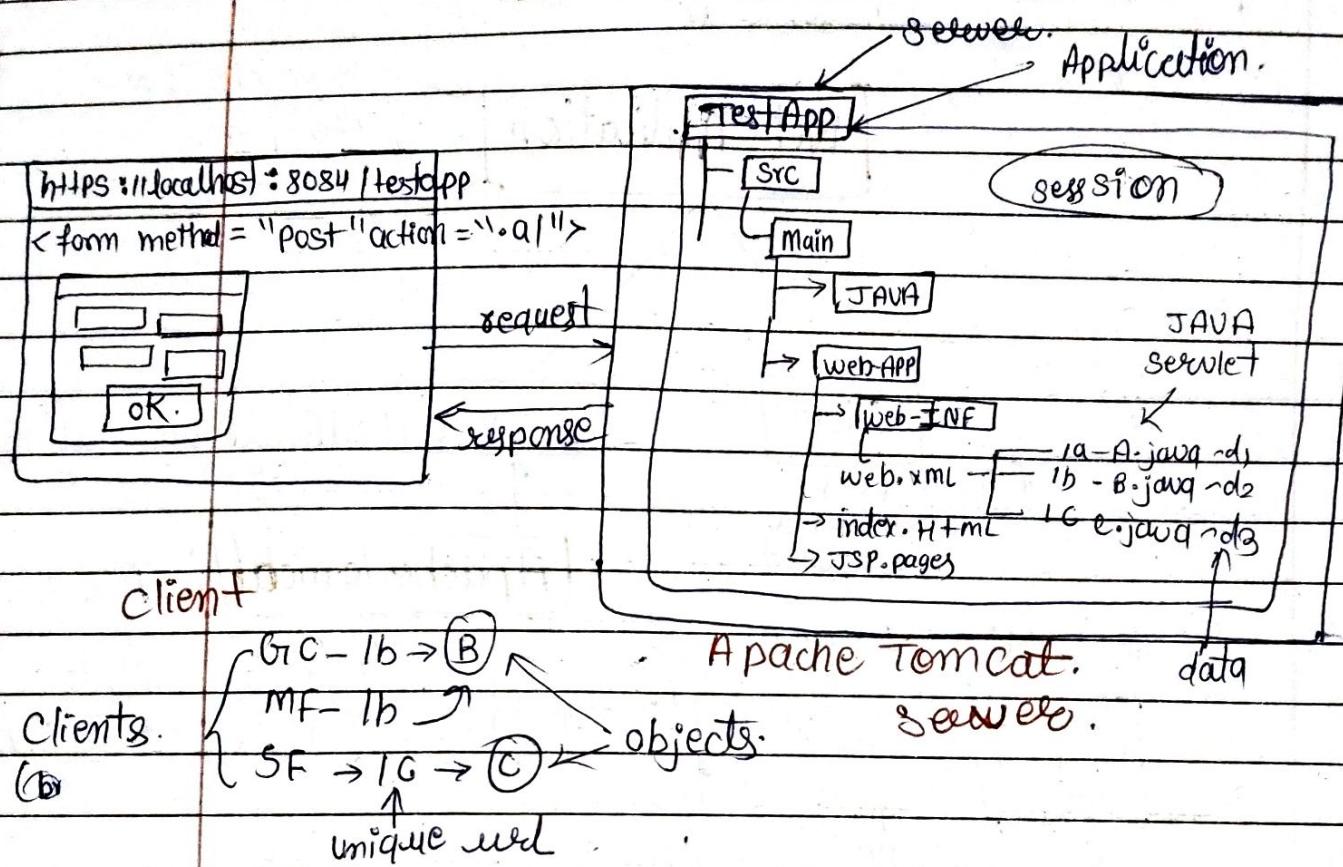


(Data Server Par tabhi Gayenga, jab app input field ko attribute name aise hone like name, email etc.)

Servlet Program :- Java servlet is a java program that runs on a web enabled server or application server.

- It handles client requests, processes them and generates responses dynamically. • Servlets are the backbone of many server side java application due to their efficiency & scalability.

Java Servlet Architecture



Every HTML pages, audio, video, files, JSP, pages care the resources.

