UNIT-3

SERVLETS

1. Introduction to Servlets ? Advantages) Applications ?)
2. Common Gateway interface (CGI) ? Difference between Servlets and CGI ?
3. Life Cycle of Servlet [Architecture of Servlet]
4. deploying a Servlet [steps for deploying Servlet], Tomcat Server, Difference between tomcat and JDSK
5. The Servlet API [interfaces and classes]
6. Reading Servlet parameters [or form Data]
7. Reading Initialization parameters
8. Handling Http Request & Responses.
9. Using Cookies and Sessions
10. Connecting to a database using JDBC.

① Java Servlet

⇒ 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 the web Server, process the request, produce the response, then Send a response back to web Server.

Properties of Servlets:-

① Servlets work on Server Side.

② Servlets are Capable of handling Complex requests obtained from the web Server.

⇒ fig below shows Servlet Architecture.

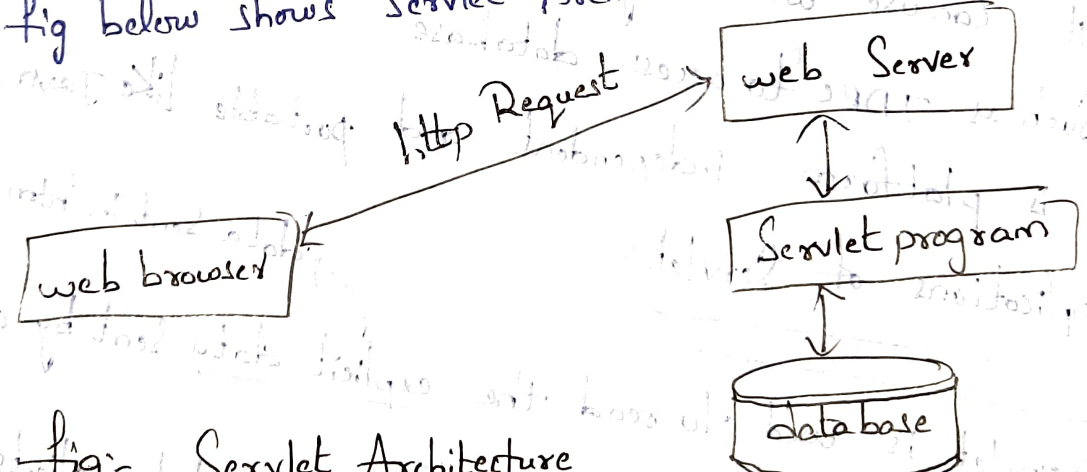


fig: Servlet Architecture

⇒ ~~Exect~~ Execution of Servlets basically involves Six basic Steps:-

① client Send the request to the web Server.

② The web Server receives the request.

③ web Server passes the request to the Corresponding Servlet.

④ The Servlet processes the request and generates Response in the form of output.

⑤ The Servlet Sends the response back to the web Server.

⑥ The web Server Sends the response back to client and client browser display it on the Screen.

Advantages of Java Servlet:

⇒ Servlet is faster than CGI as it doesn't involve the creation of a new process for every new request received.

⇒ Servlets, as written in Java, are platform independent.

⇒ need less memory with good Security

⇒ It Can use wide range of API's Created on java platform Such as JDBC to access database

⇒ It is platform independent and portable like Java.

Applications of Servlet:

data Sent in html form

① Servlet is used to read the explicit data Sent by clients (brows browser.

② Servlets is used to read implicit data Sent by clients. This includes cookies, media files, etc.

③ Process the data and generate results

④ we can Send explicit data and Implicit data to browser.

⇒ Servlet techn
at Server si

⇒ Servlet is an
Including a
be Impleme

② COMMO

. CGI is actua
using any of
this is resp
dynamic Conte

In CGI A
access dynamic
following operat

1) when user Sen
webpage and
Client reques

2) For each client
and it destroy
client.

⇒ Servlet technology is used to create a web application resides at Server side and generates a dynamic web page.

⇒ Servlet is an API that provides many interfaces and classes including documentation. Servlet is an Interface that must be implemented for Creating any Servlet.

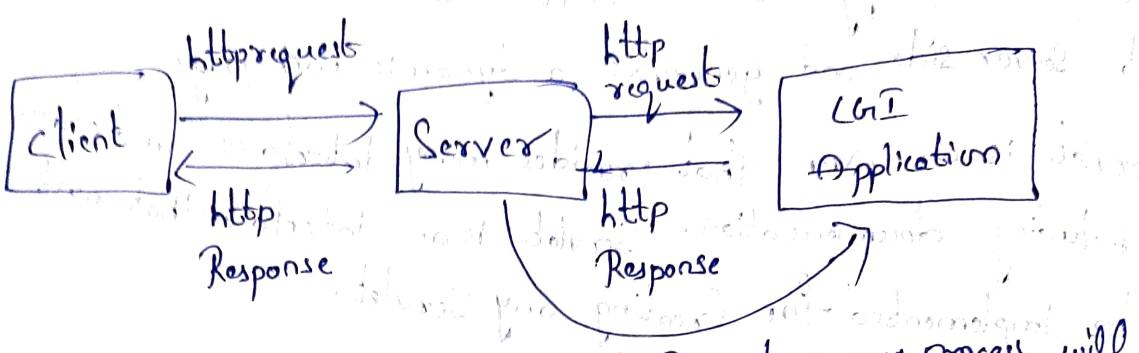
2) COMMON GATEWAY INTERFACE (CGI)

, CGI is actually an External Application that is written by using any of the programming languages like C or C++ and this is responsible for processing clients request and generating dynamic Content.

