Sep 21 Intro

Adv.java (Before adv.java :: u r a website visitor , After adv.java :: u r a website developer)

|-->pre-requisite:: Core Java oops

|--> Parts of adv.java
- a) JDBC and Misc [pre-requisite: Core Java oops]
- b) Servlet and Misc [pre-requisite: Core Java oops]
- c) Jsp and Misc [pre-requisite: servlet]
- d) Mini Projects [pre-requisite: JDBC, servlet, jsp]

Some other slot :: 9 am/ 4pm/6pm

11 am slot

Free Topics :: Html , Java script ,css
15+ Java real time tools like git, svn,maven,gradle and etc..
10+ common topics like resume preparation, interview tips ,...
Hibernate with JPA

Naresh IT JOB program :: Clear screening test conducted by Naresh IT ..Placing u in the company is the resposibility of Naresh IT

duration of course :: 100+ sessions
course fee :: only adv.java + free topics :: Rs.3500.0
course fee :: only adv.java + free topics +hibernate with JPA :: Rs.5000.0

=>In java servlet,jsp technologies are given to develop the websites / web applications.

=> A web application is collection of web comps(resuable program/file) having capability to generate webpages..

=>Two types of webpages

a)static web pages
=> The content of the web page remains same for all the requests
eg:: aobut us page, contact us page , terms and conditions page..

b) Dynamic webpages
=>These content of the webpage will change based on time of request generation or based on the input values of the request.

eg:: gmail inbox , Live game score page, Socket market share values page , weather report page and etc..

Find out gmail login page is static web page or dynamic webpage?
Ans) It is dynamic webpage becoz its content changes dynamically when given wroing username or passwrd.

3 types web comps based on the kind of web pages they generate
==========

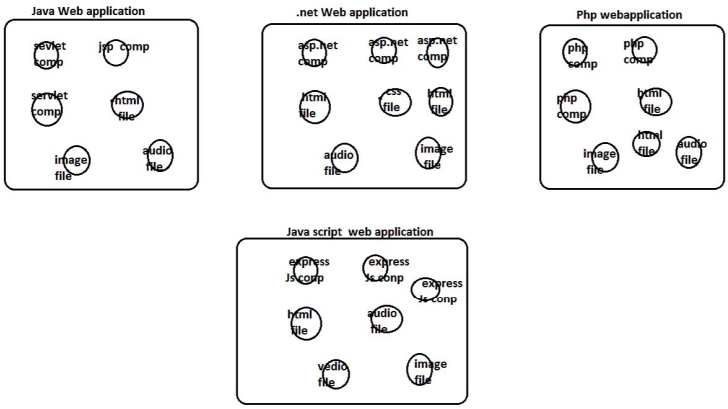
a) static web comps (Generate static webpages)
eg: html files

b) dynamic web comps (Generate dynamic webpages)
eg: servlet comps,jsp comps, asp.net comps, php comps ,express Js comps and etc..

c) Helper comps (These comps do not generate webpages, but they act heper comps to other static and dynamic web comps)
eg:: image files, audio files , audio and vedio files and etc..

In All technologies based websistes development the static web comps and helper comps are same.. only the dynamic webcomps will be changed based on the technologies we use.

=>In java webapplication the dynamic webcomps are servlet,jsp comps
=>In .net webapplication the dynamic webcomps are asp.net comps
=>In php webapplication the dynamic webcomps are php comps
=>In Javascript webapplication the dynamic webcomps are expressJs comps



Need of web server software
==========

=>The standalone App is specific to one computer and allows only one user at a time operate the application.. so manual execution of the application is very much sufficient.
=> Once develop the web application or website , it will be requested by thousands or lacs of users 24/7 simultaenously.. So the manual execution of web comps in web application when they are requested is pratical impossible for those huge no.of simultaenous requests.. if we do so .. that will be error prone process. So we need one special software to automate the web application and its web comps execution that is nothing but web server s/w.

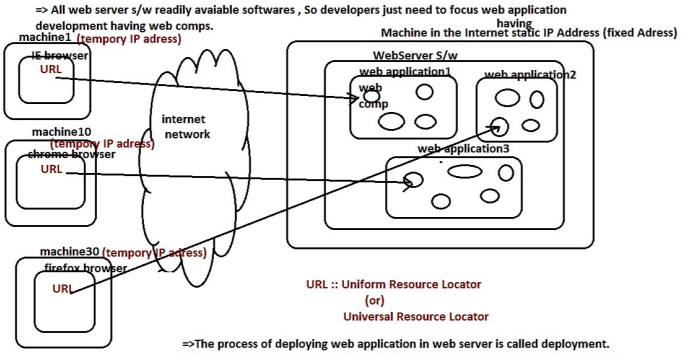
=>WebServer is a piece of server s/w that listens client requests continuosly, takes the requests, maps/links the requests appropriate web application and web comps, executes the web comps dynamically and collects results from web comps , sends to clients (browser) as responses..

eg:: Tomcat , Resin , jetty and etc.. (java env..)
IIS , PWS (.net env..)
AWS , wamp server (php env..)
Nodes JS (java script env..)

IIS :: Internet Information server
AWS :: Apache webServer
PWS: Personal web server

=>Standlone Apps execution takes place manually.. where as the the web application execution takes in automated env.. with the support web server software.

=> All web server s/w readily avaiable softwares , So developers just need to focus web application having development having web comps.



URL :: Uniform Resource Locator
(or)
Universal Resource Locator

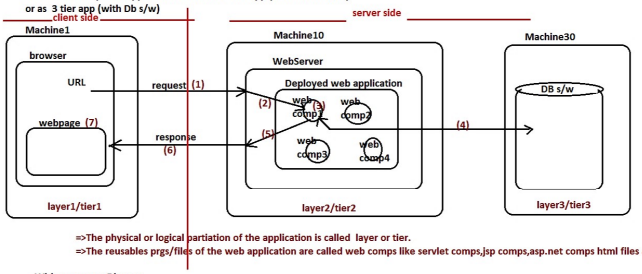
=>The process of deploying web application in web server is called deployment.
=>The process of removing web application in web server is called undeployment.
=>In One webSErver ,we can deploy any number of web applicaitons..

=> We must choose Dyamic web comps of the web application and web server s/w having the compitbility.
=> For example In java web applications the servlet,jsp web comps are java based , So we must choose java web server like Tomcat
=> For example In .net web applications the asp.net web comps are .net based , So we must choose .net web server like IIS

Sep 22 Web Application setup

Setup required for the web application development and execution (Generic Setup)

=>we can develop web application either as 2- tier app (with out DB s/w)
or as 3 tier app (with Db s/w)



=>The physical or logical partiation of the application is called layer or tier.
=>The reusables prgs/files of the web application are called web comps like servlet comps,jsp comps,asp.net comps html files and etc..

With respect to Diagram
==========
(1) Enduser give request to web application by typing URL in the browser address bar
(2)WebServer takes the request and maps/links the request with web comp of the web application based on the mapped URL
(3) web comp process the request by executing the logics
(4)web comp interacts with Db s/w if neccessary
(5) The results/outputs generated by the web comp comes to web server
(6) web server sends the outputs to browser as response

(7) browser recieves the response and displays it as the web page.

=>To execute html code we need html interpeter (part of browser s/w)
=> To execute java script code we need java script engine (part of browser s/w)
=> To execute java code we need JRE/JVM directly or indirectly.
note:: All webServer of java internally runs on the JRE/JVM

Two types of web comps based on place where they execute

(a) Server side web comps
=>These only reside in the webserver .. they also execute in web server
when they are requested from browser s/ws (clients)
eg:: servlet comp,jsp comps, asp.net comps, php comps and etc..

(b) Client side web comps
=>These comps's code goes to browser for execution .. wheen they are requested from the browser s/ws (clients)
eg:: -html files , java script files and etc.

=>Do not decide wheather web comp is client side for server side web comp based on the place where it is residing .. decide it based on the place where it is executing..

=>images , audio files, video files and etc. are the helper web comps for both client side and server side web comps..
=>Generally static web comps are client side web comps (eg:: html file)
and dynamic web comps are server side web comps (eg: servlet,jsp, asp.net ,php and etc..comps)

=>In the development and execution web application we need
=> browser s/w
=>web server s/w
=> Server side web technologies
=> Client side web technologies
=> DB s/w

List of browser softwares (Acts as web clients or user-agents for enduser to send requests and to recieve responses)
==========
InternetExplorer ---> from MicroSoft (2)
Microsoft Edge ---> from MicroSoft
Google chrome ---> from Google (1)
Mozilla firefox ----> from Mozilla
Opera ----------> from opera soft
safari -----> from apple (3)
Hot java ----> Red Hat
and etc..

List of webServer s/ws

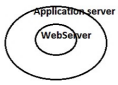
(To auomate the execution of web application and its web comps)
F-->open soruce /free
C--> commercial

apache tomcat ----------> from apache (java based) (1) (F)
JWS (Java Web Sever) ----> from Sun Ms (java based) (F)
Jetty ----------> from Eclipse (java based) (2) (F)
IIS (Internet Information server) ----> from microsoft (.net based) (C)
PWS (Personal webServer) ----> from microsoft (.net based) (C)
Apache Web server -----> from Apache (PHP based) (F)
NodeJS -----> from netscape (JAva script server env..) (F)
+ OpenJS fundation
Resin server ----> from Resin soft (java based) (3) (C)
Undertow server -----> from Jboss (RedHat) (java based) (F)
and etc..

note:: if want to execute java script outside the browser then we need to java NodeJS setup.

note:: Here ratings are given only for java based web servers

Application server = webServer++ (web server and many other features)



note:: All applications severs java based.

