

UNIT - 11 :-

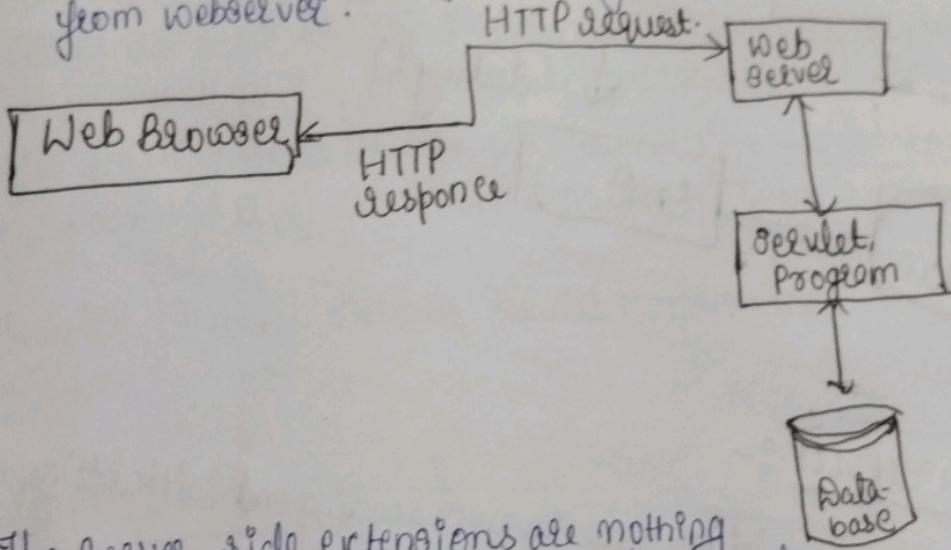
SERVLET

J

- Servlets are the Java programs that runs on the Java-enabled web server or application server.
- They are used to handle the request obtained from webserver, process the request, produce the response, then send back to webserver.

→ Properties :-

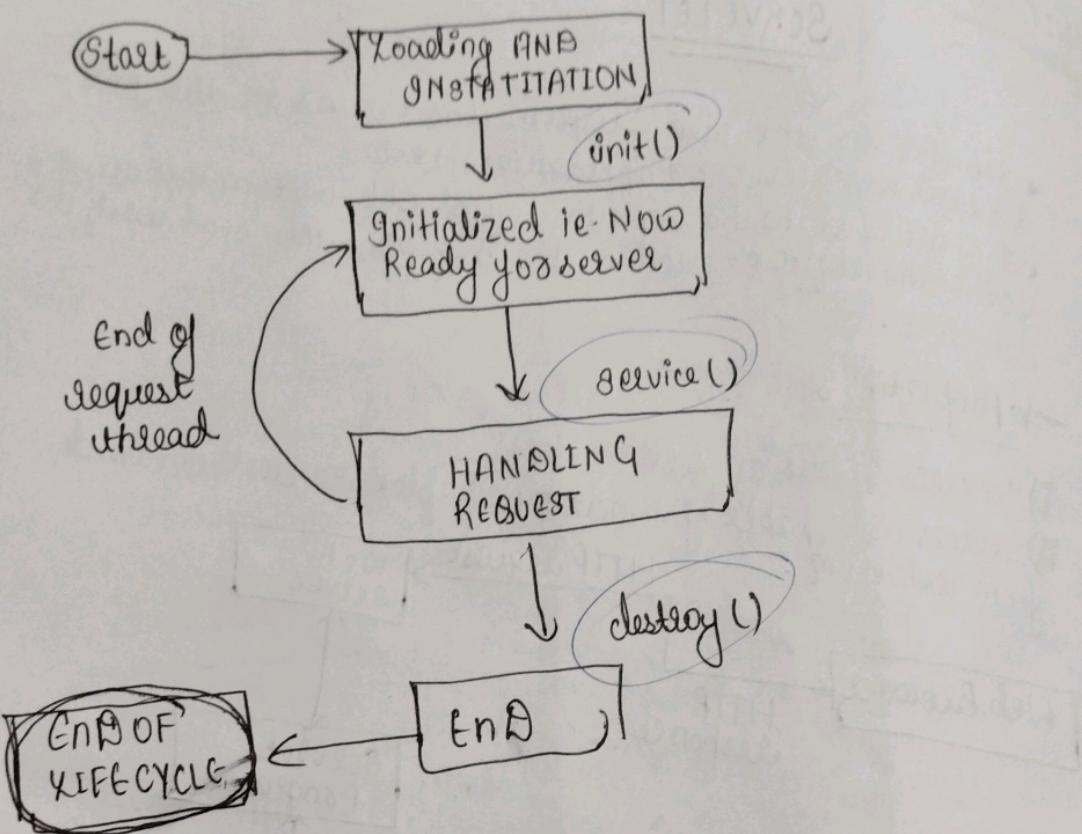
- i) Servlets works on server-side.
- ii) Servlets are capable of handling complex request obtained from webserver.