In CGI Application, when a clients makes a request to access dynamic web pages, the web Server performs the following operations.

1) when user Send request then CGI first locates the requested web page and then Creates a new process to Service the Client request.

2) for each clients request CGI Application Creates new process and it destroy process after providing HTTP response to the client.



For each Request a new process will be created

fig: Common gateway Interface (CGI) for Processing a clients Request

⇒ So, in CGI Server has to create and destroy the process for each request. It's easy to understand that this approach is applicable for handling few clients but as the number of clients increases, the workload on the Server increases and so the time is taken to process requests increases.

Difference between Servlet and CGI

(or)

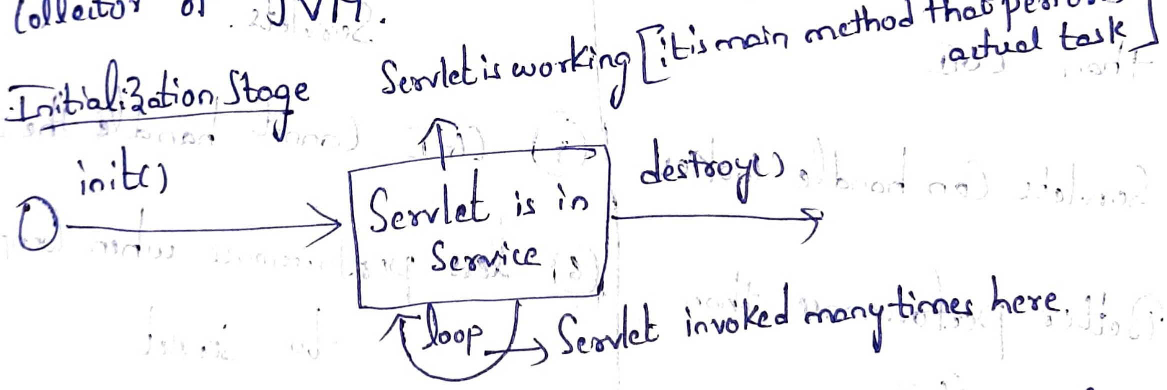
Advantages of Servlet over CGI.

Servlet	Common Gateway Interface (CGI)
1) Servlets are portable and efficients	① CGI is not portable.
2) In Servlets, sharing data is possible	② In CGI, sharing data is not possible.
3) Servlets can directly communicate with web Server	③ CGI Cannot directly Communicate with the web Server
) Servlets are less expensive than CGI	④ CGI is more expensive than Servlets.
) Servlets Can handle Cookies	⑤ CGI Cannot handle the cookies.
) Better performance than CGI	⑥ less performance when Compare to Servlet
) platform independent	⑦ platform dependent
) we can communicate with other Applications like RMI ↳ Remote method Invocation	⑧ Cannot Communicate with other Applications
) more Secure than CGI	⑨ less Secure Compared to Servlet.

③ Life Cycle of Servlet :-

A Servlet lifecycle Can be defined as the entire process from its Creation to till the destruction. The following are the paths followed by a Servlet.

① The Servlet is Initialized by Calling the init() method.
② The Servlet Calls Service() method to process a client's request.
③ The Servlet is terminated by Calling the destroy() method.
④ Finally, Servlet is garbage Collected by the garbage Collector of JVM.



Initialization Stage
Servlet is working [it is main method that perform actual task]
init()
Servlet is in Service
destroy()
loop
Servlet invoked many times here.

As displayed in the above diagram, there are three States of a Servlet: new, ready, end. The Servlet is in new State if Servlet instance is Created. After invoking the init() method Servlet Comes in the ready State. in ready State, Servlet performs all the tasks. when the web Container invokes the destroy() method, it shifts to end State.

① Servlet class is loaded:

The classloader is responsible to load the Servlet class. the Servlet class is loaded when the first request for the Servlet is received by web Container.

② Servlet instance is Created:-

The web Container Creates the instance of a Servlet after loading the Servlet class. the Servlet instance is Created only once in Servlet life Cycle

③ init method is Invoked:-

The web Container Calls the init() method only once after Creating the Servlet instance. The init method is used to initialize the Servlet.

Syntax:- public void init(ServletConfig Config) throws Servlet Exception

④ Service method is invoked:

The web Container Calls the Service method each time when request for the Servlet is received. if Servlet is not initialized, it follows the first three Steps as described above and then Calls the Service method. if Servlet is initialized, it Calls the Service method. Servlet is initialized only once.

