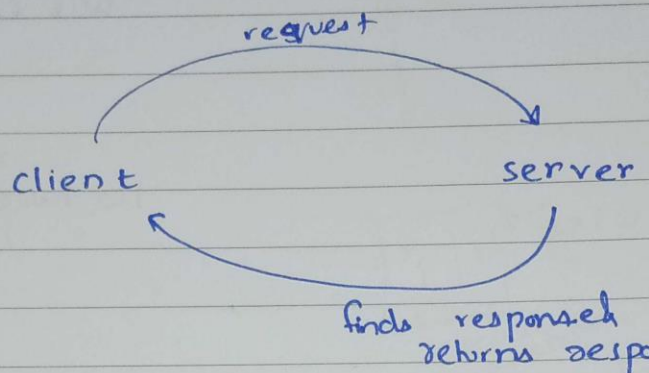
Date 20
M T W T F S S

↔ JSP : Java Server Pages

Fundamentals of the Web:
- Clients & Servers.
- Connected through wire/wireless networks.



response : in the form of any "content" in the form of HTML
client sends & receives requests from a web server typically.

website : HTML Pages images etc.

web app : website with dynamic functionality on the server
FB/ Twitter, etc.

HTTP: Hyper Text Transfer Protocol - A protocol for client-server communication over the web.

→ Is a stateless protocol. [Only one requests per client]
(That is handles one request, the disconnects, instead of keeping up a constant connection)

Why? (so more users can interact with it)

HTTP Methods: GET
POST
(other methods are there too)

Date 20
M T W T F S S

SERVLETS : ∘ A technology used to create web applications.

∘ An API that provides many interfaces & classes.

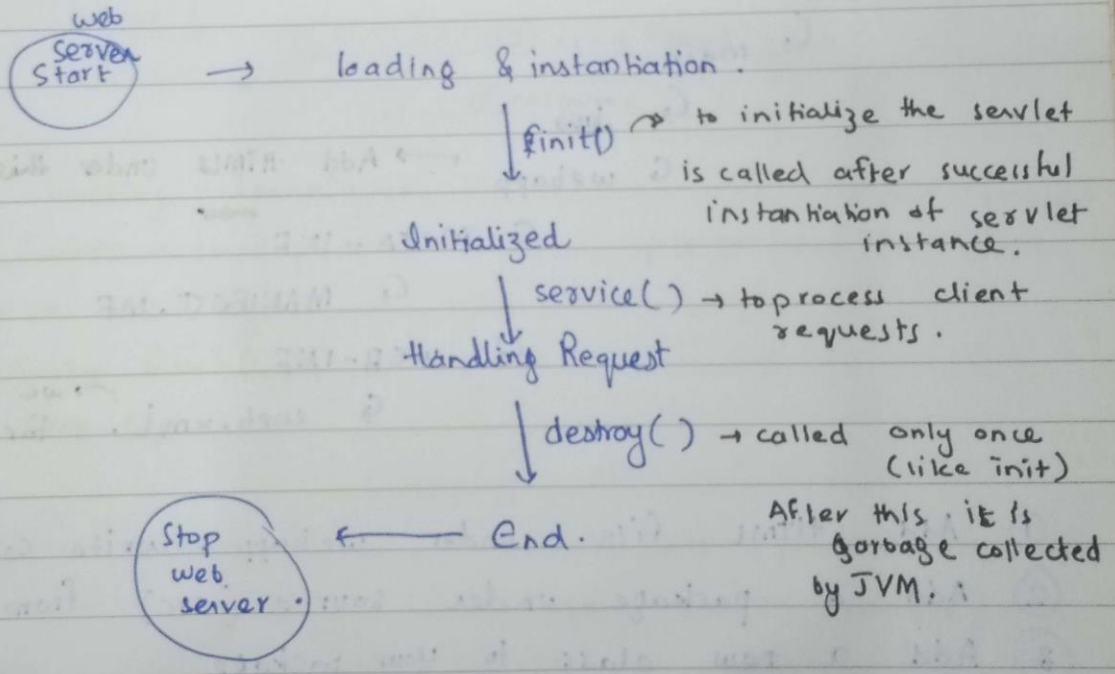
∘ An Interface that must be implemented for creating any servlet.