- The Server-side extensions are nothing but the technologies that are used to create dynamic web-pages.

→ Life Cycle of a Servlet :-

Basically life cycle of a servlet means how the servlet container manages the servlet obj.



LOADING A SERVLET :-

The first stage of the Servlet lifecycle involves loading and initializing the servlet by the servlet container.

The servlet container performs two operations in this stage:-

1) Loading :- Load the servlet class.

2) Instantiation :- Create an instance of a servlet. To create an instance of the servlet, the container uses the no-argument constructor.

ii) Initializing a servlet :-

- After the servlet is instantiated & loaded, the container initializes the loaded servlet successfully.
- The container initializes the servlet object by invoking the servlet.init (servlet config) method which accepts servletConfig object reference as parameter.
- If the servlet fails to initialize it will throw the ServletException or UnavailableException.

iii) HANDLING A REQUEST

After the initialization of servlet instance it is ready to serve the client requests.

- Firstly it creates the ServletRequest & ServletResponse objects. The web creates HttpServletRequest & HttpServletResponse objects.
- After creating these things it invokes the servlet.service method by passing request & response object.

→ service () may throw ServletException or UnavailableException or IOException.

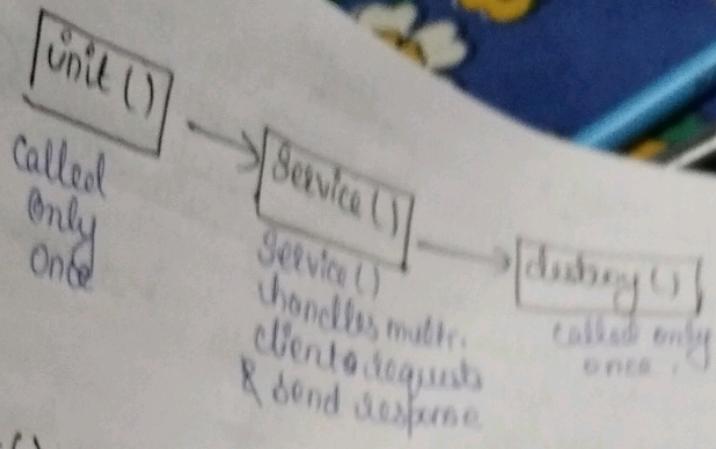
iv) DESTROYING A servlet :-

To destroy the servlet it before :-

- i) It allows all the running threads of servlet instance to complete their job & get released.
- ii) After this, the servlet container calls the destroy() method on servlet instance.

SERVLET LIFE CYCLE METHOD :-

- i) init()
- ii) service()
- iii) destroy()



i) unit() method:

The ServletInit() method is called by the Servlet container to indicate that this Servlet instance is initialized successfully.
 public class MyServlet implements Serializable
 public void init(ServletConfig config) throws ServletException { }

ii) Service() method:

- The service() method of the Servlet is invoked to serve the Servlet abt client req.
- ServletRequest object - to collect data requested by client.
- ServletResponse object - to generate the output for requested data.

iii) destroy() method:

~~This get run only once during life cy. of S.~~ to end the Servlet instance.

// destroy() method
 public void destroy()

→ Servlet life cy. can be defined as the stages through which a Servlet passes from its creation to destruction.

- Servlet is borned
- - - - initialized
- - - - lead to service
- - - - Serving

- - not used to service.
- servlet to destroy

Q: Difference b/w Servlet & CGI.

