UNIT - IV

chapter 1 - Getting started with web applications

A web application is an application program that is usually stored on a remote server, and users can access it through the use of software known as web-browser.

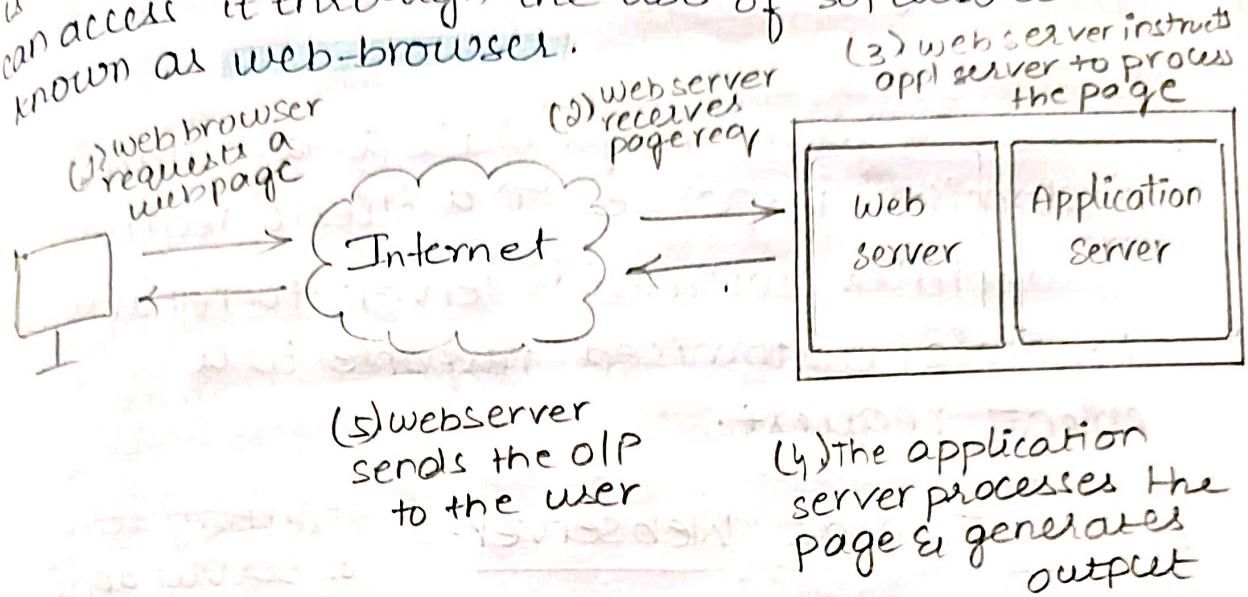


Fig. Flow of the Web Application.

web client ⇄ web browser or Internet ⇄ web server (also contains application server to process the page)

- Web pages can be
  1. static : does not require any processing on the server
  2. dynamic : requires server-side processing.
- To operate a web-application, we usually require a web server to mange the clients upcoming requests and requires an application server.
- Application server performs tasks requested by the client and need database sometimes to store the information.