The Syntax of Service method is

```
public Void Service (Servlet Request request, Servlet Response response)
    throws Servlet Exception, IOException }
```

⑤ destroy method is invoked :-

The web Container Calls the destroy method before removing the Servlet instance from the Service. It gives the Servlet an opportunity to cleanup any resource for example memory, thread etc. The Syntax of destroy method is given below.

```
Syntax :- public Void destroy()
```

Following fig represent a typical Servlet life Cycle Scenario.

⇒ First the http request Coming to the Server are Sent to the Servlet Container. The Servlet Container loads the Servlet before invoking the Service() method. Then the Servlet Container handles multiple requests by Sending multiple threads, each thread executing the Service () method of a Single instance of Servlet.

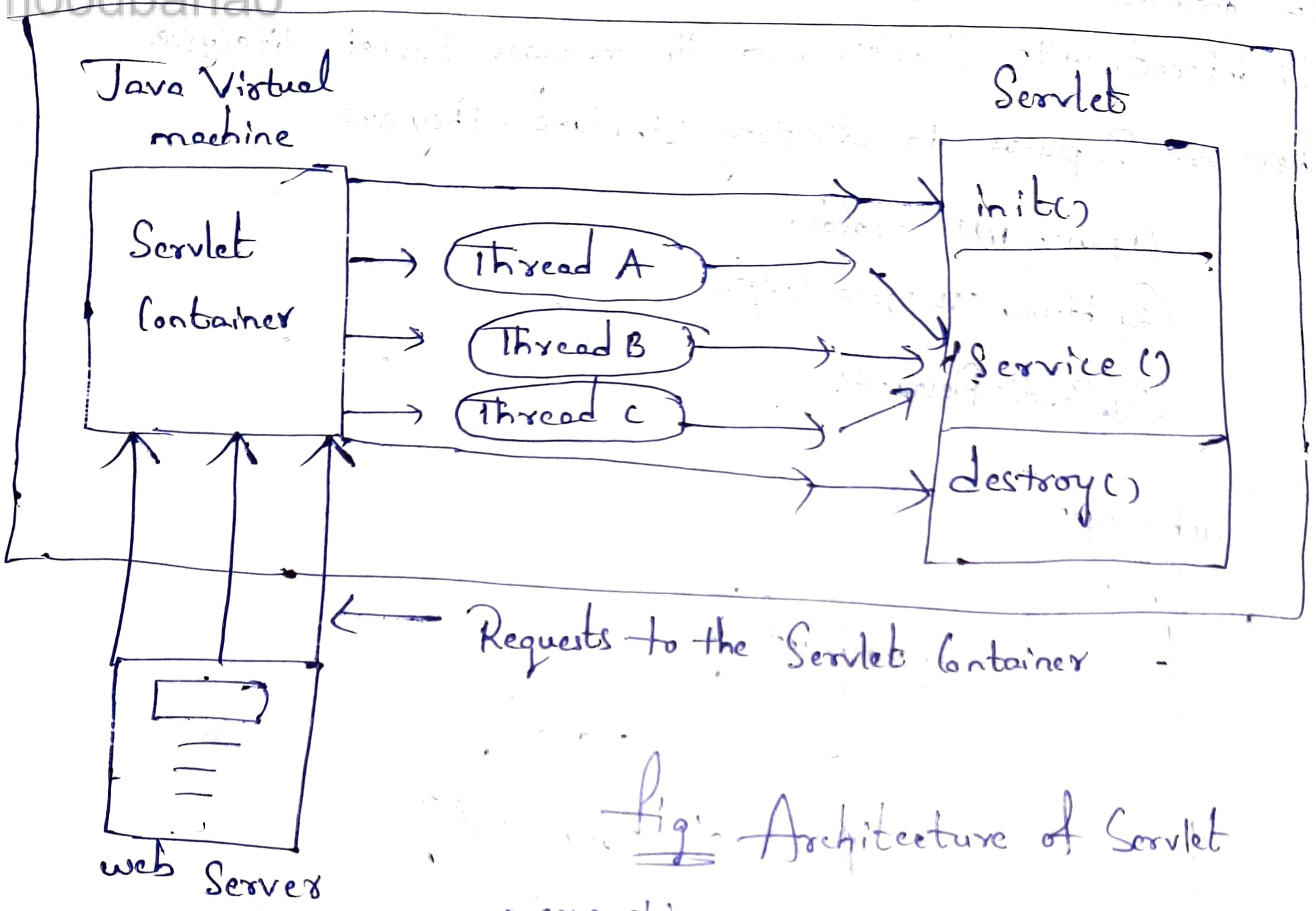


fig: Architecture of Servlet

DEPLOYING A SERVLET

executing

Servlet deployment Steps:- There are Six Steps in deploying a Servlet.