HTML pages. IF found (then, we get the code in response)

IF not found [then,

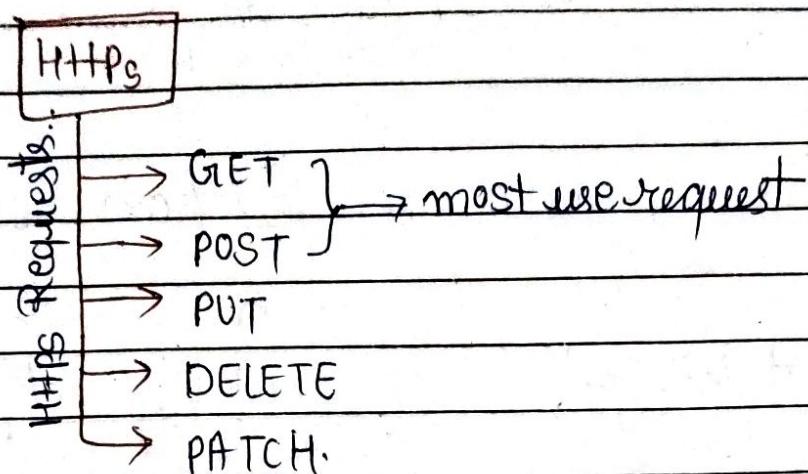
Error:- Request Resource Not available

Web.xml Configuration-file 404

(It stores the information of mapping of Java servlets)

like :— A.java,
B.java
C.java

web.xml (Deployment Descriptor file)



Difference b/w GET and POST

"GET"

(1) whenever, we want ^{to} get the data from server, we use "GET"!

(2) It is less - Secure, the data show on URL.

(3) it is use to get the / send the data to the server.
(less - amount of data)

(4) It is more - fast

(5) whenever we send a file in request, must use "GET".

"POST"

(1) whenever, we want to post / send the data to the server, use "POST"!

(2) It is more - secure, because when we post the data to the server, it doesn't show the URL (server side).

(3) It is used to send large - amount of data to the server.

(4) less - fast as compared to "GET"

(5) IDEMPotent,
you can bookmark the GET request.

PUT and PATCH :-

- To update the data in server-side

DELETE :- To remove the data from server-side



to make the data, Servlet-specific, than we do entry in servlet config of the the data

Data is public to every servlet

Servlet Container ← It is responsible to create the servlet's object.

Apache Tomcat

Servlet ← simple Java class.
(.java)

NOTE :- A Servlet object create only one

time when to create :- At the time of first request

GC → Ib → (B) ← Object
↓
URL pattern

client GC puts a request to of p.java (servlet)
than the object (B) create

GC → Ib → (B)

MF → Ib → (B)

SF → Ic → (C)

Ways To Create Servlet :-

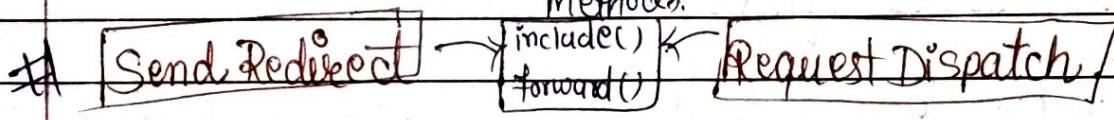
① Servlet (Interface) :- By implementing servlet interface

② Generic Servlet (class) :- By extending generic servlet class.

③. HTTP Servlet (class) :- By extending HTTPS servlet class.

• Servlet Config Object : - each servlet config object corresponding to each servlet

• Servlet Context Object :- servlet context object same for all methods.



Always create new request, we can move from one application to another application.

Dispatch the older request, same application par hi move kar sakte hai

Both are used, when we want to move from one application request to another application request.

• Non-authenticate web-page (do don't need authentication to access the page)

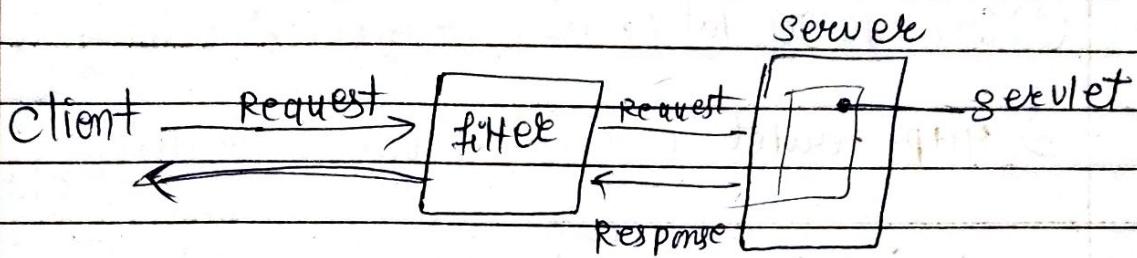
• Authenticate-webpage (need authentication to access the page)