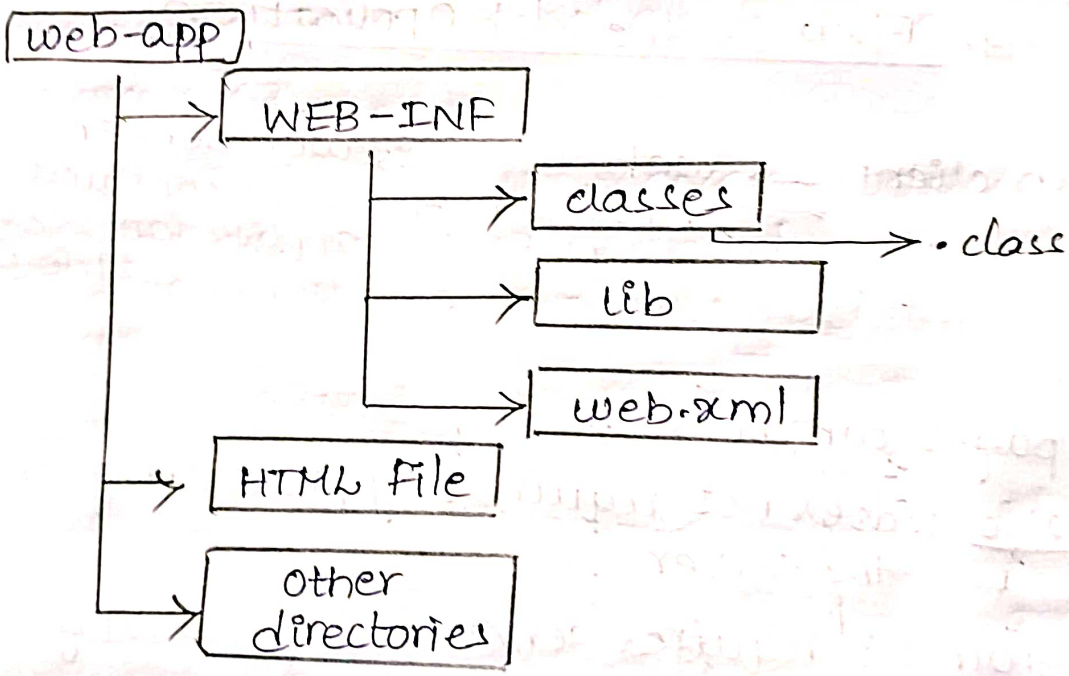
Web server

- Software and hardware that uses HTTP and other protocols to respond to client requests
- Functionalities: storing, processing and delivering web pages

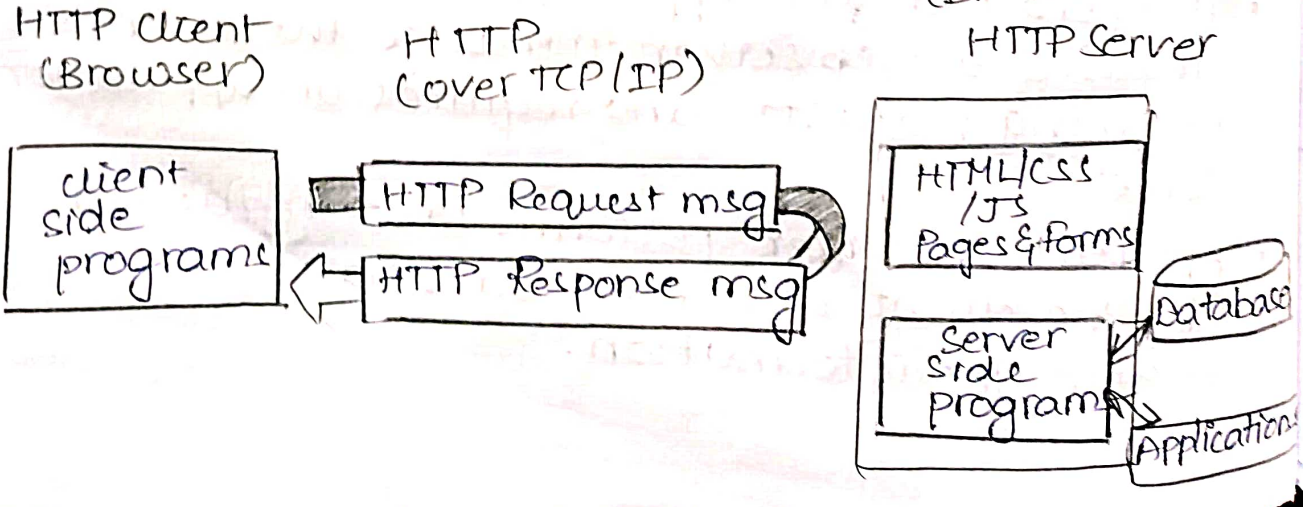
Application server

- Enables a server to generate a dynamic, customised response to a client request.
- Collaborates with web server to return a dynamic, customised response to a client request.