Ans \Rightarrow Servlet :- Servlet are the java program that runs on Java enabled web servers and applications.

CGI :- Common Gateway Interface provides the middle ware b/w WWW servers & database & info-sys.

Difference :-

SERVLET

- i) It is thread based
i.e. for every new request
new thread is created.
- ii) codes written in Java.
- iii) It is portable.
- iv) It can be used on any of
the webserver.
- v) Data sharing possible
- vi) It can read & set HTTP
headers.
- vii) It can be platform
dependent.

CGI (Common Gateway Interface)

- ii) It is processed based i.e.
for every new request new
process is created.
- iii) Any programming language.
- iv) It can use the web-sites
that support it.
- v) Not poss.
- vi) It can neither read nor set.
- vii) Platform independent.

Q: Explain the Servlet lifecycle with diagram & explain its
methods in details.

Q: Recall the working of init() method of a Servlet.

Q. What is ServletConfig & ServletContext, its advantages. Diff. b/w them with help of program — 10marks.

Ans → SERVLETCONFIG :-

- ServletConfig is an object containing some initial parameters or configuration info. created by ServletContainer & passed to the Servlet during initialization.
- It is for a particular Servlet, that means one should store obj. of Servlet specific info in web.xml and retrieve them using this Servlet inside web.xml can be retrieved using getServletConfig(). getInitParameter(" ") in the Servlet.

ServletContent :-

- ServletContent is the object created by ServletContainer to share initial parameters or configuration information to the whole application.
- ServletContent is achieve the website name using — getServletContent.getInitParameter(" ") ...

→ Methods in ServletConfig interface :-

- i) public abstract java.lang.String getServletName()
- ii) public --- javax.servlet.ServletContext ---
- iii) public --- java.lang.String getInitParameter(java.lang.String)
- iv) --- java.util.Enumeration<java.lang.String> getInitParameterNames()

→ Advantages :-

- i) The core advantage of ServletConfig is that we don't need to edit the servlet file if info. is modified from the web.xml file.

→ Advantage of servlet context :-

- Easy to maintain if any info is shared to all the servlet.
It is better to make it available for all the servlet.

Difference b/w them :-

ServletConfig

i) ServletConfig is servlet specific parameters of ServletConfig are present as name-value pair in <init-param> inside <servlet>.

ii) SC object is obtained by getServletConfig() method.

iii) Each servlet has got its own ServletConfig object.

iv) Use SC when only one servlet needs info shared by it.

CODE :- (ServletConfig)

```
import package com.jspbook;  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;
```

```
public class InternationalizedHelloWorld extends HttpServlet {  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
        IOException, ServletException {
```

ServletContext

i) ServletContext is for whole application parameters of ServletContext are present as name-value pair in <context-param> which is outside <servlet> and inside <web-app>

ii) SC obj. is obtained by getServletContext() method.

iii) SC obj. is only one and used by diff. servlets of the appli.

iv) Use ServletContext when whole application needs of info shared by it.

```
response.setContentType("HTML");
PrintWriterOut = response.getWriter();
out.println("<HTML>");
out.println("<head>");

String greeting;
greeting = getServletConfig().getInitParameter("greeting");
out.println("<title>" + greeting + "</title>");
out.println("<head>");
```

```
out.println("<body>");
```

```
out.println("<h1>" + greeting + "</h1>");
```

```
out.println("</body>");
```

```
out.println("</html>");
```

```
}
```

→ **doGet()** shall be used when small amount of data and insensitive data like a query has to be sent as a request.

→ **doPost()** shall be used when comparatively large amount of sensitive data has to be sent.

→ GENERIC SERVLET :-

- GenericServlet implements the servlet interface and provides an implementation for all its methods except the service() method hence it is abstract.
- Gen-S. class implements servlet, servletconfig & serialization interfaces.
- GSC can handle any type of request so it is protocol-independent.
- We may create a generic servlet by inheriting the GenericServlet class & providing the implementation of the service method.

→ HTTP Servlet :-

- HttpServlet is an abstract class, it comes under package 'java.servlet.http.HttpServlet'.
- To create a servlet the class must extend the HttpServlet class and override at least one of its methods (doGet, doPost, doDelete, doPut).
- The HSC class extends the GenericServlet class and implements a Serializable interface.