① Creating a directory Structure:-

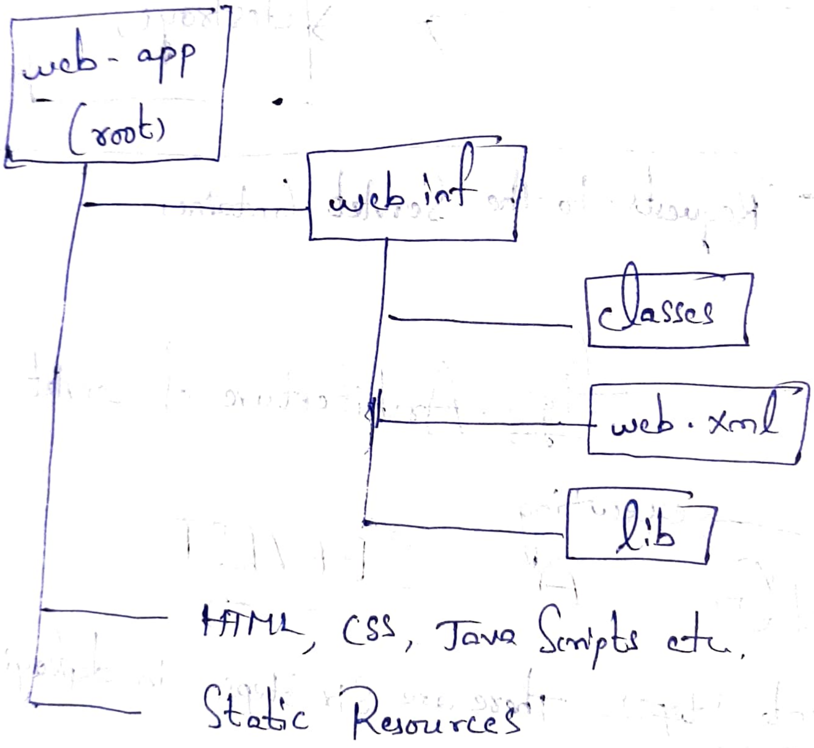
⇒ This directory Structure will define where to Store different kinds of files that are present in web Applications.

⇒ web Container will take information from the web Application and this web Container will respond to the client.

web Container is a Component of web Server. this Container will interact with Servlets and it manages Servlet lifecycle

there are 3 parts in directory Structure. they are

1. web INF folder
2. html, css, Java Scripts.
3. Static Resources.



② Creating a Servlet :-

we can Create Servlet in 3 ways

1. By implementing Servlet interface
2. By inheriting Generic Servlet class
3. By 〃 http Servlet class → this is most widely used

⇒ to handle http Request, Servlets will provide ~~two~~ some method Such as do get() method; do post() method etc. to Communicate with the client and Server

Syntax :- public class DemoServlet extends Http Servlet.

DemoServlet ↓ demo is name of Servlet we are extending Demo Servlet from http Servlet.

extends ↓ we are using extends because we are inheriting

3) Compiling a Servlet :-

inorder to Compile Servlet we use jar file with .jar extension (Java archive)

this jar file Can be loaded in 2 ways

① by Setting class path

② paste jar file in JRE/ lib/ ext (Java runtime environment)

Different Kinds of Servers uses different Kinds of Jar files

⇒ in Apache Tomcat Server → we use api.jar file

4) Creating a deployment descriptor:

this deployment descriptor is a Xml file (web.xml) It Contains all the information related to a Servlet.

it This Xml file contains <servlet>, <servlet-name>, <servlet-class
<url-pattern> etc.

$\downarrow$ name of Servlet

⇒ web Container uses parsers to get information from (web.xml) file

(5) Start Server and deploy the project.

(5) to Start Server Goto → $\boxed{\text{Apache tomcat/bin.directory}}$
and then click on $\boxed{\text{Startup.bat}}$. This is the process to Start the Server.

to deploy the project, Copy and paste the project in webapps folder under tomcat.

(6) Accessing the Servlet:

⇒ http: // hostname : portno / Contextroot / urlpattern

$\downarrow$ localhost   9999 $\downarrow$ demo

Tomcat Server, Difference between JDSK and Apache tomcat

, <Servlet class)

from

tory

ess to

webapps

Apache tomcat

⟹ Servlets are the best choice if we want to go for serious Seover side programming. These Servlets needs a Special environment In which they can be executed. For that we use Servlet Container. ie Tomcat.

⟹ Tomcat is open Source product. It is maintained by Jakarta project of Apache Software foundation. Tomcat is basically a Servlet Container that Contains the class libraries, documentation, runtime Support which is useful for executing and testing the Servlet.

For executing of Servlet following Software must be installed on Computer.

① JDK (Java development kit)

Servlets are basically Java files we should have JDK Installed in our Computer.

② Tomcat :- Tomcat is Servlet Container using which Servlet can be executed.

Features of Tomcat:

① reduces Garbage Collection

② It Improves performance & Scalability

(3) parses JSP efficiently
(4) platform changability.

⇒ Tomcat is an opensource web Server ~~and~~ ~~Servlet Contain~~ developed by Apache Software foundation (ASF).

Tomcat 4.x released with following features:

(5)

(1) Catalina:- It is Servlet Container. It helps to execute Servlets and JSP. when you Startup tomcat, you actually Startup Catalina.