Different Application server softwares
==========
=>weblogic from BEA systems /oracle corp (C)
=> websphere from IBM (C)
=> Jboss from Apache/RedHat (F)
=>Wildfly from RedHat (1) (F)
=>GlassFish from Sun Ms/oracle corp (2) (F)
and etc..

note:: we can automate the execution of java web application either using java based web server like Tomcat, jetty and etc.. or by using Application servers like weblogic, websphere, jboss and etc..

note:: Upto Tomcat 6 is called called web server.. From Tomcat7 onwords it can be called as Application server becoz more facilities added to Tomcat from 7 version.

List of server side web technologies
==========
=>To Develop the server web comps which executes in webServer /Applicaiton server itself

servlet ----------> from Sun Ms /oracle corp (java based) (1)
JSP (java server pages) ----> from Sun Ms/oracle corp (java based) (1)
asp.net ----------> from microsoft (2)
php -----> from apache (4)
expressJs ---> from TJ Holowaychuk, StrongLoop (3)
and etc...

note:: we must choose webServer /Application server having compitability with server side web technology.

In web applications java script can be used in two angels
=>Basic Java script as client side web technology becoz its goes browser and executed by JAva script engine of browser s/w JS
=> advanced JAva scropt like express as severside web technology which can execute only in NodeJS server env..
ExpressJS code is alternate to servlet,jsp, asp.net comps..

List of Client side web technologies
==========
=>Given to develop Client side web comps whose code goes browser for execution

=>html,css, java script , bootstrap, ajax , angularJS, anglur, jquery, reactjs and etc..

css :: cascading style sheets
ajax :: asynchronous java script and xml

note:: The client side technologies of java script like basic java script, ajax, jquery , angularjs and etc.. can be used in JAva script web application and also java ,.net and php web application.

NodeJS ---> Java script based web server
ExpressJS --> Java script based server web technology
Java script, ajax,angular, angularjs,jqyery are java script based clietn side web technologies.

List of DB s/ws
==========
a) Oracle from oracle corp (1)
b) mysql from devx (2)
c) postgreSQL from enterprise DB
d) DB /2 from IBM
and etc..

note:: browser s/ws , Db s/ws, client side web technologies compitable to use in all domains based web applications like java, .net,php,java script..
note:: we must choos web server or Application sever and server side techologies having compitiblity.

Sep 23 ServletCotnainer and Jsp Container

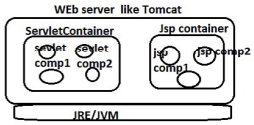
ServletContainer & Jsp Container
==========

=>To execute html code we need Html interpeter
=> To execute java script code , we need Java Script engine

=>To execute Standalone Java App , we need JRE/JVM

=>Similarly , To exects servlet comps , we need Servletcontainer and Jsp comps we need Jsp container
note:: Servlet container and Jsp Container internally uses JRE/JVM becoz they are also Java programs executing the other java comps called servlet comps, jsp comps.

=>We need to not develop ServletContainer and jsp container becoz once we install Java based web server like Tomact they come as built-in container.



=>The requested servlet comps go to servlet container for execution automatically.
=>The requested jsp comps go to jspcontainer container for execution automatically.

=> All statements must be executed by using JVM directly or indirectly.. the java code of servlet comps/jsp comps will also executed by JVM indirectly through Servletcontainer/Jsp Container.

=> A container is software prg that takes care the whole life cycle of given comp/resouce from birth to death (generally object creation to object destruction) .. It actually takes care of the following activities automatically

a) Loading class
b) creating object
c) Initilaizing object
d) managing object
e) destroyring the object

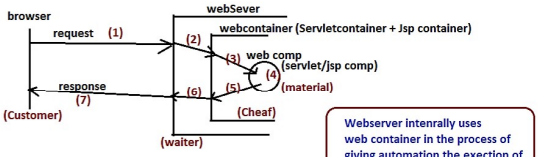
=>We can say the container is like an aquirium managing the life cycle of given comps /resources called fishes.

WebContainer = ServletContainer + Jsp Container

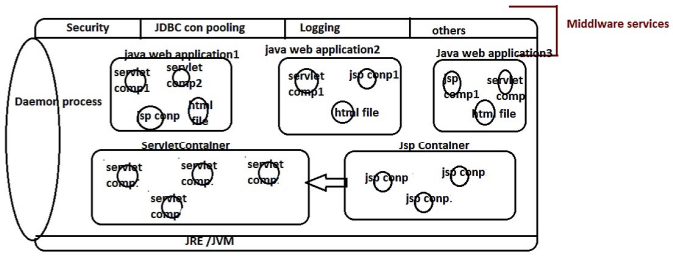
What is the difference WebServer and Web Container?

=>Web server is mainly responsible to take requests from clients by listening to requeusts continusoly and also responsible to deliver the generated outputs back to clients/broswsers as response.

=>webContainer is reposible to map the request to appriate web comps of web application and to execute them dynamically by managing their life cycle



High Level architecture of Java based web server s/w like Tomcat
==========



=>Once the web server is started , One daemon process will created indicating that server is running continuosly to take requests given by clients by waiting for requests.
=>The process that will be running continuosly either foreground or in background is called Daemon process

=>Some web servers give built-in JRE/JVM along wit the web server s/w installation.. (eg:: weblogic, websphere, ...) where as some other servers like Tomcat will use the JRE/JVM by collecting from active JDK installed the computer.

=> The addtional ,optional and configurable services that can be applied on the deployed web applications of web server to make our web application more perfect and accurate are called middlware services..
=> Middleware services are not minimun logics of application devleopment.. they are addtional and configurable services to make our Apps as the more perfect and accurate applications.

eg:: Security :: Protects the Application
eg:: JDBC con pooling:: Gives the reusablity of JDBC con objects
eg: Logging :: keeps track of the Application's flow.

=>Since middlware services ready made services offered by webserver s/w .. so they can applied on multiple deployed applications..

=>The request given to servlet comps , makes the servlet comps going to servlet container for execution
=>The request given to jsp comps , makes the jsp comps going to jsp container for execution
note:: JspContainer internally uses ServletContainer becoz very jsp comp will be translated servlet comp as part of execution.
=>The requests given html files , js comps makes the servlet container sending their code to browser for execution becoz they are client side web comps..
=> we cannot give direct requests to helper web comps like images, audio files and etc.. i.e they will be used by static and dynamic web comps internally.

Responsibilties of java based web server
==========

a) Listening to client requests continuosly by having daemon process
b) Providing middlware services
c) Providing servlet container, jsp container (web container)
d) Providing its own JRE/JVM or using using system supplied JRE/JVM
e) Admin console screen to perform deployment , undeployment of web applications
f) Admin console screen to perform start , stop , reload (stop+start) activities on the deployed web application.
g) Delivering responses to browsers (clients) after getting results from web comps execution
h) Recieving requests and handovering the requests to web container.

and etc..

Responsibilites of Web container in Java based web server
==========

=> Taking requests given by web server and mapping /linking them to appropriate web comps of the deployed web applications
=> Executing server side web comps (serlvet,jsp comps) by managing their life cycle
=> sending the code client side web comps (html,js comps) to browser for execution
=> gives its own garbage collector to destroy and clean the objects
=> performing jsp comps to servlet comp translation
=> gathering results from server side web comps execution and giving them to web server
=> providing env.. from inter communication b/w web comps like servlet to servlet , jsp to jsp, servlet to jsp and etc.. communicaitons
and etc..

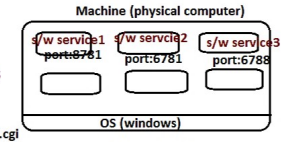
Tomcat server
==========

Sep 23 Tomcat server

type :: java based web server (upto 6 version) / java based app server (from 7 version)
version : 10.x (latest) / 9.x (running) (compitable with jdk1.8+)
vendor :: Apache
Open Source s/w (free ware)
default port number :: 8080
(changable)
creator :: Mr. David

ServletContainer name :: CATALINA JAMES
Jsp Container name :: JASPER

To download tomcat sever s/w::
https://tomcat.apache.org/download-10.cgi
(download setup file --service installer)



=>computer provides physical ports to connect hardware devices
eg:: USB port , monitor port , priter port and etc..

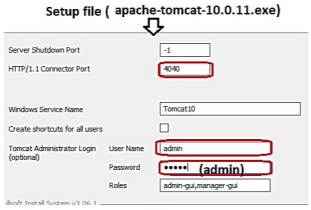
=>The OS of computer uses managers different services on different logical software ports .. Every software port is identified with its port number .

Windows OS is having total 65,535 software ports ..
In that 1 to 1024 are reserved from Windows services like task manager and etc..
The remaining software ports will be for the externally installed softwares like Tomcat, weblogic, oracle ,mysql and etc..

s/w	default port number
Tomcat	8080
weblogic	7001
oracle	1521
mysql	3306

=>During the installation of Tomcat server we need to use choose the following detals
=> Http port number :: 4040
=> Admin username(admin), password(admin)
=> <Tomcat_home> (Tomcat installation folder)
=> JRE folder location

Setup file (apache-tomcat-10.0.11.exe)



<Tomcat_home>

To start Tomcat server
==========

use <Tomcat_home>\bin\tomcat10.exe file (Internally creates Daemon process to listen to client requests continuosly)

To see the home page of Tomcat server
==========

Type this URL in browser address bar
---

http://localhost:4040
protocol host name and the port number of Tocmat server in the current computer.

To stop Tomcat server
==========

=>Use "crtrl+C" or close the server console window.

To stop the enabled automcatic startup type on Tomcat server
==========

go service.msc --->search for apache tomcat server -->right click properties ---> change the startup type to "manual".

Tell me , wheather Tomcat is web server or web container?