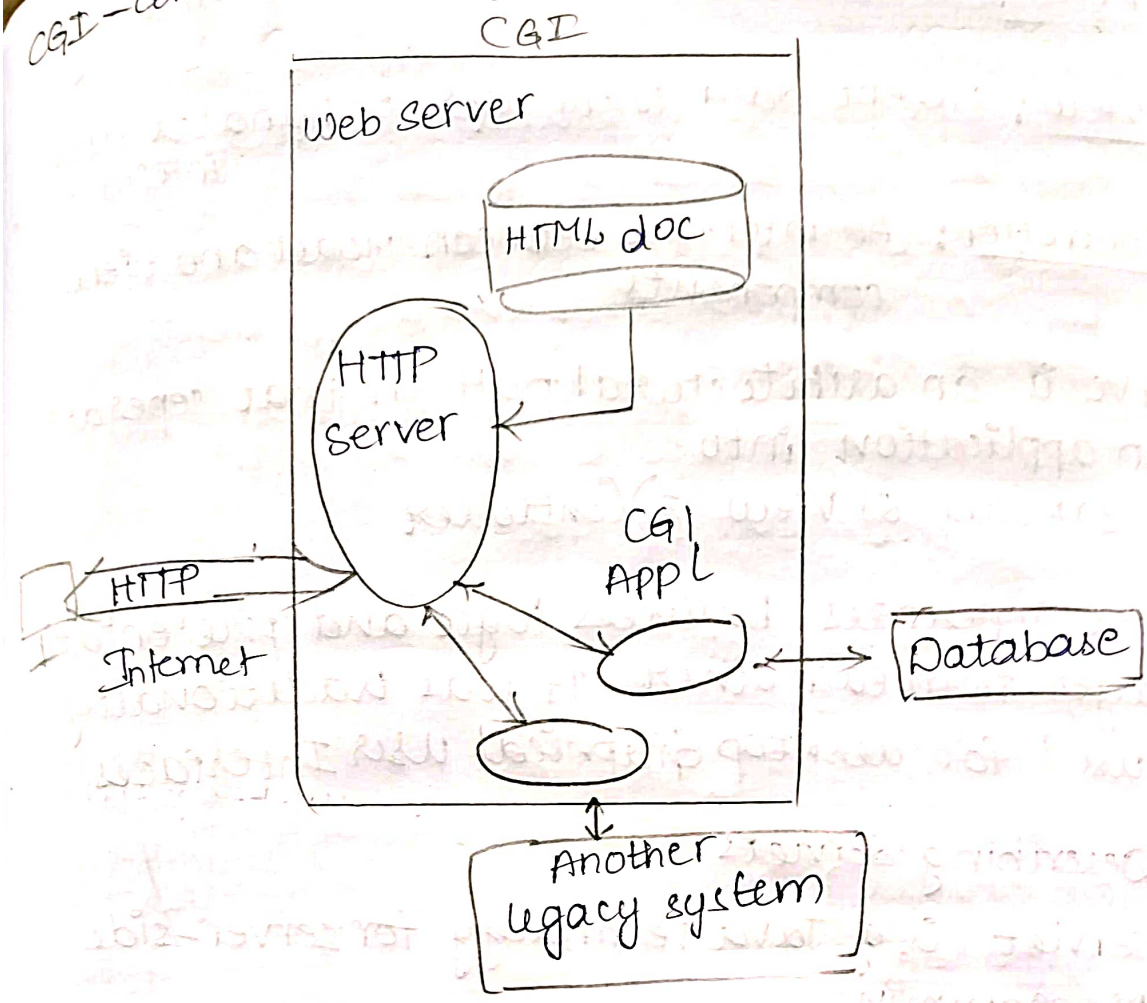
Apache Tomcat Webserver – directory structure of servlet application



Web Application using HTTP protocol

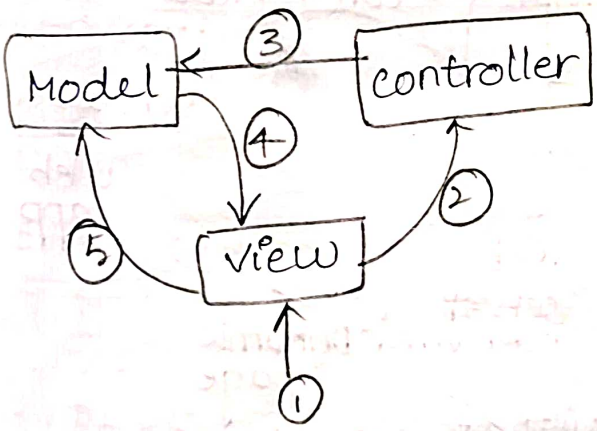


CGI - Common Gateway Interface.



- A few disadvantages in CGI implied to moving to servlet technology.

MVC Architecture



1. User interacts with a view
2. View alerts controller of particular event
3. Controller updates model
4. Model alerts view that it has changed
5. View grabs model data and update itself.

Model: It includes all data and its related logic.

View: Present data to the user or handles user interactions

Controller: An interface between Model and view components

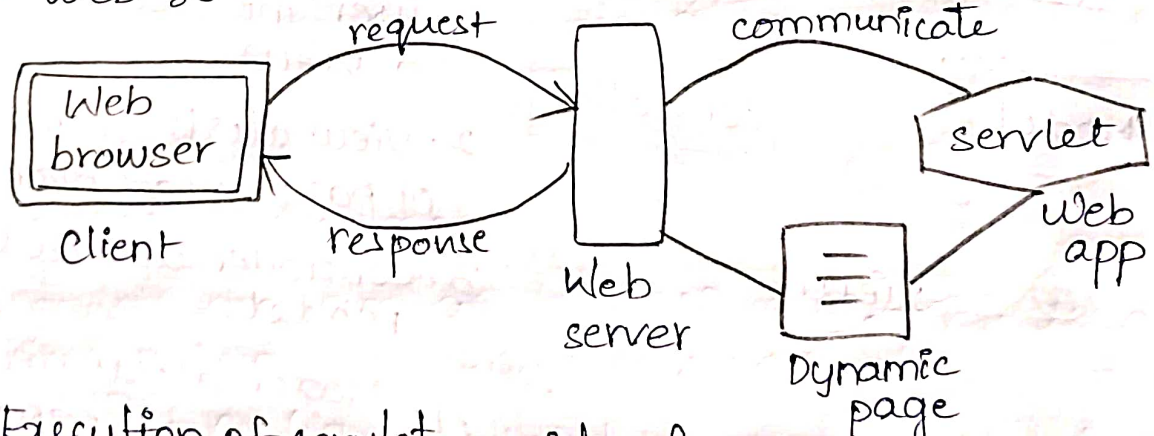
MVC is an architectural pattern that seperates an application into
1) Model 2) View 3) Controller