(2) Coyote: It is a tomcat web Connector Component. It listen for Incoming Connectors to Server on Specific TCP port and forwards the request to tomcat engine to process the request and send back the response to client

(3) Jasper: It is basically a tomcat JSP engine. It parses the JSP files & Compile them to Java Code as Servlet. Jasper detect the changes to JSP files and recompiles them

JDSK:-

(Java development Standard Edition Kit)

It is available for Sun microSystem inorder to develop & deploy Java Applications on desktops and Servers. It is freely available in internet

JDSK	Apache tomcat
platform for developing the core Java Applications	It is web Server using which Server Side programs Such as Servlets or JSP's Can be executed
It is openSource product developed by Sun microSystem	Apache

To execute any Servlet or JSP we require both JDSK as well as Tomcat installed on our Computer.

SERVLET API

To represent interfaces and classes for Servlet api, we use two packages they are

① javax. Servlet

② javax. Servlet. http

→ The javax.Servlet package Contains many interfaces and classe
that are used by the Servlet or web Container. These are not
Specific to any protocol

→ The javax.Servlet.http package Contains interfaces and classes
that are responsible for http requests only

Interfaces in javax.Servlet package:

5

There are many interfaces in javax.Servlet package they are
as follows.

1. Servlet
2. Servlet Request
3. Servlet Response
4. RequestDispatcher
5. Servlet Config
6. Servlet Context
7. Single thread model
8. Filter
9. Filter Config
10. Filter chain
11. Servlet Request ~~Lst~~ Listener
12. " " Attribute listener
13. Servlet Context listener
14. Servlet Context Attribute listen

classes in javax.Servlet package

there are many classes in javax.Servlet package. They are

1. Generic Servlet
2. Servlet input Stream
3. Servlet output Stream
4. Servlet Request wrapper
5. Servlet Response wrapper
6. Servlet Request Event
7. Servlet Context Event
8. Servlet Request Attribute Event
9. Servlet Context Attribute Event
10. Servlet Exception
11. Unavailable Exception.

Interfaces in javax.Servlet.http package they are :-

1. Http Servlet Request
2. Http Servlet Response
3. Http Session
4. Http Session Listener
5. Http Session Attribute Listener
6. Http Session Binding Listener
7. Http Session Activation Listener
8. Http Session Context

classes in javax. Servlet. http package:

1. HttpServlet
2. Cookies
3. http Servlet Request wrapper
4. Http Servlet Response Wrapper
5. Http Session Event
6. Http Session Binding Event
7. Http Utils

Reading Servlet parameters:

⇒ ~~As we know~~, when you need to pass some information from your browser to web server and ultimately to your backend program. the browser uses two methods to pass this information to web server. these methods are

① GET method

② POST method

⇒ GET method is the default method to pass information from browser to web server. The GET method sends the encoded user information added to page request

⇒ POST method send information to backend program. It is same as get method except that the data which we have entered on form will appear on url.

Reading Form Data using Servlet

Servlet handles form data parsing automatically using the following methods depending on situation:

① getparameter()

you Call request.getparameter() method to get the value of a form parameter.

② getparametervalues()

you call this method if the parameter appears morethan once and returns multiple Values, for example:- checkbox.

③ getparameterNames()

you call this method if you want a Complete list of all parameters in the Current request.

index.html

Example:-

```
<form ~~action~~ name = "welcome" method = "get" >
      action= "http://localhost/servlet/welcome"
~~expr~~
Enter yourname <Input type = "text" name = "name" ><br>
<input type = "Submit" value = "login" >
</form>
```

DemoServ.java

```
import javax.Servlet.http.*;
import javax.Servlet.*;
import java.io.*;

public class DemoServ extends HttpServlet {
public void doGet (HttpServletRequest req, HttpServletResponse res)
throws ServletException, IO Exception
{
[
res.set ContentType ("text/html");
Printwriter pw = res.get writer ();

String name = req. get parameter ("name");
pw.println ("welcome" + name);

pw. close();
    }

}
```

in the above example, we are displaying the name of the user in the Servlet. For this purpose, we have used the get Parameter method that returns the value for the given request parameter name.

Reading Initialization Parameters:

most of the time, data (ex. admin email, database Username and password) need to be provided in production mode (clientchoi initialization parameters can reduce the Complexity and maintenance,

we can pass Some parameters to the Servlet using web. xml file. using <init-~~parameter~~ para> tag we can Specify name and value of parameters with the help of <para-name> and <paravalue> tags.

Example of Servlet Config to get initialization parameter

In this example, we are getting the one initialization parameter from the web.xml file and printing this information in Servlet

DemoServlet.java

```
import java.io.*;
import javax.Servlet.*;
import javax.Servlet.http.*;
public class Demo Servlet     extends HttpServlet
     {
   public void doGet( HttpServlet Request request,
                      Http Servlet Response response)
       throws ServletException, IoException
        {
   response.Set ContentType("text/html");
   printwriter out = response.getwriter();
   Servlet Config Config = get Servlet Config();
   String driver = Config get init parameter("driver");
out. print ("Driver is :" +driver);
out. close();
}

}
```

web. xml

```
<web-app>
<servlet>
<servlet-name> DemoServlet </servlet-name>
<Servlet-class> Demo Servlet </Servlet-class>

<init-param>
<param-name> driver </param-name>
<param-value>Sun.jdbc.odbc.jdbcodbc driver </param-value>
</init-param>
</servlet

<Servlet mapping>
<servlet-name> DemoServlet </servlet-name>
<url-pattern> /Servlet1 </url-pattern>
</Servlet mapping>
</web-app>
```

Handling Http Request & Responses:

We Know that the client can make the request to the web-Server using HTTP protocol. there is HTTP Servlet class in which there are Some Special methods which can be used to handle HTTP requests. These methods are.

