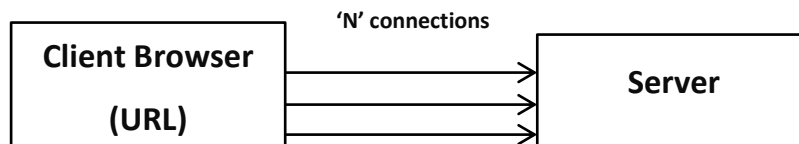


Servlet

- ❖ **Goal of Servlet:** creating dynamic web pages
- ❖ **Servlet:** It is a Java class that runs on a web server and handles requests from clients, such as browsers or other applications.
- ❖ A servlet can *process data, perform business logic, and generate responses*, usually in the form *of HTML, XML, JSON, or other formats*.
- ❖ A servlet is not a standalone program, but a component that is managed by a servlet container, such as **Tomcat, Jetty, or GlassFish**.
- ❖ The servlet container is *responsible for loading, initializing, executing, and destroying servlets, as well as providing them with services such as security, concurrency, and communication*.
- ❖ **Difference between servlet and applet:**

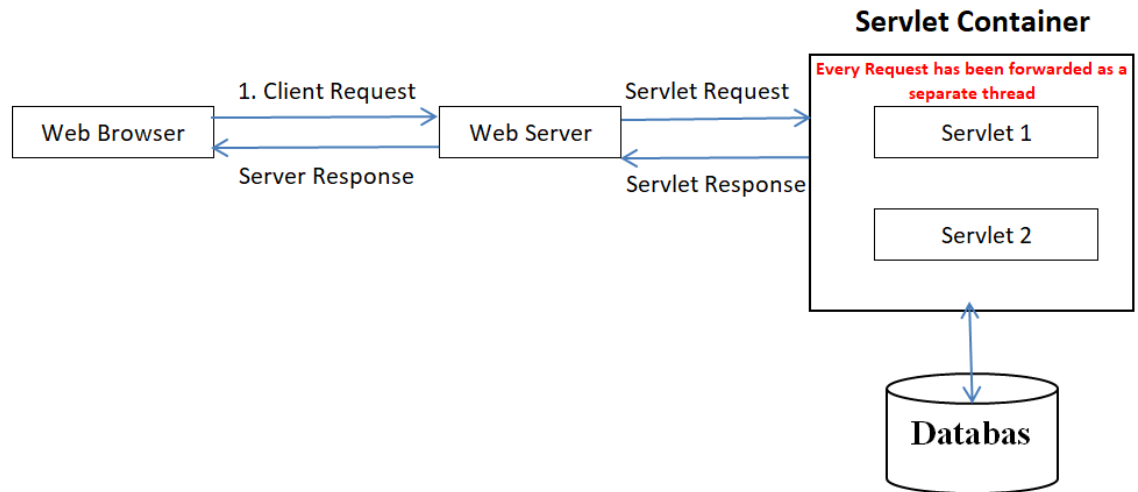
S. No.	Servlet	App--let
1.	It is a small program that execute on the server side of a web connection.	It is a small program that execute on the client side of a web connection.
2.	Dynamically it extends the <u>functionality of web server</u> .	Dynamically it extends the <u>functionality of web browser</u> .

- ❖ **Need for Servlet:**



- Need separate connection to handle each client request
 - Web server dynamically constructs a web page to handle each request.
 - Terminate the connection when the response sent to it.
- ❖ **Advantages of Servlet:**
 1. Performance is better (Because servlet executes within the address space of a web server) – Thread based
 2. Platform independent
 3. It provides more security for accessing the resources.
- ❖ **Working principle of Servlet:**
 1. The client sends the request to the web server (enter the URL in the browser)
 2. Web server receives the client request and forwards the request into the web service container. (respective servlet)
 3. The servlet processes the request and generates the response in the form of output

4. The servlet sends the response back to the web server
5. Web server sends the response back to the client and client browser display the output it on the screen.



❖ Life Cycle of Servlet:

1. `public void init(ServletConfig)`
2. `public void service(ServletRequest, ServletResponse)`
3. `public void destroy()`

❖ Servlet API

Two packages contain the classes and interfaces that are required to build servlets.

1. **`javax.servlet.*`**; - it provides framework to work on servlet [It provides the interfaces and classes which is used to create a servlet framework.

Interfaces

1. Servlet
2. Servletconfig
3. ServletContext
4. ServletRequest
5. ServletResponse

Classes:

1. GenericServlet
2. ServletInputStream
3. ServletOutputStream
4. ServletException
5. UnavailableException

2. **`javax.servlet.http.*`**; - it works with Http Request and Response [It provides the functionality to build a servlet to work with HTTP request and response.

Interfaces:

1. HttpServletRequest
2. HttpServletResponse
3. HttpSession
4. HttpSessionBindingListener

Classes:

1. Cookie – allows to store a state information on a client machine
2. HttpServlet – provides a method to handle HTTP request and response
3. HttpSessionEvent – Session maintenance purpose
4. HttpSessionBindingEvent – Session mapping

Servlet Config Interface

- ❖ **ServletConfig Interface:** Servers use ServletConfig objects to pass initialization and context information to servlets. The initialization information generally consists of a series of initialization parameters (init parameters) and a ServletContext object, which provides information about the server environment.
- ❖ For every Servlet class in our application, the web container will create one ServletConfig object and the web container will pass this object as an argument to the public void init(ServletConfig config) method of our Servlet class object.
- ❖ **Properties of ServletConfig:**
 1. It is an object containing some initial parameters or configuration information created by the Servlet Container and passed to the servlet during initialization.
 2. It is for a particular servlet, which means one should store servlet-specific information in web.xml and retrieve it using this object.
- ❖ **Methods in ServletConfig Interface:** (abstract methods)
 1. getServletName()
 2. getServletContext()
 3. getInitParameter(String)
 4. getInitParameterNames()

ServletContext

- ❖ it is used to get configuration information from the web.xml file. The ServletContext object can be used to set, get or remove attribute from the web.xml file.
- ❖ **It is created by the web container at the time of deploying object.**
- ❖ Only one servletContext object is created per web application.
- ❖ *If any information is shared to many servlet, it is better to provide it from the web.xml file using the <context-param> element.* (sub element of <web-app>)
- ❖ **Methods:**
 1. getInitParameter(String name)
 2. getInitParameterNames()
 3. setAttribute(String name, Object object)

4. `getAttribute(String name)`
5. `getInitParameterNames()`
6. `removeAttribute(String name)`

❖ **Example:** Accessing context-param infor

```
ServletContext context=getServletContext();  
String driverName=context.getInitParameter("dname");  
pw.println("driver name is="+driverName);
```

Example web.xml

```
<web-app>  
    <servlet>  
        <servlet-name>sonoojaiswal</servlet-name>  
        <servlet-class>DemoServlet</servlet-class>  
    </servlet>  
    <context-param>  
        <param-name>dname</param-name>  
        <param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>  
    </context-param>  
    <servlet-mapping>  
        <servlet-name>sonoojaiswal</servlet-name>  
        <url-pattern>/context</url-pattern>  
    </servlet-mapping>  
</web-app>
```