Ans) Tomcat is a server having servlet container (CATALINE JAMES), jsp container (Jasper)

Tomcat is web server upto version 6
Tomcat is Appicaiton server from version 7

from Tomcat console

from Tomcat home page

Tomcat documentation

Importanat modules of Java
==========

a)JSE module (installable module as jdk s/w) → Core Java Course ++
b) JEE (Not a installable module ,,, So we need to arrange the relevant impl class called web server or App server sotfware. → Adv java ++
c) JME (installable mdodule as as MDK s/w)
(out date module becoz of IOS, Andriod OS)

Since JEE not is not installable module , So we arrange the JEE module based webServer s/w (like Tomcat) or Application server s/w (like GalssFish) in our system tow work woth JEE module

Application server= webserver++
https :: http over SSL
http: Hyper Text Transfer protocol
SSL :Secured Socket Layer.

=>Working with JDK s/w is nothing but working with JSE module
=>Working with Web server or Applicaiton server s/w is nothing but working with JEE module
=>Working with MDK s/w is nothing but working with JME module (outdated)

JSE :: Java standard edition
JEE :: JAva enterprise edition / (Jarkarta Enterprise Endition)
JME :: Java Micro Endition

=> JSE module contains Java language + few technologies like JDBC, JNDI ,RMI and etc..

JNDI :: Java Naming And Directory Interface
RMI :: Remote Method Invocation

=>JEE mpodule contains only Technologies like servlet, jsp, EJB, JMS,JTA, JAAS,Java mail and etc.

JSP :: Java server pages
EJB :: enterprise Java beans (outdated)
JMS :: Java Messaging service
JTA :: Java Transaction API
JAAS :: Java Authenticaiton and Authorization service

=>spring,hiberante, JSF,Struts are etc.. Java frameworks and they are not part any Java Modules like JSE ,JEE

To see all port numbers that are in use
cmd> netstat -a (open cmd prompt as admin)

Sep 25 RowSets

RowSets (These are extension JDBC ResultSet objs)

=>JDBC ResultSet are having the fillowing limitations

a) ResultSet obj is not searizable to send over the network

b) Does not support Bean style programming (setter and getters methods based programming)

c) JDBC ResultSEt is connected object i.e with out having conection with Db s/w we can not manipulate DB s/w data. (offline processing is not possible)

=>To overcome these problem take the RowSets

=> RowSet object is a object of a java class that implements javax.sql.RowSet(I) which is the sub interface of java.sql.ResultSet(I).

There are 5 types of RowSets

a) JDBCRowSeet
b) CacheRowSet
c) WEbRowSet
d) JoinRowSet
e) FilteredRowSet

Popularly used RowSet.

advantages of RowSets

a) These are serializable objects, i.e we can send them over the network
b) Supports Bean style progrramming
c) These Disconnected objects .. i.e supports offilne Programming.

Oracle thi driver supports all the 5 types RowSet by different classes

OracleJdbcRowSet
OracleCacheRowSet =>Rowsets are very useful while developing PDA (Personal Digital Assitants) like tab
OracleWebRowSet carried flikpart delivey boy..
OracleJoinRowSet
OracleFilteredRowSet

JDBCRowset

=> It is like ResultSet obj in RowSet model
=> It is a Connected object
=> It is not a serializable object
=>Supports Bean style programming.

```
//JdbcRowSetDemo.java
package com.nt.rowset;

import java.sql.SQLException;

import oracle.jdbc.rowset.OracleJDBCRowSet;

public class JdbcRowSetDemo {

	public static void main(String[] args) {
		try(OracleJDBCRowSet jrowset=new OracleJDBCRowSet()){
			jrowset.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
			jrowset.setUsername("system");
			jrowset.setPassword("manager");
			jrowset.setCommand("SELECT * FROM DEPT");
			jrowset.execute(); //To execute the SQL query
			while(jrowset.next()) {
				System.out.println(jrowset.getInt(1)+"  "+jrowset.getString(2)+"  "+jrowset.getString(3));
			}//while

		}//try
		catch(SQLException se) {
			se.printStackTrace();
		}
		catch(Exception e) {
			e.printStackTrace();
		}
	}

}
```

Bean style programming (working with setter and getter methods)

CachedRowset

=> It is Serializable object, So its data can be sent over the network
=> It is DisConnected object i.e with out having connection with DB s/w .. we can modify RowSet object data.. and we can sync changes back to DB s/w once the connection is established
=> Supports Bean style programming..

```
//CachedRowSetDemo.java
package com.nt.rowset;

import java.sql.SQLException;

import oracle.jdbc.rowset.OracleCachedRowSet;

public class CachedRowSetDemo {

	public static void main(String[] args) {
		try(OracleCachedRowSet jrowset=new OracleCachedRowSet()){
			jrowset.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
			jrowset.setUsername("system");
			jrowset.setPassword("manager");
			jrowset.setCommand("SELECT * FROM DEPT");
			jrowset.execute(); //To execute the SQL query
			while(jrowset.next()) {
				System.out.println(jrowset.getInt(1)+"  "+jrowset.getString(2)+"  "+jrowset.getString(3));
			}//while
			System.out.println("Stop Db s/w during this sleep time");
			Thread.sleep(40000);
			// modification in offline mode/ disconnected mode
			jrowset.absolute(2);
			jrowset.updateString(3,"delhi123");
			jrowset.updateRow();
			System.out.println("Start the Db s/w during the sleep time");
			Thread.sleep(70000);
			jrowset.acceptChanges();
			while(jrowset.next()) {
				System.out.println(jrowset.getInt(1)+"  "+jrowset.getString(2)+"  "+jrowset.getString(3));
			}//while
		}//try
		catch(SQLException se) {
			se.printStackTrace();
		}
		catch(Exception e) {
			e.printStackTrace();
		}
	}

}
```

WebRowset

=>same as CachedRowSet but can render db table records xml file content
=>We can db table records over the network as xml file from one DB s/w to another Db s/w or from one technology project to another technology Project becoz xml data is neautral text data...

```
//WebRowSetDemo.java
package com.nt.rowset;

import java.io.FileWriter;
import java.sql.SQLException;

import oracle.jdbc.rowset.OracleWebRowSet;

public class WebRowSetDemo {

	public static void main(String[] args) {
		try(OracleWebRowSet wrowset=new OracleWebRowSet()){
			wrowset.setUrl("jdbc:oracle:thin:@localhost:1521:xe");
			wrowset.setUsername("system");
			wrowset.setPassword("manager");
			wrowset.setCommand("SELECT * FROM DEPT");
			wrowset.execute(); //To execute the SQL query
			while(wrowset.next()) {
				System.out.println(wrowset.getInt(1)+"  "+wrowset.getString(2)+"  "+wrowset.getString(3));
			}//while

			//write db table recrods to xml file
			FileWriter writer=new FileWriter("dept.xml");
			wrowset.writeXml(writer);
			System.out.println("Db table records are written dept.xml file ");


		}//try
		catch(SQLException se) {
			se.printStackTrace();
		}
		catch(Exception e) {
			e.printStackTrace();
		}
	}

}
```

Sep 27 Tomcat server

Understanding the Directory structure of Tomcat server after installation

```
E:\Tomcat 10.x (<Tomcat_home> directory)
   |------>bin
      |-----> *.exe (tomcat10.exe, tomcat10w.exe,....)
   |-------->webapps
      |------>Dir1,Dir2,Dir3,....        | Deployed
      |----->.war file1, .war file2, .war file3 ,....  | web applications.
   |-----> logs
      |--------->*.txt  (every day 1 log file ..keeping track of that day's activities)
   |---->conf
      |---------------->*.xml file  ( server.xml , tomcat-users.xml , Context.xml ,....)
   |---->temp
      |--->
   |----->catalina
      |----->
   |---->lib
      |----->*.jar  ( servlet-api.jar , jsp-api.jar , catalina.jar , jasper.jar ,....)
```

=>We can deploy web application in tomcat server either as directories or as the war file
war file :: web application archieve.

Deployment :: Keeping web applications in webSErver
Undeployment :: Removing the web applications from the webServer.

=><Tomcat_home>\lib folder gives libraries in the form jar files..

servlet-api.jar ---> represents servlet api (pkgs having classes,interfaces,annotations ,enums related to servlet programming)
jsp-api.jar ----> represents jsp api
jasper.jar ----> represents JspContainer/Jsp Engine
catalina.jar ----> represents ServletContainer/Servlet Engine

=> Servlet,jsp are not part of JDK s/w (not part of JSE module)..So we will not servlet api ,jsp api in the Jdk s/w..
=> Servlet,jsp are JEE module technologies and to work with JEE module we need to use Web Server or Applicaiton server s/w , So the web server or Applicaiton server s/w installation given gives jar files representing servlet api,jsp api.

=>In Tomcat server servlet-api.jar represents servlet api
=>In Tomcat server jsp-api.jar represents jsp api

API (Applicaiton Programming interface)
=>It is base for the programmers to develop s/w Apps using languages, technologies and frameworks

=> In "C" language api means set of functions which comes as header files
=> In "C++" language api means set of functions,classes which come as header files
=> In "Java" language api means set of classes, interfaces, enums and annoations which come as java packages..

java apis examples
---
lang api ==> java.lang and its sub pkgs
utility api ==> java.util and its sub pkgs
jdbc api ===> java.sql, javax.sql and thier sub pkgs
servlet api ==> javax.servlet and its sub pkgs
jsp api ===>javax.servlet.jsp and its sub pkgs.
and etc...