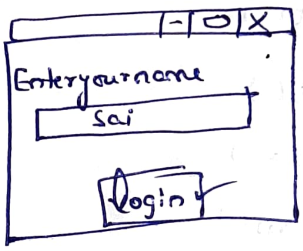
1. doDelete()
2. doGet()
3. doPost()
4. do Put()
5. do Head()
6. do Option()
7. do Trace()

For handling the input, the HTTP request makes use of two commonly used methods. Such as GET and POST In HTTP GET request the doGet method is used. In HTTP-Post request the doPost request method is used. when user Sumbits his request using doGet method then URL String displays the request Submitted by user. But if dopost method is used then URL String does not show the Submitted Content. For handling httpRequest and Response here we write two programs.

1. HTML
2. Servlet.

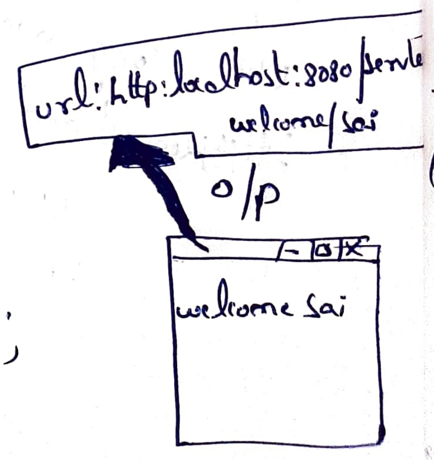
Colorbet.html

```
<html>
<body>
<center>
<form ~~[illegible] action~~ name = "welcome"      method = "get" action = http://localhost:8080/servlet/welcom
Enteryour name < input type = "text" name = "name"> <br>
        <input type = "Submit" value = "login">
    </form>
```



DemoServ.java

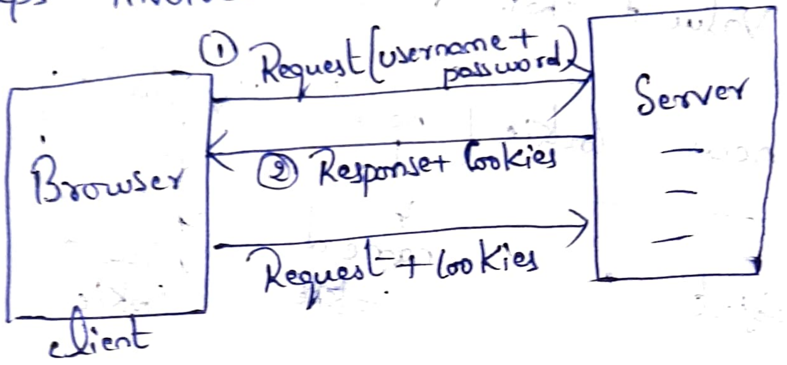
```
    import java.io.*;
    import javax.servlet.*;
    import javax.Servlet.http.*;

public class Demoserv extends HttpServlet {
public void doGet (HttpServletRequest request, HttpServlet Response respo
        throws ServletException, IoException
    {
res. set ContentType ("text/html");
printwriter pw = res.getwriter();
String name= req.getparameter ("name");
pw.println ("welcome" + name);
pw.close(); } }
```


url: http:localhost:8080/servle welcome/sai

Cookies in java Servlet:

- Cookies are the text file Stored on the client Computer. they are used for tracking purpose.

- Steps involved in identification



⇒ whenever user Send Request to Server by Entering username and password. then Server Respond back by Sending Cookies along with Response. when ever user ~~again~~ Send Request again to the Server, then Server Receive Request and as well as Cookies.

⇒ Cookies are classified into 2 types.

Non-persistent Cookie

Non persistent Cookies are valid for Single Session only. ie, Cookie is Removed each time when user closes the browser.

Persistent Cookie

Persistent Cookies are valid for multiple Session. ie, Cookies are not removed from the browser even user closes the browser

Steps to Set Cookies in java Servlet:

① Creating Cookie object:

inorder to Create Cookie object we call Cookie Constructor with Cookiename and Value of that Cookie"

```
Cookie c = new Cookie("name", "value");
```

c → name of Cookie object; "name" → name of Cookie; "value" → Value of Cookie

② Set maximum Age:

⇒ how long ~~a~~ Cookie should be Valid (in Seconds).

that Can be Achieved by using Set maxage method.