object

Session

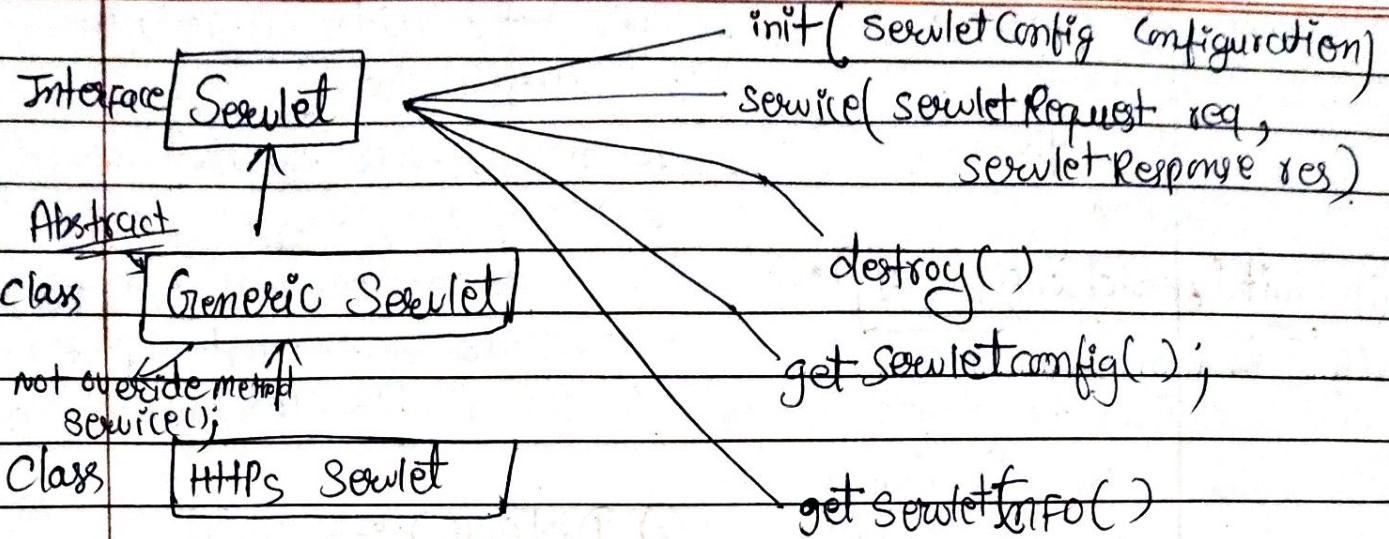
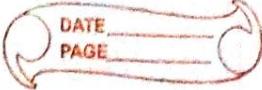
why we create session?

- To track the user/client
- one session for each client/user



- Request process hone se pehle, response send hone ke bad, kuch karna ho to, vo code filter me likha jata.
 - Filter kyu use karte hai :- Taki wahi request servlet tak jaye jo actual me real hai, yg authenticate hai islye filter lagaya hai
 - filter bhi response generate kar deta
- # filter :- used for pre-processing and post-processing (logging, authentication, etc.)

Servlet initialized only once
at base hi init() chalga, or.
contains both.



SERVLET Interface ↴ ↵ in servlet

(1) **init()** : instansized the object, when we create the object, once when servlet loads.

(2) **service()** : when the request sen per service for per request responsible to process the request
~~not overriding~~ Abstract method and send response,

(3) **destroy()** :— cleanup code. (servlet destroy hne se phebe khaane ka code)

(4) **getServletConfig()** :— It return the Config object.

When we want servlet specific data, then we write it into servletconfig

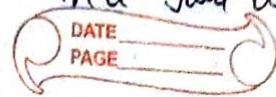
(5) **getServletInfo();**

Interface

Interface

public void service(ServletRequest request, ServletResponse response)

SERVLET LIFE CYCLE
Every Servlet (including HttpServlet) follows a lifecycle in a Java web container (like Tomcat etc.)



• Three main phases :-

① Initialization

② Request handling

③ Destruction.

① init (ServletConfig config) :-

Runs only once when servlet is first created (loaded into memory).

Purpose :- initialization of

Servlet.

uses :- DB Connection, Load configuration
Allocate resources.

After init(), the servlet stays alive in memory and is ready to serve requests.

