servlet Container" (e.g. web server) in order for them to work such as Jelly, Apache Tomcat that are free Java web servers.

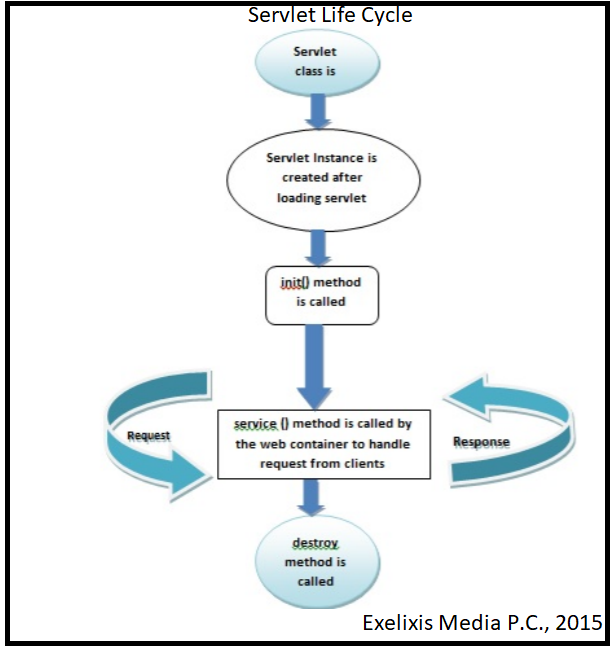
Servlets is hosted in a ‘servlet container’ that provides the environment in which the server run as the ‘servlet life cycle’.



Servlet Container is known as servlet engine which manages Java Servlet components on top of a web server to the request send by the client.

Servlet Life Cycle

Servlet lifecycle describes how the servlet container manages the servlet object.



The servlet lifecycle:

First phase: Instantiation

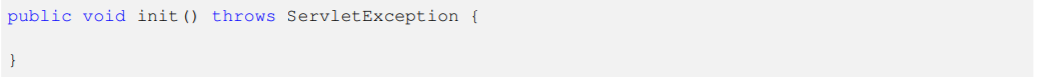
- ‘constructor all…’

- Creates an instance of the servlet to service client request

-Invoked implicitly by the servlet container when the servlet is called then it creates instance to create servlet.

Second phase: Initialization

- ‘init() method’ : This is called only once when the servlet is created.

- Invoked once and is intended for any startup initialization code for the servlet.

Third phase: Request Handling

- ‘service method’: It is called by the web container to handle request from client

-Invoked repeatedly for every client request

-Servlet performs whatever logic necessary in order to serve the request and response.

-Multi-client request typically handled by single multi-threaded

- Must have ‘thread safe’

Fourth phase: Destruction

- ‘destroy method’ : It is used to clean resources and called before removing the servlet instance.

- Invoked before the servlet instance is ‘unloaded’ and is intended for ‘housekeeping’.



Environment Setup

### Creating First Servlet Application using Netbeans IDE

We will setup the Servlet first in using Integrated Development Environment(IDE) on NetBeans. Developers take it as the easiest way to create Servlet Applications.

Java Servlet Demo

HTTP Servlet

Java Servlet use to handle HTTP request and generate HTTP response.It is hosted in ‘web-container’ which is component in Java ‘application-server’. Application server like WildFly, GlassFish, Apache Tomcat, IBM WebSphere and more.

What happens on the Request handling on the third phase of the servlets life cycle. The service() method call is routed to a d0XXX() call, depending on the HTTP request method , e.g. doGet(), doPost(), doHead(), doPut(), doDelete().

|  |  |
| --- | --- |
| doGet () | if the servlet supports HTTP GET requests |
| doPost() | for HTTP POST requests |
| doPut() | for HTTP PUT requests |
| doDelete() | for HTTP DELETE requests |
| Init()and destroy() method | to manage resources that are held for the life of the servlet |
| getServletInfo() | which the servlet uses to provide information about itself |

The doXXX() methods passed two arguments: the object representation of the HTTP request sent by the client and received by servlet. It is use to accessed information from the request message (e.g. HTTP request, request URI, query string, message readers, message payload).

Interfaces of Servlets

In reference with the Apache tomcat documentation on the HttpServlet, HttpServlet provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site.

HttpServletRequest

-getAuthType() ,getContextPath() ,getHeaders(), getPathInfo() ,getMethod() ,getRequestURI() ,etc.

HttpServletResponse

- containsHeader(java.lang.String name) ,addIntHeader(java.lang.String name, int value) ,setStatus(int sc) ,setHeader(java.lang.String name, java.lang.String value) HttpSessionActivationListener]

HttpSessionAttributeListener

HttpSessionBindingListener

HttpSessionContext

HttpSessionListener