Syntax
```
c.SetmaxAge(60*60*24);
```

60 → 60min; 60 → 60seconds; 24 → 24 hours (1day)

60 * 60 * 24 = 86400 Seconds in a day.

③ Send Cookie into http response header:-

next we need to Send Cookie into http response header. So that it Can be Stored on client Side.

```
response.addCookie(c);
```

c → name of Cookie object

⇒ A Cookie has a name; Value, optional attributes Such as a Comment, path, domain, maximum age, Version number.

Advantages:

① Simple technique of maintaining the State
② Cookies are maintained at client Side

Disadvantages:

① It will not work if Cookie is disabled from the browser
② only textual information Can be Set in Cookie object.

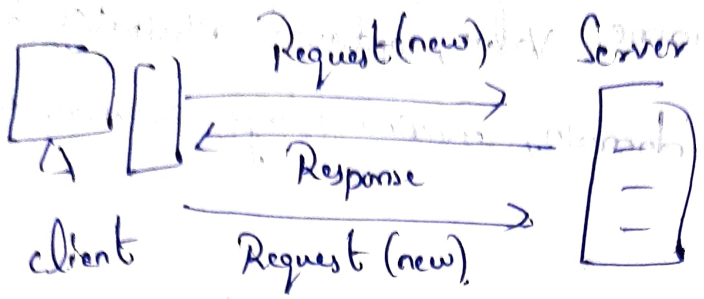
Sessions in java Servlet:

⇒ Session Simply means a particular Interval of time.

⇒ Session tracking is a way to maintain State(data) of an user. It is also Known as Session management in Servlet.

⇒ Http protocol is a Stateless So we need to maintain State using Session tracking technique. Each time user requests to the Server, Server treats the request as the new request. So we need to ~~maint~~ maintain the State of an user to recognize to particular user.

⇒ http is a Stateless that means each request is Considered as the new request. it is shown in fig.



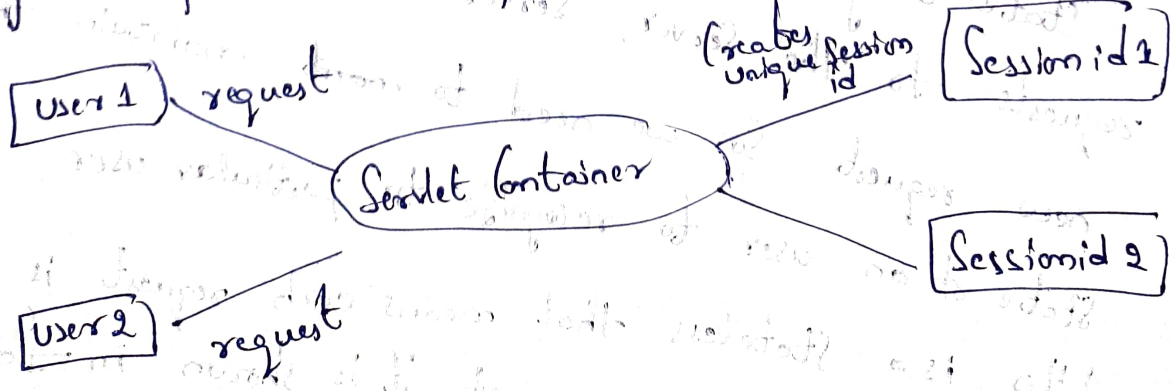
Inorder to achieve Session tracking in Servlets, Cookies have been one of the most commonly used technique. however, they have the following disadvantages.

① they Can Keep only textual information

② they are browser dependent. Hence, if client disable them, your web application Cant make use of them

③ Individual Cookie Can Contain not morethan 4 kb of information

⇒ we Can Create Sessions with a unique Session id for each user in java Servlet. For this, Servlets provide an interface called "Http Session interface". the following diagram explain how http Sessions work in Servlet.



Advantages of http Sessions in Servlet:

① Any kinds of object can be stored into a Session, be it a text, database, dataset etc..

② Usage of Sessions is not dependent on clients browser.

③ Sessions are Secure and transparent

Disadvantages of http Session:

① performance overhead due to Session object being Stored on Server.

② overhead due to Serialization and deserialization of data

there are four techniques used in Session tracking

① Cookies
② hidden form field
③ URL Rewriting
④ http Session object

① Cookies:

A web Server Can assign a unique Session ID as a Cookie to each client and for Sub Sequent requests from the client they Can be recognized using the received Cookie.

② Hidden form fields:

A web Server Can Send a hidden HTML form field along with a Unique Session ID as follows

<input type = " hidden " name = " Sessionid " value = " 12345 "

the entire meaning is that, when the form is Submitted, the Specific name and Value are automatically included in the GET and POST data. Each time when web browser Sends request back, then Session_id Value Can be used to keep the track of different web browsers.

③ URL Rewriting:

you Can add Some extra data on the end of URL, that identifies the Session.