In Java Apis are also called as Libaries and they will be given in the form of jar files...

jar file :: Java archieve file
(java level zip file/rar file)

=>upto JEE 7 , the module name is Java enterprise edition the api pkg naming starts with javax.
=>from JEE 8 , the module name is Jakarta enterprise edition the api pkg naming starts with jakarta

servlet api pkgs upto JEE7	servlet api pkgs from JEE8
javax.servet	jakarta.servet
javax.serlvet.http	jakarta.serlvet.http
javax.servlet.annotation	jakarta.servlet.annotation
javax.servlet.descriptor	jakarta.servlet.descriptor

=>In the programming servlet,jsp comps we take supports serlvet,jsp apis..

jsp api pkgs upto JEE7	jsp api pkgs from JEE8
javax.servlet.jsp	jakarta.servlet.jsp
javax.servlet.jsp.el	jakarta.servlet.jsp.el
javax.servlet.jsp.tagext	jakarta.servlet.jsp.tagext

note:: Tomcat 9 server is given based JEE7 version rules and guidelines.
note:: Tomcat 10 server is given based JEE8 version rules and guidelines
note:: JEE is not a install software .. it is just speficication.. So installing one or other web Server or Applicaiton server s/w like Tomcat and working with the server is nothing but working with JEE module.

How to View and change Tomcat serer port after installation ?

=> Go to <Tomcat_home>\conf\server.xml file and modify port attribute value of first <Connector> tag
==> restart the server .

```
<Connector port="3535" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

Testing URL :: http://localhost:3535
protocol — Host name and port number of Tomcat server

=>To refer a computer being from that computer , we use "localhost" whose ip address 127.0.0.1

(or)

http://127.0.0.1:3535
IP address of Localhost

Q) How to view and modify admin user userame, pasword details after installing Tomcat server

Ans) Go to <Tomcat_home>\conf\tomcat-users.xml file and modify <user> tag atributes values --> restart the server

```
<user username="admin1" password="admin1" roles="admin-gui,manager-gui" />
```

Testing
======
http://localhost:3535
|--->gives home page ---> ManageApp
↓
submit username :admin1
password : admin1

Your batch code :: NTAJ115

Sep 28 First web application Development, Deployment and execution

Procedure to develop and deploy first Web application having html files , images as webcomps in Tomcat server

html files —> static web comps generatig static web pages
images,audio files, vedio files –> helper web comps

step1) make sure that following software setup is avaiable
=> Tomcat 6 + (Tomcat 10.x)
=>Edit plus 2+ (Edit plus 3)

step2) Create Deployment Directory structure/staging directory structure

=>Deployment Directory structure is common for all java based web servers/Application servers

=>Deployment directory structure gives various logical parttiations to keep different types of comps

WEB-INF/classes folder contains java classes based web comps like servlet comps,
outside the WEB-INF folder we place non-java classes based web comps like html files,
css files, js files , images, audio files, video files and etc..

```
E:
 |--->FirstApp (web root fodler becoz it root folder for entire web application)
        |---------->WEB-INF
        |--->home.html      |--->classes
        |--->about_us.html
        |--->contact_us.html
        |--->nit.jpg
        |--->home.jpg
```

=>After deployment of the web application the webroot folder name becomes the name of the web application.

=>web root folder can created in any place having any name.

=>if the web application not having java classes based web comps like servlet comps then placing WEB-INF, classes folders is purely optional

=>WEB-INF,classes folder names are fixed and case-sensitive.

=>While talking html file names , we do not follow java naming conventions i.e we take all letters in lowercase and multiple words will be seperate with "_" symbol.

step3) Develop and arrange the web comps

home.html
==========

```
<h1 style="color:blue;text-align:center"> N A R E S H  I T  <img src="nit_logo.jpg"> </h1>
<hr>
<br><br>
<a href="about_us.html"> AboutUS </a>          
<a href="contact_us.html"> ContactUS </a>
```

Horizontal rule — heading — To display image — new line — To generate spacje — anchor tag for hyperlink

Press ctrl+B in Ediplus to see the preview of html page in the browser.

about_us.html
==========

```
<h1>Training Institute Overview </h1>
<h2>Naresh i Technologies </h2>

<pre><b>Naresh i Technologies (Pronounced: NareshIT) is a leading software training institute providing Software
Training, Project Guidance, IT Consulting and Technology Workshops. </b></pre>

<pre><b>Using our enhanced global software training delivery methodology, innovative software training
approach and industry expertise, we provide high-value corporate training services that enable our
clients to enhance business performance, accelerate time-to-market, increase productivity and improve
customer service. </b></pre>

<pre> <b>We serve Global 100 companies and the leading software vendors in Banking & Financial Services,
Insurance, Telecommunications, Technology and Media, Information & Education industries. We design
and mentor human resources for our clients who create competitive advantage.</b> </pre>
<br><br>
<a href="home.html"><img src="home.jpg"></a>
```

contact_us.html
==========

```
<h1>Get in Touch with Naresh i Technologies </h1>
<b>Thank you for your interest in NareshIT <b>

<pre><b>Please read below that best fits your needs and provide some brief information that
will enable us to route your request to the appropriate person. We look forward to quickly responding
to your inquiry. </b> </pre>

<pre><b>Do you have Questions?                                  <pre> —>pre-foramtting paragraph
We really appreciate you taking the time to Get In Touch with us.   <b> –> to bold the content
We would love to hear from you! </b> <pre>
<br>
<b>India – Hyderabad Office </b>
<pre>2nd Floor, Durga Bhavani Plaza, Ameerpet, HyderabadTel: +91 40 2374 6666 (IN – Hyderabad)
Tel: +91 40 2373 4842 (IN – Hyderabad)
+91 9000994007
+91 9000994008 </pre> <br>

<b>For Projects : +91 9000994005 </b>
<pre> Email: info@nareshit.com
India – Chennai Office
2nd Floor Plot No.172 & 173, Above Axis Bank, Behind PTC Bus Stop, OMR, Thoraipakkam, Tamil Nadu, Chennai – 600097.
Mobile/Whats App: +91 9566042345
Email: chennai@nareshit.com <pre> <br>
<b>USA Office <b>
<pre>5007 Arbor View Pkwy NW Acworth, GA, 30101
Ph: +1 404-232-9879, +1 248-522-6925
Email: sriram@nareshit.com </pre>
<b> for Online Training </b>
<pre> Mobile/Whats App:
+91 81 79 19 1999
+91 92 93 22 6789
Email: online@nareshit.com </pre>
<br>
<a href="home.html"><img src="home.jpg"></a>
```

Gives graphical hyperlink

Two types hyperlinks
==========

a) Textual hyperlinks

```
<a href="about_us.html"> AboutUS </a>
```

(b) Graphical hyperlinks

```
<a href="home.html"><img src="home.jpg"></a>
```

=>step1 to step3 completes the development of the web application.

step4) Start the Tomcat server..

Go to <Tomcat_home>\bin directory and use tomcat10.exe file

step5) Deploy the web application

Copy E:\FirstApp folder to E:\Tomcat10.x\webapps folder..