→ Methods of HttpServlet Class :-

- i) doGet() Method - This method is used to handle the GET request on the server-side.
 - The get type request is usually used to process a request.

Parameters - Request, Response.
Exceptions - IOException, ServletException.

- ii) doPost() Method - This method is used to handle the POST request on the server-side.
 - The post request is usually used to post-process a request.

→ game.

^{syntax}
iii) doHead() Method

- This method is overridden to handle the Head request.

- This method is used to improve performance.

→ p 41

iv) doPut() Method - This method is overridden to handle the PUT request.

- This method is called by the browser.

→ p 41

→ p 41

→ MOVING SERVLET TO HTTP :-

Q. Differentiate GenericServlet and HttpServlet.

Ans → GENERICSERVLET

HTTP SERVLET

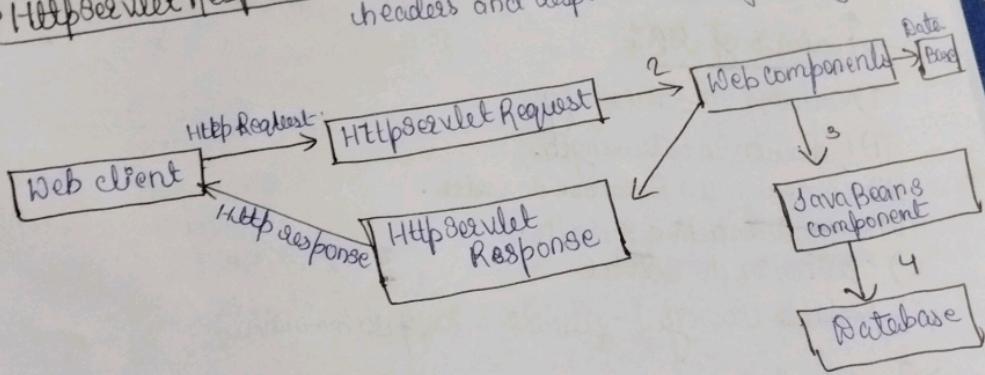
- i) It is defined by javax.servlet.http package.
- ii) It describes protocol-independent servlet.
- iii) The service method is abstract.
- iv) It is an immediate child class of GenericServlet class.
- v) GenericServlet is a superclass of HttpServlet class.

→ HttpServlet is a subclass of GenericServlet class.

LINKING SERVLET TO HTML :-

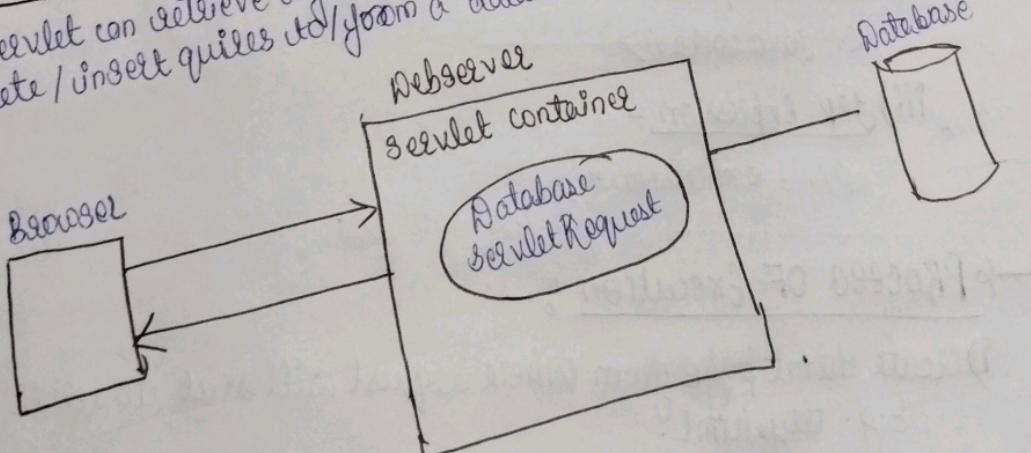
→ HttpServlet Request - It can be used to retrieve incoming Http request headers and form data.