MVC seperates business logic and presentation layer from each other. It was traditionally used for desktop Graphical User Interfaces

Describing servlets

servlet is a Java technology for server-side programming.

It is a java ~~based~~ ~~webserver~~ module that runs inside a java enabled web server and services the requests obtained from the web server.



- Execution of servlet consists of 4 steps:-
  - client sends request to server
  - The server alerts appropriate servlet
  - The servlet process the request and generates the output (if any), and sends back to the webserver
  - The webserver sends back response to the client

Adv. over Applets

. Applets are downloaded from the server and executes in clients browser
  - uses JRE
  - Problems with browsers (not compatible)

. servlets run on the server machine and generates html code.

servlets Vs CGI	servlet	CGI
Portability	✓	X
sharing data	✓	X
Direct communication	✓	X
cost	low	high
Handle cookies	✓	X

Advantages of servlet.
- Better performance
- Portability
- Robust
- Secure

Exploring features of servlets

Servlet - a Request-Response Model

Generic servlet - unrelated to protocol. it accepts any type of servlet request

HTTP servlet - Accepts HTTP type servlets requests.
(child class of Generic servlet (It automatically extends GenericServlet)

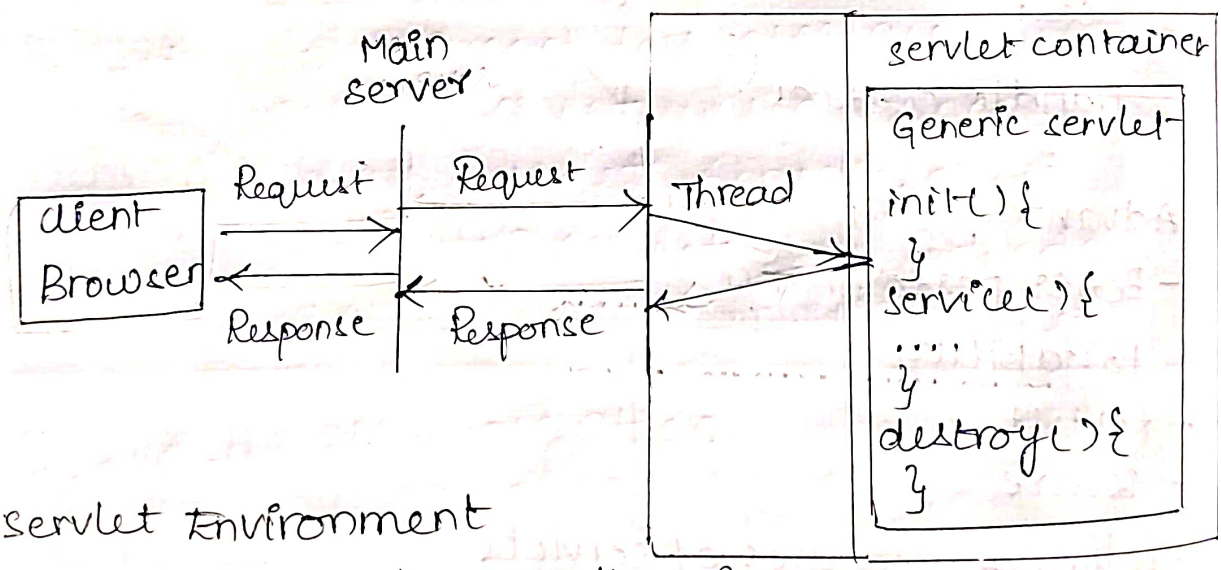
- Usage model.
  - servlets can be chained together into filter chains (recording all incoming requests), logs the IP addresses of computers from which requests originate, conversion, data compression.
  - support protocols such as HTTP
  - replacement of CGI

Sources of servlets

Primary servlet methods
- init()
- service()
- destroy()