② service (ServletRequest req,
ServletResponse res) :-

Called everytime a request comes to the servlet.

what it does :-
• Handles client request
• Produces response.

with HttpServlet, service()

internally dispatches request to:

doGet() for GET

doPost() for POST

Real life use :-

Normally you do not override service() in HttpServlet.

Instead, override doGet(), or doPost().

Why Not override service() in HttpServlet?
Because service() method already does :-

if GET → call doGet()

if POST → call doPost()

You override doGet() and doPost()
because it fits HTTP request types

③ Destroy() :- Called when servlet container removes from memory (servlet shutdown by code plug). Usage :- close DB connection.

② Release resources ③ Stop threads.

Methods to access Configuration &

Application Context

④ getServletConfig () :-

→ Returns ServletConfig object.

→ It contains initialization parameters (from web.xml or annotations).

Use Case :-

① You need servlet-level config

Eg :- DB credentials, API keys for a particular servlet

⑤ getServletContext () :-

Give access to ServletContext =

Application-wide shared data

Used Cases :- ① Share data among servlets.

② Get global init params

③ Read / write files of your web pages.

ServletConfig

ServletContext

local to servlet

Global to every servlet

< servlet >

< init-param >

< param-name > m1 < /param-name >

< param-value > Good Morning < /param-value >

< /servlet >

Servlet obj

when we want to make
the data-specific to servlet,
then.

servlet

config. →



< servlet >

< init-param >

< init-param >
< /init-param >

Has servlet

Globally define it ⇒

< context-param >

< param-name > GlobalMessage < /param-name >

< /context-param >

Has servlet & give name

automatically put the
data into the
servlet obj.
like alg

SERVLET \Rightarrow HTML in JAVA

Controller :-

↳ loginServlet (this request se, konsa response generate)

Karne hain, controller karta he

controller.)

getParameter :- return String

JSP :- Java Server Pages

Servlet = HTML in java

getSession() method returns HttpSession Set obj.

HttpServletRequest is the child of ServletRequest

↳ chain.doFilter(request, response); (filter method)

GET \rightarrow form par use.

POST \rightarrow [./LoginServlet] (path).

Servlet init() \Rightarrow RUNS ONLY ONCE

• Servlet class load hone ke bad. | why do they run only once?

• Instance create hone ke bad. | Because of the Singleton Concept

init() methods ke 2 variations :- | Servlet ya JSP ke liye server

① init(ServletConfig config)

ek hi instance banata hai

② init()

Ek baar instance ban

Flow:-

gaya, fir baat baar call
nahi hota

• Tomcat Servlet ko load karta hai

• Object banata hai

• init() first request ke time call
hota hai

• Baaki request ke liye service

service() ya doGet() / doPost()

run shote chain

init() \rightarrow instance creation time

Multiple requests \rightarrow

same instance reused.

"JSP"

"JAVA SERVER PAGES"

DATE _____
PAGE _____

JSP (JAVA SERVER PAGES)

JSP is a server-side technology used to build dynamic web pages using JAVA.

It lets you write JAVA CODE inside HTML page

JSP as :-

(custom tags)

HTML + Embedded Java + JSP tags
It is part of JAVA EE (JAKARTA EE) web technology stack.

Server-side code
Client-side code → JSP.

Tag.

[welcome-file list]

↪ Database ke.

■ How JSP works Internally :-

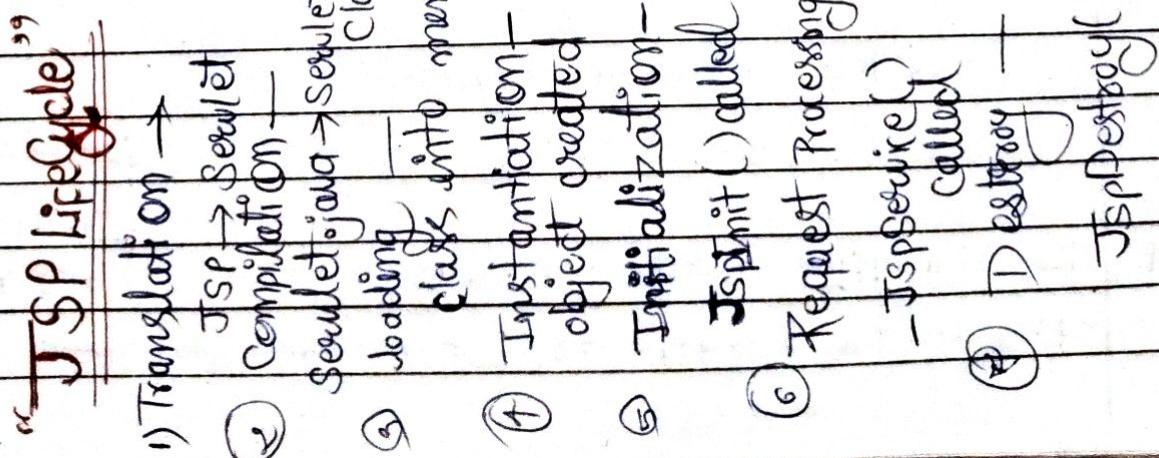
- ① Client (browser) → requests a JSP page (home.jsp) on try kahase Hogi
- ② JSP container → converts JSP into a servlet (.java) file.
- ③ Servlet → Compiles into .class
- ④ Generated servlet runs and produces HTML response
- ⑤ Response sent to browser.