∘ A class that extends the capabilities of the servers & responds to incoming requests

∘ A web component that is deployed on the server to create a dynamic web page

Servlets are robust & scalable

Servlet Life Cycle



Steps for creating a Servlet.

1- Create & compile simple servlet code.

2- Add Mappings to web.xml file.

3- Start Apache Tomcat.

4- Start web browser & request servlet

- Create New Server :
  - add a runtime environment (installation directory)
  - Double click server to view details.
  - Select "Use tomcat installation" under Server Locations.
  - right click, "start" to start server
  - Go to localhost:8080

- Creating a Servlet :

New → Dynamic Web Project → next → select "Generate web.xml"

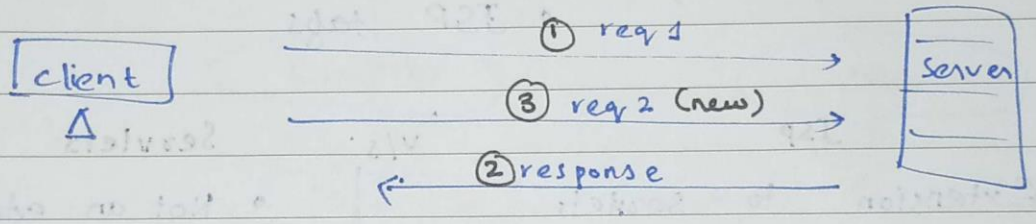
src
- main
  - Java
  - webapp → Add HTMLs under this
    - META-INF
      - MANIFEST.MF
    - WEB-INF
      - web.xml → we add mappings here for servlet

1. Add .HTML file under webapp : write some HTML.
2. Add a package under source (src) from JavaResources
3. Add a new class to your package
4. Import HTTPServlet & extend your class from it.
5. Write logic in class.
6. Edit web.xml to add mappings. ① add servlet → ① servlet name ② servlet class
7. add servlet mapping ① servletName ② url-patern

Date M T W T F S S

Generic Servlet : → is an abstract class, with abstract method service()
a protocol independent servlet that should always override the "service()" method to handle client req.
↙ ↘
has 2 parameters
↙
servlet request object — servlet response object.

Session Tracking | [Session] Management in Servlets. → a way to maintain state/data of a user.
Http is stateless → server does not remember the users. Each request is a new request to the server.



Session Tracking Techniques: client request
- Cookies → small piece of information w/ is persisted b/w multiple
- URL Rewriting.
- Hidden form field
- Http Session

Has a:
Name, single value, comments, etc (path domain qualifiers age, version number)

Cookie is sent with the server response first, then it is sent back to server w/ newer responses.

JSP

→ Java Server Pages.

↳ You may call it an extension of servlets

→ Also used to make web apps.