Applet

init()
start()
paint()
stop()
destroy()



Servlet Environment

- servlets are similar to other java objects and runs within the server environment
- Servlet Config - related to only one servlet
  - used to obtain configuration information
  - deployment descriptor (web.xml)
- Servlet context
  - It is related to multiple servlets of a web appl

javax.servlet.Servlet (Interface)

↓ Implements

javax.servlet.GenericServlet (class)

↓ Extends

javax.servlet.HTTPServlet (class)

Methods
- init()
- service()
- destroy()
- getServletInfo()
- getServletConfig()

javax.servlet.Servlet (Interface)

javax.servlet.GenericServlet (class)

- init()
- service()
- getInitParameter()
- getInitParameterNames()
- getServletInfo()
- getServletConfig()
- getServletContext()
- destroy.

javax.servlet.HttpServlet (class)

- doPost()
- doGet()
- service()
- doDelete()
- doPut()
- doHead()
- getLastModified()

javax.servlet package

Interfaces.
- Servlet
- ServletConfig
- ServletContext
- ServletRequest
- ServletResponse
- RequestDispatcher
- FilterChain
- FilterConfig

classes
- GenericServlet
- ServletInputStream
- ServletOutputStream
- ServletException
- UnavailableException

servletInputStream
To read data given by client (binary data)

servletOutputStream
To write binary data / for sending binary data to the client

GenericServlet
To define a generic & protocol independent servlet

servletException
Problem with user request or problem to send response

unavailableException
Requested resource not available

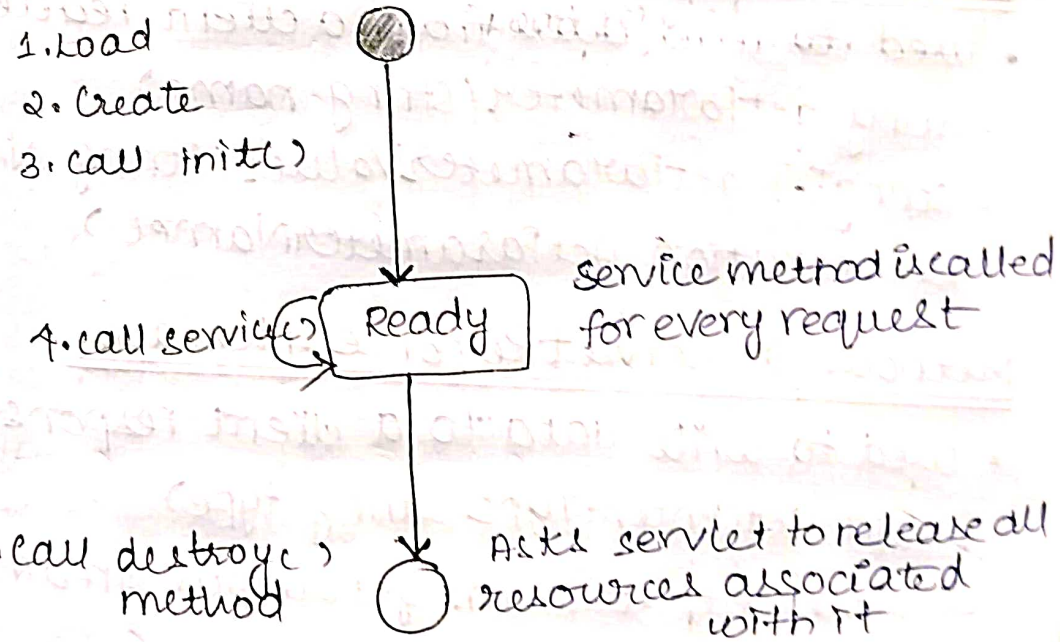
Interface	
servlet	to implement this interface, write a generic servlet which extends javax.servlet.GenericServlet or an HTTP servlet that extends javax.servlet.http.HttpServlet
servlet config	pass information to the servlet during initialization.
servlet Context	Information related to webap available in web.xml file is stored in servletContext object.
Filter	To perform filtering tasks either on request or response
Request Dispatcher	It receives requests from the client and sends them to a servlet/ HTML file/JSP file on the server. (sending request from one resource to other)

servlet request	Defines an obj that is created by the servlet container to pass client request information to a servlet.
servlet response	To pass response information to client

Lifecycle of a servlet

- Load servlet class
- create servlet instance
- call init() (checks if there are initialisation parameters)
- call service() (procedure)
- call destroy() (memory deallocation)

- Processing request is created as a thread but not as a process for better memory utilisation.



service() method syntax

```
public void service(ServletRequest req, ServletResponse res)
    throws ServletException, IOException
```

init() method

```
public void init(ServletConfig config) throws
                                    ServletException.
```

Other methods

- ServletConfig getServletConfig();
  interface method.
- String getServletInfo();
  servlet Information => version, author.