→ HttpServlet Response - It can be used to set the HTTP response headers and response message body.



SERVLET WITH JDBC :-

A servlet can retrieve information from a database or perform updates, delete / insert quires it/ from a database.



JSP

- Java Server pages.
- It is a server side technology.
- It is used for creating web application.
- It is used to create dynamic web content.
- In this Java code can be inserted in HTML / XML.

→ Features of JSP :

- i) Coding in JSP is easy.
- ii) Reduction in code length.
- iii) Connection to Database is easier.
- iv) Make interactive Website.
- v) Extension to Servlet.
- vi) Portable, Powerful, flexible & easy to maintain.

→ Syntax :

i) Declaration Tag :-

`<%! Decl val %>`

ii) Java Scriptlets -

`<% Java code %>`

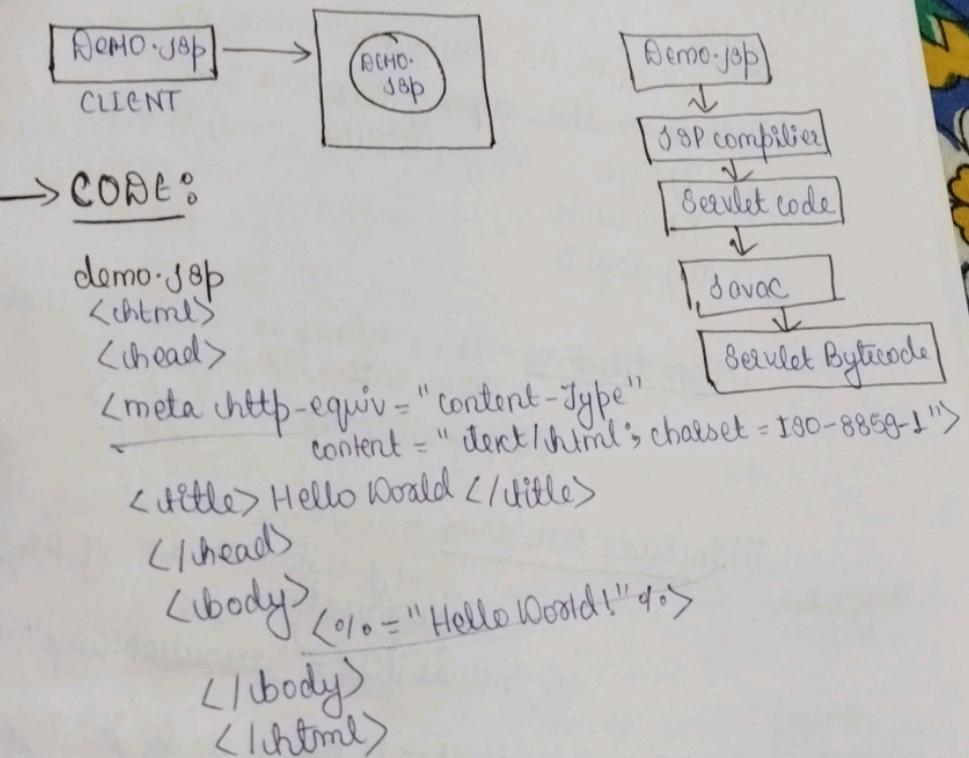
iii) JSP Expression -

`<% = expression %>`

→ PROCESS OF Execution :-

- i) Create html page from where request will sent to server.
E.g. they.html
- ii) To handle the request of user next is to create .jsp file.
- iii) Create project folder structure.

- iv) Create XML file.
- v) Create WAR file.
- vi) Start Tomcat.
- vii) Run Application.



ADVANTAGE :-

- i) It is capable of handling exceptions.
- ii) Easy to use & learn.
- iii) It can tags which is easy to use & understand.
- iv) It is suitable for both JAVA & non JAVA programmer.

DISADV. :-

- i) Difficult to debug errors.
- ii) First time access leads to wastage of time.
- iii) Its output is HTML which lacks features.

→ JSP Directives :

JSP directives are the elements of a JSP source code that guide the web container on how to translate the JSP page into its respective servlet.