```
28-Sep-2021 12:08:36.386 INFO [Catalina-utility-1] org.apache.catalina.startup.HostCo
web application directory [E:\Tomcat10.x\webapps\FirstApp] has finished in [86] ms
```

step6) Test the web application.

request url in the browser address bar.

web comp name

http://localhost:3535/FirstApp/home.html

protocol — host name & port number of Tomcat — Named of the web application /Context path/Context root of the web application

note: requst goes to home.html file of FirstApp web application deployed Tomcat server running on port number 3535.

=>Based on the the technology that is being used in the Container creation, there are different types of Containers

=>ServletContainer :: Developed using Servlet technology
=>JspContainer :: Developed using JSP technology
=>EJB Container :: Developed using EJB technology

Sep 29 Different types of Containers

=>Based on the location of the Container w.r.t Server .. there are 3 types of Containers

a) Standalone Container :: Here Server and Container togather comes as single program
eg:: Servlet Runner (given by Ms in the initial days of servlet technology)

b) In Process Container :: Here Container resides inside the server as seperate detachable unit
eg1:: ServletContainer of Tomcat server
(catalina.jar)
eg2:: Jsp container of Tomcat server
(jasper.jar)

c) Out of process Container :: Here container resides outside the server as extenally configurable unit
eg:: Servletcontainer configured with IIS.
eg: Jsp Container configured with IIS

note:: In IIS , we can not actually deploy and execute the java web applications.. So we need to cfg Servlet container,jsp container with IIS for achieving the same task.

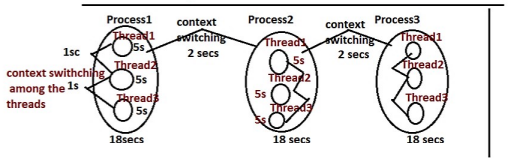
---

b/w

What is the diffrence process and Thread?

=> Process is heavy weight ... and directly executed by OS
=> Thread is light weight sub process and indirectly executed by OS ..thread resides inside the Process.
=>One Process can have any no.of threads.

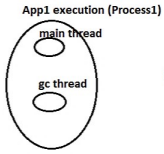
60 Seconds time frame



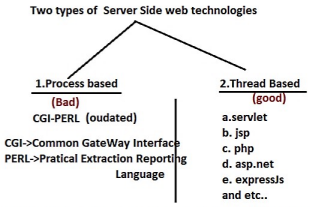
note:: The process of transffering control from one process to another process or one thread to another thread of same process is called context swithching or control jumping.. This always happens according to cpu scheduling algorithm

=> The contextswitching b/w processes take more time compare to the context switching b/w threads of a same process.

note:: when we run the Java App , One process will be created.. In that process two default threads will be created they are main thread , gc thread.

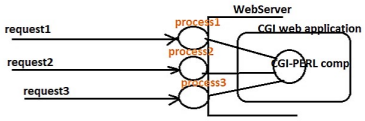


gc:: Garbage collector



=>CGI Technology can be written in multiple technologies like VB,c++, perl (best)

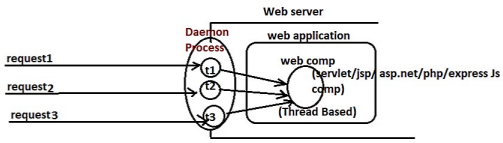
=> if we given multiple requests Process based server side web technology web application then multiple processes will be created representing muliple threads.



note:: Since processes are heavy weight and take huge amount of time for context switching .. So if we give more no.of requests then performence of the web application goes down. This make s CGI web applicaitions as the non-scalable web applications

---

if we give multiple requests to server Thread based side web comp.. then multiple threads will be created for multiple requests in a daemon process of a server ..Since threads are light weight and they take very less time in context switching .. so there will not be any performence issue even though no.of requests are increased or decreased.



t1,t2,t3 are threads of single daemon process.

=>The web applications that are developed using Thread based web technologies are scalable web applications becoz they are creating threads representing the requeests ..

Can u define Servlet ?

Ans1) It is a java based sever side web technology to develop dynamic web comps having capability to generate dynamic web pages.

(or)

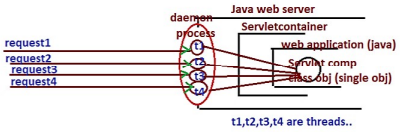
Ans2) It is JEE module web technology having copability to enhace the functionalaties of Web server or Http server or Application server.

=> The web server is getting work to do becoz of the deployed web applications and its web comps .. These web comps are developed using the support of web technologies.

Ans3) It is JEE module server side web technology that allows us to develop server side web comps as "Single instance and multiple threads based web comps" in java web application.

=>Every Servlet comp is a java class using servlet api ...

if we give 4 requests to a servlet comp ,the ServletContaienr creates only object/instance for servlet comp class and starts 4 threads representing 4 requests on that single object.



t1,t2,t3,t4 are threads..

Ans4) It is JEE module server side web technology that gives servlet api and this api will be used by Vendor companies as rules and guidelines to creates ServletContainer . the same api will be used programmers as based to create servlet comps as the web comps of the web application.

=>Servlet comp is java class that uses servlet api i.e uses the classes, interfaces enums ,annoation available in servlet api pkgs.

Sep 29 Different types of Containers

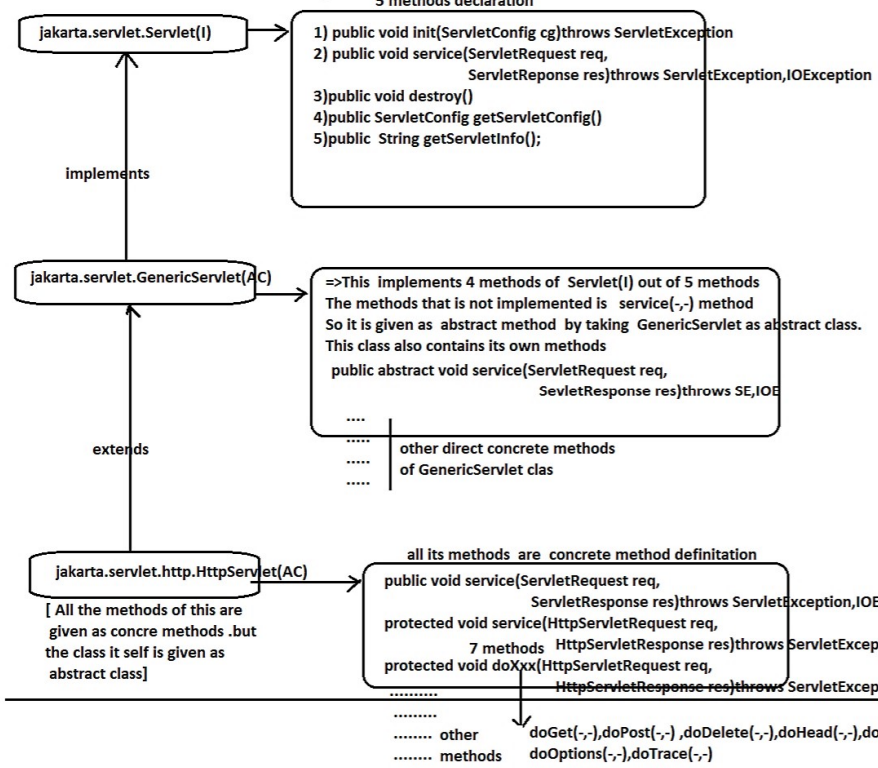
=>sevlet api latest version is 4.0.1 and the packages are
- jakarta.servlet,
- jakarta.servlet.http,
- jakarta.servlet.annotation,
- jakarta.servlet.descriptor,

In earlier version of servlet api,the packages are
- javax.servlet,
- javax.servlet.http,
- javax.servlet.annotation,
- javax.servlet.descriptor.

3 imp resouces of servlet api

=> class extends from a class
=> class implements interface(s)
=> interface extends interface(s)

=>class implenting an interface but not providing implentation to that interface methods . then we should take class as abstract class



public void m1(); abstract method / method declaration
(or)
public abstract void m1(); abstract method / method declaration

```
public void m2(){
  ...
  ..
}
```
Concrete method Definitation

=>Servlet container takes only java classes as servlet comps who implements jakarta.servlet.Servlet(I) directly or indirectly.i.e servlet container does not ordinary java classes as servlet comps.

=> To create a user-defined thread .. take a class implementing java.lang.Runnable(I) directly or indirectly
=> To create a serializable obj .. take a class implementing java.io.Serializable(I) directly or indirectly
=> To create a cloneable obj.. take a class implementing java.lang.Cloneable(I) directly or indirectly
=> To create a List colelction .. take a class implementing java.util.List(I) directly or indirectly..

=>jakarta.sevlet.GenericServlet(AC) class based seervlet comp development does not allow us to use the special features of protocol http.
=>jakarta.sevlet.HttpServlet(AC) class based servlet comp development allows us to use the special features of protocol http like auto refresh , knowing browser software name and etc..

=>Refershing content of web page at regular intervals is called auto refresh of web page .. This is required while displaying Live game scores, stcok market share values in the web pages.

=> For evey request given servlet comp , the ServletContainer automatically calls service(-,-) .. So we override service(-,-) in our servlet comp having logic to process the request and to generate results..

=> JVM begins the execution of standalone App by calling main(-) on the given class name.
=>Servlet Comp is not stadalone App .. It is web comp of web application whose life cycle mangement inclucing execution is taken by ServletContainer .. This servlet container actually calls service(-,-) method for every request, So there is no need to placing main(-) method in our servlet comp becoz it is not standlone App.. it is web comp whose execution is done by Servletcontainer.

In java,
concrete class :: The java class for which object can be created
abstract class :: The java class for which object can not be created (Incomplete class)
->Abstract class can have only abstract methods or only concrete methods or both.
Upto Java7 :: Interface can have only abstract methods declaration
From Java8 onwords :: interface can have methods declaration (abstract methods), default methods definitation, static methods definitation

Different approaches of developing servlet comp

Oct 01 Different types of Developing servlet comp

=> Servlet comp is a java class that implements jakarta.servlet.Servlet(I) directly or indirectly.

3 approaches of developing servlet comp

=>Our servlet comp class must always to concrete class

Approach1 (bad) Take a java class implementing jakarta.servlet.Servlet(I)

```
public class TestServlet implements jakarta.servlet.Servlet(I){

  ......
  ..... //provide implentation for 5 methods including service(-,-)
  ..... // place request processing logic in service(-,-) method
}
```

limitations of Approach1

=> Programmer is forced to implement all the five methods .. though he is not intrested in other methods except service(-,-) method.

=> Does not allow to work with protocol "http" features like auto refresh.

Approach2 :: (Bad) Take a java class extending jakarta.servlet.GenericServlet(AC) and provide implementation for service(-,-) method having request processing logic

```
public class TestServlet extends jakarta.servlet.GenericServlet(AC){

  public void service(SevletRequest req,
                      ServletResponse res )throws SE,IOE{
    .....
    ..... // request processing logic
    ....
  }
}
```

advantages of approach2

=>we need to implement only one method .. that to required method service(-,-) having request processing logic

=> The other methods can be overridden as needed.

Disadvanges

=> we can not work with protocol "http" features like autorefresh..

note:: Approach1,Approach2 style servlet comp development takes protocol requests.. but we can not use the advanced features of protocol http i.e we can use only common fetures of web programming like request processing and response generation.

Appraoch3 (Good Approach) :: Take a java class extends jakarta.servlet.http.HttpServlet(AC) and override either one of two service(-,-) mehods or one of the seven doXxx(-,-) to place the request processing logic. (Best)