Methods in servletConfig Interface

- allows servlets to get initialization parameters
  - String getInitParameter(String name)
  - Enumeration getInitParameterNames()
  - ServletContext getServletContext()
  - String getServletClass()

Methods in servletContext Interface

- used to read data from a client request
  - String getParameter(String name)
  - String[] getParameterValues(String Name)
  - Enumeration getParameterNames()

Methods in servletResponse Interface

- used to write data to a client response
  - void setContentType(String type)
  - ServletOutputStream getOutputStream()
  - PrintWriter getWriter()

Methods in RequestDispatcher Interface.

1. public void forward (ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException;

- Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.

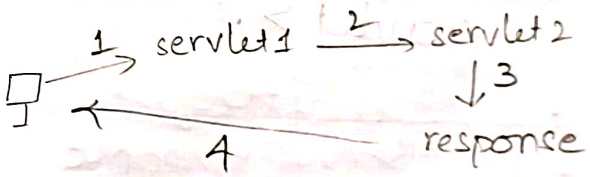
2. public void include (ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException;

- Includes the content of a response resource (servlet, JSP page, or HTML file) in the response.

- Request dispatcher will dispatch a request (forward) from one servlet to the other and accepts the response (include) back from that servlet

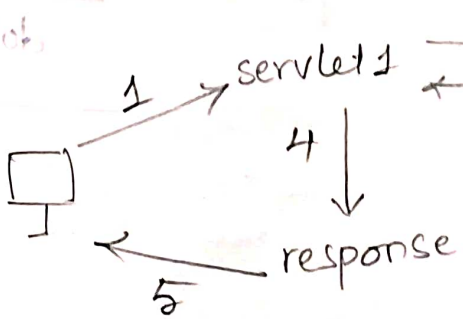
FORWARD:-

1. Request
2. Forward
3. Response is generated
4. Response is sent back to the client/browser.



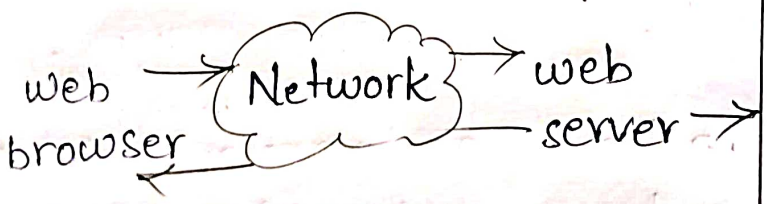
INCLUDE:-

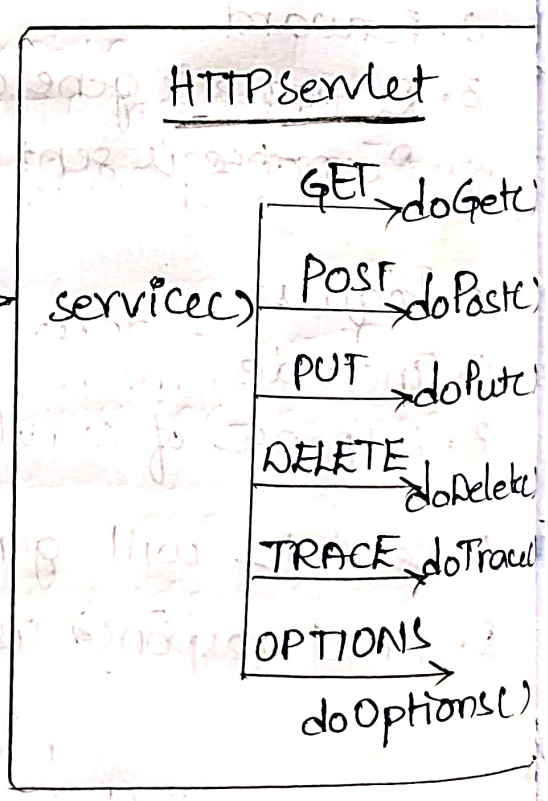
1. Request
2. Include
3. Response of servlet is included in servlet 1 response
4. Servlet 1 will generate final response
5. Final response is sent back to the client.