Syntax -

<%@ directive attribute = "value" %>

There are three types of directives :-

i) page <%@pageattribute = "value" %>

ii) include

iii) taglib

i) Page directives - It is defined as attributes that supply to an entire JSP page.

<%@ pageattribute = "value" %>

ii) Include Directives - It is used to include the content of any resource it may be jsp file, html, text.

<%@ include file = "resourceName" %>

iii) Taglib Directives - It is used to define tag library that defines many tags.

<%@ taglib uri = "uri of the taglibrary" prefix = "prefix of taglibrary" %>

→ JSP Scriptlets :-

Scriptlets are used to write the Java code inside JSP page.

- A Scriptlets tag is used to execute Java source code in JSP.

Syntax -

<% java source code %>

E.g. :-

```
<html>
<body>
    <% out.println("Pikachu"); %>
</body>
</html>
```

→ JSP Include Tag :-

- The JSP include action tag is used to include the content of another resource it may be JSP, HTML, text file.

- The JSP include action tag includes the resources at request time so it is better for dynamic pages because there might be changes in future.

- It can be used to include static as well as dynamic pages.

→ AVN :-

code reusability.

Syntax -

<jsp:include page = "relative URL | <% = expression %>">

E.g. :-

```
<h2> this is ... </h2>
<jsp:include page = "point date.jsp" %>
    <h2> ... </h2>
    <h2> ... </h2>
```

→ JSP page tag :-

→ JSTL :-

- JSP standard tag library.
- It aims to provide an easy way to maintain JSP pages.

E.g.:-

```
<c:set var = "bookname">  
My book : java 8  
</c:set>  
<c:out value = "${bookname}" />
```

Q. Explain various Scriptlets tags in JSP with the help of program.
Ans:- Scriptlets tags are used to write the Java code inside JSP page.

i) Scriptlet tag - This tag allows user to insert Java code in JSP.

E.g.:-

```
<html>  
<body>  
<% out.print("Welcome to jsp"); %>  
</body>  
</html>
```

ii) JSP Expression Tag - It is used to print the values of variable or method.

E.g.:-

```
<html>  
<body>  
<% = "welcome to jsp" %>  
</body>  
</html>
```

```
index.jsp  
<html>  
<body>  
Current Time : <%= java.util.Calendar.getInstance().getTime() %>  
</body>  
</html>
```

→ JSP Declaration Tag : It is used to declare fields & methods.
• The code written inside the JSP declaration tag is placed outside the service() method of auto generated servlet.

Syntax -
<%! field or method declaration %>

E.g. : <html>
<body>
<%! int date = 50;%>
<% = "Value of variable is :" + date;%>
</body>
</html>

```
index.jsp  
<html>  
<body>  
<%!  
    int cube (int n){  
        return n*n*n;  
    }  
%>
```

<% = "cube of 3 is :" + cube(3)%>
</body>
</html>

Q. Differentiate Servlet & JSP & also write code for both!

Ans → Servlet is a java program that runs on Java enabled web server & applications.

Basically in this we request, then it is used to handle request by users, then process the request & sent back to users.

JSP % Java Server Page. It is used to insert Java code into html & xml.

SERVLET

- i) Servlet is a Java code
- ii) Writing code for servlet is harder.
- iii) It does not have inbuilt implicit obj.
- iv) In servlet we have to implement everything like business logic & presentation logic in just one servlet file.

) There is no method for running javascript on the client side in servlet.

JSP CODE %

```
demo.jsp
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Hello</title>
</head>
<body>
```

```
<%@ = " Pikachu!" %>
</body>
</html>
```

E.g.: Servlet config Web.xml file

```
<web-app>
  <Servlet>
    <Servlet-name> Pikachu </Servlet-name>
    <Servlet-class> DemoServlet </Servlet-class>
  </Servlet>
  <Servlet-mapping>
    <Servlet-name> -Pikachu </Servlet-name>
    <url-pattern> /welcome </url-pattern>
  </Servlet-mapping>
</web-app>
```

Q. List the various implicit objects of JSP.

Ans: ⇒ Implicit objects are set of Java obj. that the JSP contains makes available to developers on each page.

- i) request
- ii) response
- iii) config
- iv) application
- v) session
- vi) page context
- vii) page object
- viii) exception
- ix) out