Example: https://www. . . . . . Com/file.htm; Session id=123

Sessionid to identify us

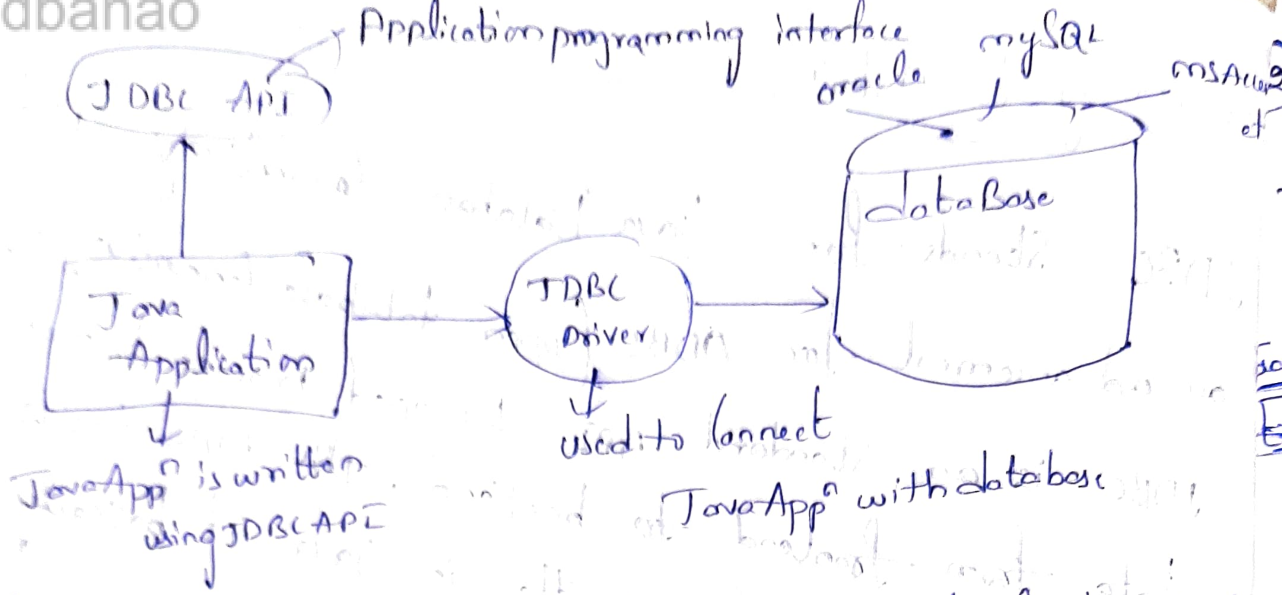
④ Http Session object:

Servlet provides Http Session interface which provides a way to identify a user across more than on page request or visit to a website and to Store information about user.

Connecting to DataBase using JDBC

JDBC Stands for java Database Connectivity. It's
an advancement for ODBC (open database Connectivity).
JDBC is an Standard API developed inorder to move
data from frontend to backend. this API Consists
of classes and interfaces written in java. It basically
Acts as an interface or channel ~~or channel~~ between your
Java program and databases. i.e it establishes a link between
the ~~two So that a prog~~ Java program and database
So that the programmer Could Send data from java Code
and Store it in the database for future use.

⟹ JDBC Came into Existence because, ODBC being platform
dependent had a lot of drawbacks. ODBC API was
written in C, C++, python, Core java, as we know ~~abov~~,
these languages (except java and python) are platform
dependent. therefore to remove dependence, JDBC was
developed by a database Vendor which Consisted
of Classes and interfaces written in java.



Application programming interface
JDBC API
oracle
mySQL
msAccess
etc
dataBase
Java Application
JDBC Driver
JavaAppⁿ is written using JDBC API
used to connect JavaAppⁿ with database

there are 5 steps to connect any java Application with the database using JDBC.

1. Register the Driver class
2. Create Connection
3. Create Statement
4. Execute Queries
5. Close Connection

① Register the driver class.

the forname() method of class class is used to register the driver class. this method is used to dynamically load the driver class.

$\boxed{\text{Syntax: public static void forName( )}}$

2) Create the Connection object:-

the get Connection() method of Driver manager class is used to establish Connection with the database

ample:-

~~example~~ Connection con = DriverManager. get Connection ("path", "username", "password");

Connection → ~~object~~ class Connection

con → objects

3) Create Statement object:-

the CreateStatement() method of Connection interface is used to Create Statement. the object of Statement is responsible to execute queries with database.

Statement Stmt = Con. Create Statement ( ) ;

Stmt → Statement name

Con → Connection

4) Executing the Query:-

the execute Query ( ) method of Statement interface is used to execute queries to the data base. this method returns the object of Result Set that Can be used to get all the records of a tables

Resultset rs = Stmt.executeQuery ("select * from Emp");

~~while~~

rs → name of Result Set

Stmt → Statement

Emp → display table of Employees

(5) ~~Close the Connection object~~:

```
while ( rs.next()
        rs.getint(), rs.getString();
```

(5) Close the Connection:

By closing Connection object Statement and Result Set will be closed automatically. The close() method of Connection interface is used to close the Connection

$$\begin{tikzcd} \boxed{\text{Con. close();}} \arrow[d] \\ \text{Connection.} \end{tikzcd}$$