```
public class TestServlet extends jakarta.servlet.http.HttpServlet{

  protected void service(HttpServletRequest req, HttpServletResponse res)throws SE,IOE {
      ....
      .... //request processing logic
      ....
  }
}
```

advantages

=> we can override our choice methods to place the request processing logics as needed

=> we can use protocol "http" features like autorefesh

=> In servlet comp we can override multiple request processing methods to place diffrent request processing logics for diffrent modes of request.

=> It is indurstry standard approach to develope servlet comp.

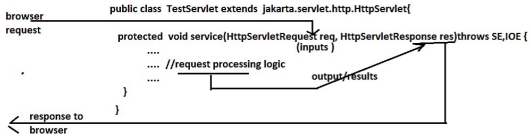
=>ServletContainer manager servlet comp life cycle from birth to death (object creation to obj destruction)

=> SErvletContainer calls service(-,-) having req ,res objs as the arguments for every request on ur servlet comp class object.

=>SErvletContainer creates 1 set of req ,res objs for every request and passes them as arguments while service(-,-) method.

=>req object holds multiple details that are coming along with request given by browser the details are like form data(req parameters), browser details(req headers), URL details(Misc info) and etc.The service(-,-) servelt comp uses req object to gather diffrent inputs that are coming along with the request.

=>res object is ready to hold the outputs/resutls generated by the request processing logic.. and useful to send that output to browser as the web pagecontent through servletContainer and web server. The output generated in service(-,-) method by request processing logic will be placed in res object ..in order to send browser as the web page content.



if we give 20 requests to servlet comp ,the ServletContainer

=> creates 1 object for our servlet comp class

=> creates 20 sets of req,res objs for 20 requests on 1 set per each request

=>creates 20 threads for 20 reqeusts.

Procedure to develop,deploy and excute SEcond Java web applicaiton having the servlet comp as the web comp

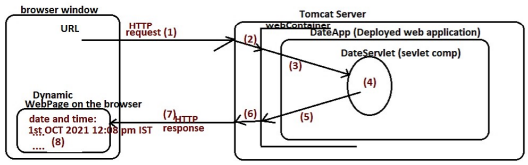
step1) arrange the s/w setup

=> jdk 1.8+ version (jdk1.8)

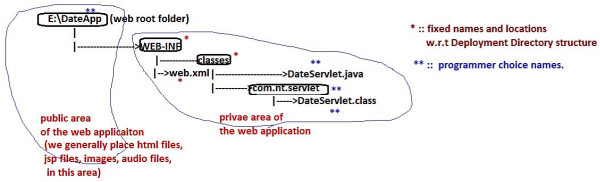
=> Tomcat 7+ (Tomcat 10)

=> EditPlus 2+ (Edlit Plus)

step2) Design the Application



step3) create Deployment Directory structure of the web application.



=>After Deployment the web application , the WEB-INF and its sub folders are called private area of the web applicaiton.. Only the underlying server and its web container can use content of private area.. i.e no outsiders or outside request can not enter into private with out having permission of web container.. So\ we place java classes like servlet comp in prvate area to protect their source and bycode from outsider.. if any outsider wants to give request to private area comps they need to permission from webcontainer by writting entries in web.xml file.

conclusion:: No outside access and modify private are web comps code.. But outside can request to private area web comps (servlet comps) by taking permission from web container using the web.xml file entries.

=> The outside area of WEB-INF folder that is there in webroot folder is called public area of the web application.. any one including outside can access /request public area web comps like html files ,jsp files and etc..with out having any permission from webContainer.

Why do not we think about placing html, jsp and other helper web comps in private area of the web application?

Ans) html files code html code , jsp files html code , helper comps goes browser along with the response beoz thier code execution and utlization takes place in the browser, So if we keep in the private area also the same process has to be continued. due to this placing the above said comps private area of the web application is totally waste of time.

js file

Different approaches of developing servlet comp
==

Oct 01 Different types of Developing servlet comp

=> Servlet comp is a java class that implements jakarta.servlet.Servlet(I) directly or indirectly.

3 approaches of developing servlet comp
====================================

=>Our servlet comp class must always to concrete class

Approach1 (bad) Take a java class implementing jakarta.servlet.Servlet(I)
===

```
public class TestServlet implements jakarta.servlet.Servlet(I){

 ......
 ..... //provide implentation for 5 methods including service(-,-)
 ..... // place request processing logic in service(-,-) method
}
```

limitations of Approach1
=====================
=> Programmer is forced to implement all the five methods .. though he is not intrested in other methods except service(-,-) method.
=> Does not allow to work with protocol "http" features like auto refresh.

Approach2 :: (Bad) Take a java class extending jakarta.servlet.GenericServlet(AC) and provide implementation for service(-,-) method having request processing logic

```
public class TestServlet extends jakarta.servlet.GenericServlet(AC){

  public void service(SevletRequest req,
                     ServletResponse res )throws SE,IOE{
     .....
     ......   // request processing logic
      ....
  }
}
```

advantages of approach2
======================
=>we need to implement only one method .. that to required method service(-,-) having request processing logic
=> The other methods can be overridden as needed.

Disadvanges
==========
=> we can not work with protocol "http" features like autorefresh..

note:: Approach1,Approach2 style servlet comp development takes protocol requests.. but we can not use the advanced features of protocol http i.e we can use only common fetures of web programming like request processing and response generation.

Appraoch3 (Good Approach) (Best) :: Take a java class extends jakarta.servlet.http.HttpServlet(AC) and override either one of two service(-,-) mehods or one of the seven doXxx(-,-) to place the request processing logic.
========================

```
public class TestServlet extends jakarta.servlet.http.HttpServlet{

  protected void service(HttpServletRequest req, HttpServletResponse res)throws SE,IOE {
        ....
        .... //request processing logic
        ....
   }
 }
```

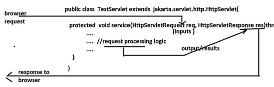
advantages

=> we can override our choice methods to place the request processing logics as needed
=> we can use protocol "http" features like autorefesh
=> In servlet comp we can override multiple request processing methods to place diffrent request processing logics for diffrent modes of request.
=> It is indurstry standard approach to develope servlet comp.

=>ServletContainer manager servlet comp life cycle from birth to death (object creation to obj destruction)
=> SErvletContainer calls service(-,-) having req ,res objs as the arguments for every request on ur servlet comp class object.
=>SErvletContainer creates 1 set of req ,res objs for every request and passes them as arguments while service(-,-) method.

=>req object holds multiple details that are coming along with request given by browser the details are like form data(req parameters), browser details(req headers), URL details(Misc info) and etc.The service(-,-) servelt comp uses req object to gather diffrent inputs that are coming along with the request.

=>res object is ready to hold the outputs/resutls generated by the request processing logic.. and useful to send that output to browser as the web pagecontent through servletContainer and web server. The output generated in service(-,-) method by request processing logic will be placed in res object ..in order to send browser as the web page content.

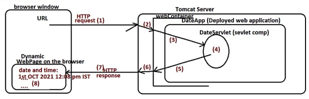


if we give 20 requests to servlet comp ,the ServletContainer
=> creates 1 object for our servlet comp class
=> creates 20 sets of req,res objs for 20 requests on 1 set per each request
=>creates 20 threads for 20 reqeusts.

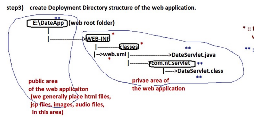
Procedure to develop,deploy and excute SEcond Java web applicaiton having the servlet comp as the web comp
==

step1) arrange the s/w setup
=> jdk 1.8+ version (jdk1.8)
=> Tomcat 7+ (Tomcat 10)
=> EditPlus 2+ (Edlit Plus)

step2) Design the Application



step3) create Deployment Directory structure of the web application.



* :: fixed names and locations w.r.t Deployment Directory structure

** :: programmer choice names.

=>After Deployment the web application , the WEB-INF and its sub folders are called private area of the web appliication.. Only the underlying server and its web container can use content of private area.. i.e no outsiders or outside request can not enter into private with out having permission of web container.. So\ we place java classes like servlet comp in prvate area to protect their source and bycode from outsider.. if any outsider wants to give request to private area comps they need to permission from webcontainer by writting entries in web.xml file.
conclusion:: No outside access and modify private are web comps code.. But outside can request to private area web comps (servlet comps) by taking permission from web container using the web.xml file entries.

=> The outside area of WEB-INF folder that is there in webroot folder is called public area of the web application.. any one including outside can access /request public area web comps like html files ,jsp files and etc..with out having any permission from webContainer.

Why do not we think about placing html, jsp, js file and other helper web comps in private area of the web application?
Ans) html files code html code , jsp files html code , helper comps goes browser along with the response beoz thier code execution and utlization takes place in the browser, So if we keep in the private area also the same process has to be continued. due to this placing the above said comps private area of the web application is totally waste of time.

step4) Develope the servlet comp in E:\DateApp\WEB-INF\classes folder

our servlet comp class must be taken as public class in order that class visible inside and outside the of its pkg.. It is very much required to make our servlet comp class visiable and accessible servletContainer to loading and instantiation of servlet comp class.

jakarta.servlet.Servlet(I)
↑ implements
jakarta.servlet.GenericServlet(AC)
↑ extends
com.nt.servlet. DateServlet (C)

=>Here "com.nt.servlet.DateServlet (c)" indirectly implementing jakarta.servlet.Servlet(I)

=> if ServletContainer gets problems during the execution of servlet comp..the ServletException will be thrown indicating the Problem
=> res obj intenrally contains streams.. if problems are raised while accessing and using the streams then IOException will be raised.

=>Output flow from service(-,-) method of servlet comp
=>service(-,-) method's req proicessing logic generates the output/results ==> writes that output/results to res object using streams ===> res object writes ServletContainer ===> ServletContainer passes to WebServer ==> web Server gives to browser as response ==> browser displays the response content as the web page.

=>What is stream?
A stream is continuos flow of data that can be used to write or read data to/from destination.
It is pipe connection b/w source and destination.
The destination of stream can be a file or object or array or device

Java gives 3 built-in streams
=>System.out (standard output stream pointing to monitor/console to write data)
=>System.err (standard error stream pointing to monitor/console to write data of type errors)
=>System.in (standard input stream pointing to keyboard)

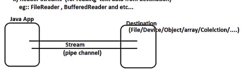
These are built-in stream having devices as the destination..

=>Two types streams
a) Byte streams (Can be used to write /read both text data and binary data)
b) Character streams (Can be used to write /read only text data)

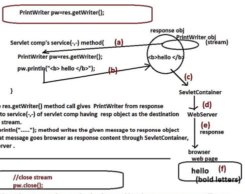
images,audio files, vedio files and etc.. represents binary data (internally 0,1)

=> Two types in ByteStreams
i)OutputStreams (for wrting text, binary data to destination)
eg:: FileOutputStream , DataOutputStream, ObjectOutputStream and etc..
ii) InputStreams (for reading text,bindary data from destination)
eg:: FileInputStream , DataInputStream, ObjectInputStream and etc..

=> Two types in Character streams
i)Write Streams (for wrting text data to destination)
eg:: FileWriter, BuffereWriter, PrintWriter and etc..
ii) Reader Streams (for reading text data from destination)
eg:: FileReader , BufferedReader and etc...