	ServletConfig	ServletContext
1.	servlet specific	For whole application
2.	ServletConfig object is obtained by getServletConfig() method	servletContext object is obtained by getServletContext() method.
3.	Each servlet has got its own servletConfig object	ServletContext object is only one and use by diff servlets of the application
4.	use servletConfig when only one servlet	use servletContext when whole application needs.
5.	Parameters are present as name-value pair in <init-param> inside <servlet>.	Parameters of servletContext are present as name-value pair in <context-param> which is outside of <servlet> and inside <web-app>

HTTPServlet (javax.servlet.http) package:

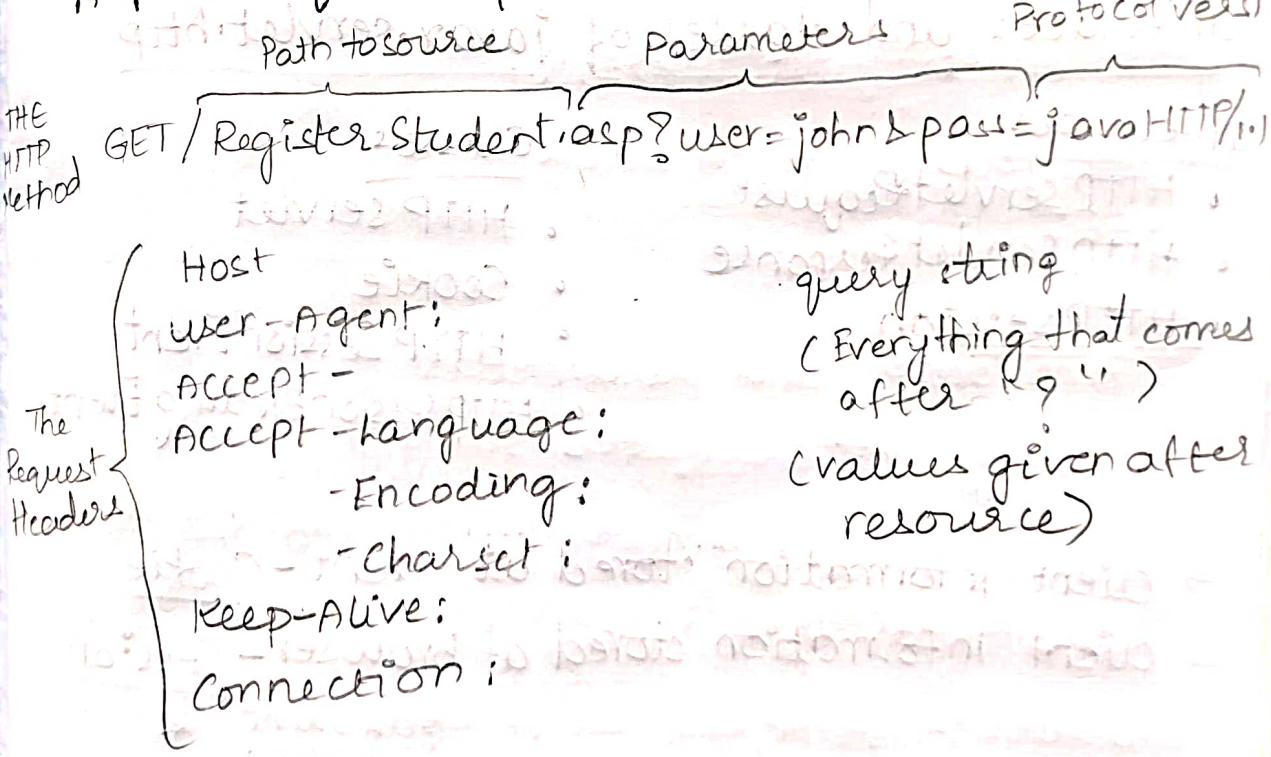


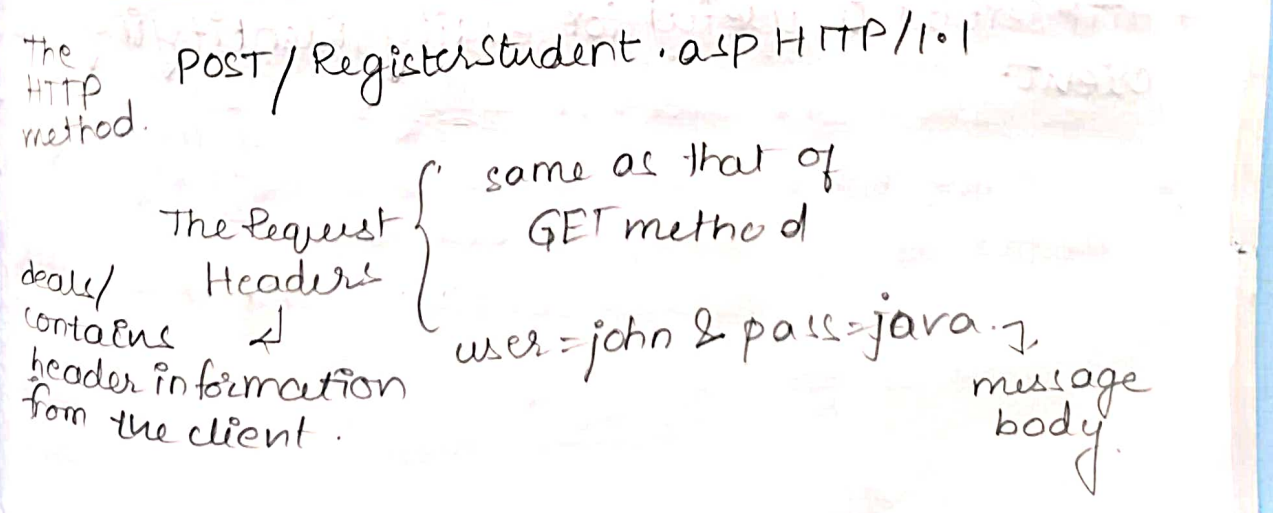


more methods :-
Head()

4/9/24

- class HTTP servlet defines methods doGet and doPost to respond to get and post requests from a client, respectively.
- These methods are called by the service method, which is called when a request arrives at the server.
- Method service() first determines the request types, then calls the appropriate method for handling the request.
- A get request gets data from a server
  A post requests &posts (or sends) data to server.





bookmarking - locating client address

GET - read operations
- can be bookmarked
- can be cached

POST - update & modify operations
- cannot be bookmarked
- Not cached

cookie :- storing information of client at client itself

User-Agent :- According to browser

→ while client sends request it also sends some header information along with it.

Interfaces and classes of javax.servlet.http

Interfaces
- HTTP ServletRequest
- HTTP ServletResponse
- HTTP Session

Classes
- HTTP servlet
- Cookie
- HTTP SessionEvent
- HTTP SessionBindingEvent

→ client information stored at client - Cookie

→ client information stored at browser - Session
permanent storage

→ HTTP Servlet is useful for easily identifying client.

Methods of HttpServlet class:-

- doGet (HttpServletRequest req, HttpServletResponse resp)
- doPost()
- doPut()
- doDelete()
- doOptions()
- doTrace()
- doHead()
- service()

req, resp parameters are present for all methods.

Methods in HTTPServletResponse Interfaces

- void addHeader(String name, String value)
- void addCookie(Cookie cookie)
- setContentType(String)
- boolean containsHeader(String name)
- void setHeader(String name, String value)
- void sendRedirect(String location)

Methods in HTTPServletRequest Header.

- String getHeader(String name)