- fast development
- easy to maintain
- Portable
- Powerful & flexible

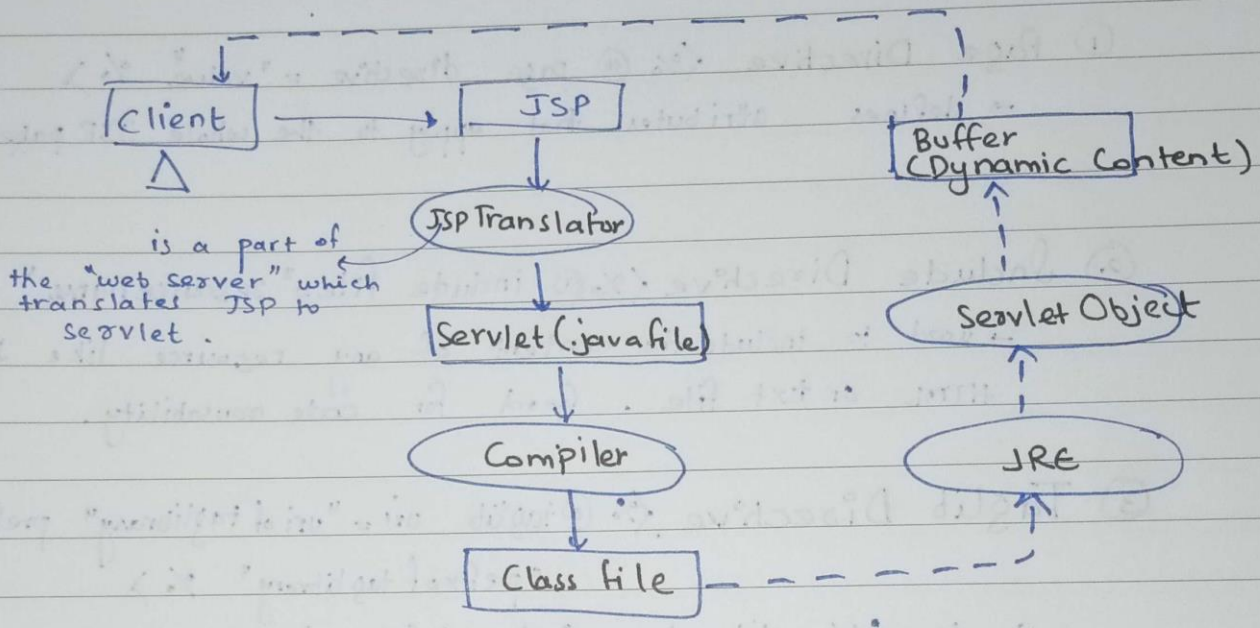
Normal JSP page contains:

- HTML Tags
- JSP tags

JSP	v/s	Servlets
◦ Extension to servlets		◦ Not an extension to JSP
◦ Easy to maintain		◦ Not as easy
◦ No need to recompile & redeploy		◦ Needs to be recompiled.
◦ Less code than servlets		◦ More code
◦ Allows custom tags, expression language		
◦ Easy to separate business logic & presentation knowledge.		

=> Life Cycle of JSP

- Compilation → Parsing the JSP, Turning JSP into a servlet, Compiling the servlet.
- Initialization → calls jspInit()
- Execution → calls _jspService()
- Clean Up



Steps to create a JSP Page:

1. Create a dynamic web Project
2. Create JSP File
3. Start TomCat.
4. Deploy the project on the server.

→ Scriptlets let you inser java code in the JSP

<% java source code %>

anything written here is returned to the output stream of the response → Expression <% = statement %>

→ Declaration <%! field/method declaration %>

↳ code here is outside the service method of the servlet, so that it does not get memory at each request

Date 20
M T W T F S S

→ Implicit Objects of JSP

out, request, response, config, application, session, page Context, page, exception.

→ JSP Directives Elements.

① Page Directive <%@ page directive = "value" %>

→ defines attributes that apply to the whole JSP page

② Include Directive <%@ include file = "resource name" %>

→ used to include the content of any resource like JSP or HTML or txt file. Good for code reusability.

③ Taglib Directive <%@ taglib uri = "uri of taglibrary" prefix = "prefix of taglibrary" %>

→ to add libraries that define tags.

JSP Request is an implicit object of type HTTP request that is created for each JSP request by the web container. It can be used to get request information such as parameters, header information, server name & port, remote address, content type, character encoding, etc.

Can be used to set, get & remove attributes from JSP request scope.

Date _____ 20
M T W T F

Java Beans

- a reusable software component
- encapsulates many objects into one object
- easy maintenance
- provides access via getters & setters.