```
PrintWriter pw=res.getWriter();
```



=>Here res.getWriter() method call gives PrintWriter from response object to service(-,-) of servlet comp having resp object as the destination for the stream.
=>pw.println("....."); method writes the given message to response object and that message goes browser as response content through SevletContainer, WebServer .

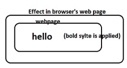
```
//close stream
pw.close();
```

=>Closes the stream and passes the signal to res object that no more messages will come to res obj .. Based on this res object starts writing the recived all messages to Servletcontainer.

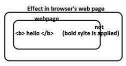
```
//Gives content type instruction to browser from res obj
  res.setCotentType("text/html");
```

our servlet comp is giving signal or instruction to browser through res obj, ServletContainer, web server to display the recived content as the html code style text content i.e html tag styles will be applied..
The default content type , if no content type is given is "text/plain" i,e browser displays recieved cotent as it is with out applying any styles.

In servlet comp
==============
```
res.setContentType("text/html");
PrintWriter pw=res.getWriter();
pw.println("<b> hello </b>");
```



In servlet comp
==============
```
PrintWriter pw=res.getWriter();
pw.println("<b> hello </b>");
```

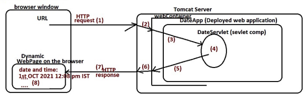


Procedure to develop,deploy and excute SEcond Java web applicaiton having the servlet comp as the web comp
==

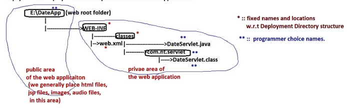
Oct 04 SEcond sErvlet App Development

step1) arrange the s/w setup
- => jdk 1.8+ version (jdk1.8)
- => Tomcat 7+ (Tomcat 10)
- => EditPlus 2+ (Edlit Plus)

step2) Design the Application



step3) create Deployment Directory structure of the web application.



=>After Deployment the web application , the WEB-INF and its sub folders are called private area of the web appliication.. Only the underlying server and its web container can use content of private area.. i.e no outsiders or outside request can not enter into private with out having permission of web container.. So\ we place java classes like servlet comp in prvate area to protect their source and bycode from outsider.. if any outsider wants to give request to private area comps they need to permission from webcontainer by writting entries in web.xml file.
conclusion:: No outside access and modify private are web comps code.. But outside can request to private area web comps (servlet comps) by taking permission from web container using the web.xml file entries.

=> The outside area of WEB-INF folder that is there in webroot folder is called public area of the web application.. any one including outside can access /request public area web comps like html files ,jsp files and etc..with out having any permission from webContainer.

Why do not we think about placing html, jsp (js file) and other helper web comps in private area of the web application?
Ans) html files code html code , jsp files html code , helper comps goes browser along with the response beoz thier code execution and utlization takes place in the browser, So if we keep in the private area also the same process has to be continued. due to this placing the above said comps private area of the web application is totally waste of time.

step4) Develope the servlet comp in E:\DateApp\WEB-INF\classes folder

our servlet comp class must be taken as public class in order that class visible inside and outside the of its pkg.. It is very much required to make our servlet comp class visiable and accessible servletContainer to loading and instantiation of servlet comp class.

jakarta.servlet.Servlet(I)
↑ implements
jakarta.servlet.GenericServlet(AC)
↑ extends
com.nt.servlet. DateServlet (C)

=>Here "com.nt.servlet.DateServlet (c)" indirectly implementing jakarta.servlet.Servlet(I)

=> if ServletContainer gets problems during the execution of servlet comp..the ServletException will be thrown indicating the Problem
=> res obj intenrally contains streams.. if problems are raised while accessing and using the streams then IOException will be raised.

=>Output flow from service(-,-) method of servlet comp
=>service(-,-) method's req proicessing logic generates the output/results ==> writes that output/results to res object using streams ===> res object writes ServletContainer ===> ServletContainer passes to WebServer ==> web Server gives to browser as response ==> browser displays the response content as the web page.

=>What is stream?
A stream is continuos flow of data that can be used to write or read data to/from destination.
It is pipe connection b/w source and destination.
The destination of stream can be a file or object or array or device

Java gives 3 built-in streams
- =>System.out (standard output stream pointing to monitor/console to write data)
- =>System.err (standard error stream pointing to monitor/console to write data of type errors)
- =>System.in (standard input stream pointing to keyboard)

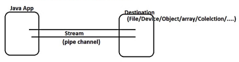
These are built-in stream having devices as the destination..

=>Two types streams
- a) Byte streams (Can be used to write /read both text data and binary data)
- b) Character streams (Can be used to write /read only text data)

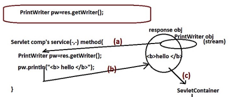
images,audio files, vedio files and etc.. represents binary data (internally 0,1)

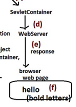
=> Two types in ByteStreams
- i)OutputStreams (for wrting text, binary data to destination)
  eg:: FileOutputStream , DataOutputStream, ObjectOutputStream and etc..
- ii) InputStreams (for reading text,bindary data from destination)
  eg:: FileInputStream , DataInputStream, ObjectInputStream and etc..

=> Two types in Character streams
- i)Write Streams (for wrting text data to destination)
  eg:: FileWriter, BuffereWriter, PrintWriter and etc..
- ii) Reader Streams (for reading text data from destination)
  eg:: FileReader , BufferedReader and etc...



PrintWriter pw=res.getWriter();





=>Here res.getWriter() method call gives PrintWriter from response object to service(-,-) of servlet comp having resp object as the destination for the stream.
=>pw.println("....."); method writes the given message to response object and that message goes browser as response content through SevletContainer, WebServer .

```
//close stream
pw.close();
```

=>Closes the stream and passes the signal to res object that no more messages will come to res obj .. Based on this res object starts writing the recived all messages to Servletcontainer.

```
//Gives content type instruction to browser from res obj
res.setCotentType("text/html");
```

our servlet comp is giving signal or instruction to browser through res obj, ServletContainer, web server to display the recived content as the html code style text content i.e html tag styles will be applied..
The default content type , if no content type is given is "text/plain" i,e browser displays recieved cotent as it is with out applying any styles.

```
//DateServlet.java
package com.nt.servlet;

//import pkgs
import jakarta.servlet.*;  ///servlet api
import  java.util.*;  // Date class
import  java.io.*;  //Streams
public class   DateServlet extends  GenericServet
{

      public void service(ServletRequest req,ServletResponse res)throws ServletException,IOException{
       //create/get Stream pointing to res object.
              PrintWriter pw=res.getWriter();
                 //Gives content type instruction to browser from res obj
          res.setContentType("text/html");
            //b.logic /request processing logic
             Date d=new Date();
             String sysDate=d.toString();
       //write result messages to response object
       pw.println("<h1 style='color:red;text-align:center'>"+sysDate+" </h1>");

             //close stream
             pw.close();

      }//service(-,-)
}//class
```

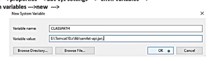
=>System.out is PrintStream object pointing to console ,So the System.out.println(-) writes the messages to console monitor
=>In PrintWriter pw=res.getWriter(); the PrintWriter stream points to response obj..So the pw.println(-) writes the messages to response object.

step 5) add <Tomcat_home>\lib\servlet-api.jar file to CLASSPAH

=>servlet api is not part jdk.. It is seperate api that comes in seperate jar file given by server In tomcat server that jar file name is servlet-api.jar in <Tomcat_home>\lib folder.

=>if any java class uses third party api or user-defined api (other than jdk api)then that api related jar file must be added to CLASSPATH in order to make java tools like "javac" , "java" and etc.. recognizing and using third party api

This PC --->properties --->adv sys settings ---> env.. variables ---> system variables --->new --->



cmd> javap jakarta.servlet.GenericServlet --> if it is not giving error then we can say that the jar file is added to CLASSPATH properly.

step6) compile the servlet comp class using "javac -d " option
makes java compiler to create .class file in the pkg of specified directory

```
E:\DateApp\WEB-INF\classes>javac  -d  .  DateServlet.java
```

The current directory (e:\DateApp\WEB-INF\classes) is the location where to create the specified pkg (com.nt.servlet)

step7) Develop web.xml file having ServletConfiguration and Servlet Mapping details.

=>ServletConfiguration means writing the details about serrvlet comp to SErletcontainer like fully class name, logical name and etc..

=>ServletMapping means linking certain incoming request url /request path to certain servlet comp that is cfg using the above process

=>web.xml file is called as web applicaiton cfg file becoz it contains entries/details about web application and its web comps
=>web.xml file is also called DD file (Deployment Descriptor) becoz the Servletcontainer reads this file the moment web application is deployed to provide info about web application and its web comps.

web.xml
========

ctrl+B in edit plus
check the correctness of xml file

```
<web-app>
  <!--Serlvet configuration -->
   <servlet>
       <servlet-name> date </servlet-name>
        <servlet-class>com.nt.servlet.DateServlet</servlet-class>
      </servlet>