- Enumeration getHeaderNames()
- String getMethod() - which method used (get, post -->
- String getQueryString()
- String getRequestURI()
- Cookie[] getCookies()
- HTTPSession getSession()
- HTTPSession getSession(boolean create)
- String getProtocol()
- StringBuffer getRequestURL()
- String getServletPath. (identify path of servlet)

Header information from client to server

The HTTP method: Path to source on web browser | Parameters (query string) to the server | Protocol version browser supports

GET/RegisterStudent.asp?user=jhon&pass=java HTTP/1.1

The Request Headers:

Host: guru99.com
User-Agent: Mozilla/5.0
Accept-text/xml, text/html, text/plain, image/jpeg
Accept-Language: en-us, en
-Encoding: gzip, deflate
-charset: ISO-8859-1, utf-8
Keep-Alive: 300
Connection: keep-alive

HTTP servletRequest Interface

- The role of form data
  - GET
  - POST
- Form data and parameters
  - string getParameter(String name)
  - String[] getParameterValues(String name)
  - Enumeration getParameterNames()

Headers :- HTTP Request ~~header~~

- HTTP headers are sent by client to the server about the software, language and other information the client is using.

* HTTP request header can be accessed by server using various methods *

HTTP request interfaces contains the method related to the following.

1. Type of request
2. Form Data
3. Request Header Information
4. File uploads.

HTTP Response header:

HTTP Response Interface contains the following methods to manipulate
- methods to manipulate HTTP Response headers
- Response redirection methods
- Methods related to automatic refreshing or waiting of webpages
- methods to store the client information and to deal with the content type.

Response Header Information
It contains the following information

Header Field	Description
Age	+ve number, seconds, represents estimated time since last response
Content-Length	Length of the content which is sent to client
Content-type	MIME type
Date	Represents date and time
Location	specifies the location of new resource - send Redirect
Retry-after	server is busy, retry after some time later
Server	Provides description about the server
Refresh	Value in seconds

Response Redirection

- the sendRedirect() method of HttpServletResponse interface can be used to redirect response to another resource, it may be servlet, jsp or html file.
- It accepts relative as well as absolute URL.
- It works at client side because it uses the URL bar of the browser to make another request.