A JSP page is compiled into a servlet the first time it's accessed.

for example :- your Request

https://localhost:8080/myapp/login.jsp.

The server converts login.jsp → login.jsp.java → Compiles → returns HTML.



JSP :- directly use request, response.

Runs only once during init.

DATE
PAGE

① Declarative Tag : if we want to write user defined class or method [`<%!` `%>` variables in JSP]

② Scriptlet Tag : [replacement of Service() method]
`<%` [responsible to process the request & send response]
`%>` [it is given to write JAVA code]

③ Expression Tag : [replacement of out.print (outputs value directly)]
`<% = expression %>`

IN JSP, there are 9 built-in objects.

Built-In Objects :— the container gives them automatically you don't need to declare them.

- ① request
- ② response
- ③ session
- ④ out
- ⑤ application
- ⑥ config
- ⑦ pageContext
- ⑧ page
- ⑨ exception

Example : Hello `<% = request.getParameter("name") %>`

MAIN DIFFERENCE

JSP	SERVLET
-----	---------

Dynamic UI, Business logic + UI logic

we act as controller.

Business logic and UI logic separate

Acts as a controller

Little example :—

```
<%  
for(int i=1; i<=5; i++) { %>
```

```
<h1> welcome </h1>
```

```
<% } %>
```

web-NF

E.JSP

JSP file ki mapping karne mandatory nahi he F.JSP

pr ase karne hu To web.xml me take sake

hai

As a programmer, hume pata ke kyo Model me data hai,

wah VI se ayega ("UI → Model")
④ "Directives tags" ➔

Package import karne ke liye.

• Page directive ➔

<%@ page import="java.util.Scanner" %>

<%@ page language="Java" contentType="text/html; charset=UTF-8" %>

• Include Directive (static include) ➔

<%@ include file="header.jsp" %>

JSP Scopes

Controls visibility of your variables

By Default

SCOPE	Duration	Available In	Scope is
Page	current page only	JSP	"Page"
request	for one request	Servlet / JSP	
session	per user		entire session
application	Entire web App		All users.

Realworld usage :-

you put thing inside
of init() or jspInit()
which you need to do
once :-

- ✓ Database Conn... pool initialization
- ✓ Load Configuration file
- ✓ Cache data from DB.
- ✓ Setup variables

Most Common interview point :-

JSP ka actual servlet class
automatically generate hota
hai, us class me jspInit()
method override hoti hai -

jo initialization ke
time server call
karta hai.

Properties
→ Deploy assembly → maven dependencies

DATE
PAGE

form submission → HttpServlet Create

POJO + Serializable ⇒ Bean → Java class.

`<jsp:useBean id="admin" class="com.info.todoapp.entity.Admin" />`

⇒ it creates the object (Admin) of Admin class

`<jsp: SetProperty name="admin." property="*"/>`

admin object ki vari properties me, form se aayi data directly copy hoga, copy tabhi hoga ja. form me input field ke name or Data Property

HTTP is a state-less protocol.

⇒ all requests are independent

JSP ⇒ problem ye hai ke ye post or get dono ke sath chal jata, esliye apni check lagegi.

`String methodType = request.getMethod();`
IF (methodType.equalsIgnoreCase("POST")) {

}

JavaScript Tag :

< Script >

```
Window.alert("Category saved successfully.");
window.location.href = ".../add-category.jsp";
```

</ Script >

Ways to Send the data :-

Request body, Query Parameter, URL Request

Confirm() :- Return boolean, (JavaScript)

System Class :-



Method

```
long date = System.currentTimeMillis();
```

```
SimpleDateFormat sd = new SimpleDateFormat("dd/MM/yyyy");
```

(VXXD)