<!--Serlvet mapping -->
   <servlet-mapping>
         <servlet-name> date </servlet-name>
         <url-pattern>/first</url-pattern>
   </servlet-mapping>
</web-app>
```

servlet comp details: Logical name* (servlet-name), fully qualified servlet class name ** (servlet-class)
mapping url details: logical name * (servlet-name), url /request path mapped to servlet comp * (url-pattern)
Both Logical names must match

* -> servlet logical name, url pattern are programmers choice.
**-->Servlet class name must be fully qualified class name

=> Internally , the servlet container creates our servlet comp class obj having the given logical name as the object name.
=> The logical name in servlet cfg and the logical name in servlet mapping must match in order to link Sevlet comp with certain request url/request path.

** servlet comp in the server or outside the server is not identified with its class name or logical name..It is always identified with its url pattern

=>The web.xml configuration gives instruction to ServletContainer.. please allow request servlet comp "DateServlet" if the incoming request url of contains "/first" as the request path (last word in the request url)

Xml basics
===========
- =>xml is strictly typed language
- =>xml is given to describe the data.. not for displaying data
- =>xml tags and attributes are cas-sensitive
- => In xml file, all tags must be nested properly
- => xml attribute values must be quoted either using single quotes or using doble quotes
- =>first tag of xml file is called root tag and the entire content must be written betwee open root tag and closing root tag.

step1 to step6 :: Development of the web application..

step8) Start the Tomcat server
use <Tomcat_home>\bin\tomcat10.exe file

step9) The deploy the web application in tomcat server
copy E:\DateApp folder to <Tomcat_home>\webapps folder

step10) Execute the Servlet comp by sending request from browser (Testing the web applicaiton)

In browser's address bar
http://localhost:3535/DateApp/first

=>The modifications done in the source code servlet comp of the deploy web applicaiton of running Tomcat server will reflect in the output/response for following two operations

Oct 05 SEcond sErvlet App improvement

a) Recompile the servlet comp

E:\Tomcat10.x\webapps\DateApp\WEB-INF\classes>javac -d . DateServlet.java

b) reload the web appliation from Tomcat manager

Tomcat home page (http://localhost:3535)----> Manager App --->

username :: admin1 , password : admin1 --->go to Dateapp ----> reload.

(stop +start of the webapplication)

note:: In most of the browsers.. the default response content type is "text/html" i.e if web comp is not specifying any response content type then the browser dispalys the recieved response by taking text/html as the default content type.

=> For the changes done in the web.xml file of deployed java web application. the server automatically reloads the web application (i.e we need not reload the web application manuall)

note:: same rule is applied for html,jsp and for other helper web comps

---

404 error will be raised (request resource not found)

==========

a) for wrong request url typed in browser address
b) for basic syntatical mistakes done in web.xml file
c) for taking worng names in standard directories of Deployment Directory structure
d) for wrong deployment directory structure order.
and etc...

500 error will be raised (Server Internal problem)

==========

a) if we give wrong class name or pkg name in <servlet-class>tag of web.xml file
b) if we have not taken the servlet comp class as public class
c) if ServletComp class is not having public 0-param costructor directly(given by programmer) or indirectly(given by javac compiler as default constructor)
d) if we place Zero-param constructor as the private constructor in servlet comp class
and etc..

All these are problems related to Loading and instantiating servlet comp class

Can we place only parameterized constructors in servlet comp class?

Ans) No , we can not place

=>ServletContainer creates our servlet comp class object using public 0-param constructor
=> if no constructor is placed in servlet comp then the "javac" compiler generated "public 0-param constructor as the default consturctor" and the SErvletcontainer uses that constructor for object creation..

=> if we place user-defiend parametersized constructor then the "javac" compiler does not generates "public 0-param constructor" as the default consturctor so the Servletcontainer fails to create our servlet comp class obj.

note:: our servlet comp must have public 0-param cosntructor directly (given by the programmer) or indirectly (given by the javac compiler as default constructor)

=>the url pattern given in sevlet mapping .. not only make servlet container to map incoming request with servlet comp of private area .. It alos helps to hilde technology of the web comp that is used in web application development.

for example, if url patter of servlet comp is /firsturl.cpp

<url-pattern>/firsturl.cpp</url-pattern>

then we request to that servlet comp using

http://localhost:3030/DateApp/firsturl.cpp

⇩

Based in this url ,the enduser thinks that web page has come using cpp program.
[This helps to protect our App from hakers and jackers]

Q) Can u tell me how many of servlet comps are there?

Ans) wrong answer1 :: two types of servlet copmps

a) "GenericServlet " , "HttpServlet"

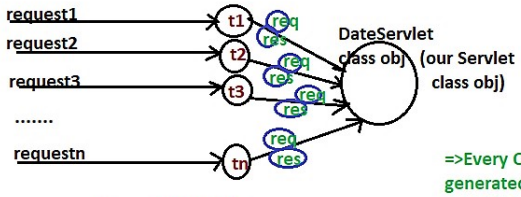
wrong answer2 :: 3 types of servlet copmps

a) "GenericServlet " , "HttpServlet" , "Servlet type"

There is possibility of developing "n" types of servlet comps like "HttpServlet", "SMTPServlet", "FTPServlet " and etc.. As of now entire industry and all the web servers and applicaitons servers are gsuing based on protocol "HTTP".. so we can say GenericServlet is not sperate type .. that is coming to common super class for multiple protocol speficific servlet comps.

if 10 requests are given to servlet comp then the servletContainer creates
=> one object for our servlet comp class.
=> 10 sets of req,res objects for 10 requests.
=> 10 threads representing 10 requests.

Oct 06 SEcond sErvlet App improvement



t1 ..... tn threads..

=>Every Object is identified with the JVM generated unique id value called hashCode.
To get hashCode any object we can call hashCode() method on object ref.

note:: hashCode() is public method of java.lang.Object class , it can be called on any java class object .

=>Thread.currentThred() gives Thread obj representing the current invoking thread
=>"this" gives current invoking object ref.. in case of servlet comp it refers to our servlet comp class obj ref
=> "req","res" parameter of service(-,-) represents req,res objs.

To prove the above concept and to prove that our servlet comp is "single instance multiple threads comp" we need to place the following in our servlet comp

```
pw.println("<br><b> our servlet class obj hashCode ::"+this.hashCode()+"</b>");
pw.println("<br><b> req obj hashCode ::"+req.hashCode()+"</b>");
pw.println("<br><b> res obj hashCode ::"+res.hashCode()+"</b>");
pw.println("<br><b> current thread   hashCode ::"+Thread.currentThread().hashCode()+"</b>");
```

static method

browser1 (Chrome)

our servlet class obj hashCode ::1835469131
req obj hashCode ::1671494295
res obj hashCode ::1738498619
current thread hashCode ::580776108

browser2 (fire fox)

our servlet class obj hashCode ::1835469131
req obj hashCode ::74359379
res obj hashCode ::178867374
current thread hashCode ::244762915

browser3 (MS edge)

our servlet class obj hashCode ::1835469131
req obj hashCode ::581901079
res obj hashCode ::1781906541
current thread hashCode ::1363749677

=>The messages kept using pw.println(-) statement in servlet comp goes to browser as web content through res obj,Servletcontainer and web server
=>The messages kept using System.out.println(-) statement in servlet comp goes to console screen underying web server or application server.

Servlet Features
==========
a) Portable
b) Powerful
c) Efficient
d)Secured
e) Integrated
f) Cost Effective
g) scalable
and etc..

Portable
=======
=>The java web applications are portable to deploy across the multiple java web servers with out haivng any change in the deployment directory structure.. becoz we write code in Servlet programing using the servlet api supplied common interfafaces.. by avoiding the server specific implementation class names.

powerful
========
=>Servlet technology is having all the facilities that required to develop modren websistes becoz it is having powerful features.

Efficient
==========
=> By creating single object for our servlet comp class and by starting multiple threads on that object for multiple requests the servletContainer using the CPU , Memory resources effeciently.

Secured
=======
=>While delivering servlet comps based web application .. we give only .class file to the clients to protect the source code
=>we keep servlet comp in private area of the web application .. This protects the source code and byte code of the servlet comp from outsiders.
=> The url /url pattern mapped to the servlet comp hides the technology that is being used in the development of web application from hackers and jackers.

Integrated
============
=> In servlet programming, we can also use other techonologies like jdbc api, jms api , java mail api, collection api and etc..

Cost -effective /In expensive
============
=>servlet technology is an open technology and most of the java based web servers and application servers are free softwares .. Servlet based web site development is going to be cost effective (less cost)

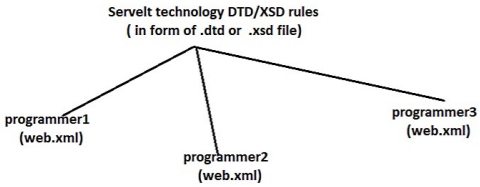
Scalable
=======
=>Since the servlet comps are Thread based comps ..So the increase or decrease in no.of requests coming to servlet comp does not effect the performence of web application.

Oct 08 SEcond sErvlet App Flow of execution

=>To make all developers of the technoloy /framework to create xml file by same set of xml tags,attributes and also using same set of rules and guidelines .. the technology creators provide DTD /XSD documents having those rules and gudiliens.. So all xml files that are created by importing those DTD/XSD rules will be created xml file using same tags,attributes having different data values.

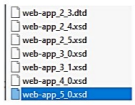
XSD :: Xml schema Definition (best)
DTD :: Document Type Definition (old)

=> To make java web application developers creating web.xml file using same tags and attributes given by Servlet Technology , the SErvlet technology gives XSD/DTD rules to programmers



=>All these programmers use same set of tags and attributes to creat web.xml files

=><Tomcat_home>\lib\servlet-api.jar file contain these .dtd , .xsd file in jakarta/servlet/resoruces folder



=> if xml file/doc is satisfying the xml syntax rules then it is called well-formed xml document
The xml syntax rules are
a) xml tags and attributes are case-sensitive
b) evey open tag should have closing tag
and etc.. (refer previous calsses)

=> if the xml file/doc is satisfying the imported XSD/DTD rules then it called Valid Xml document

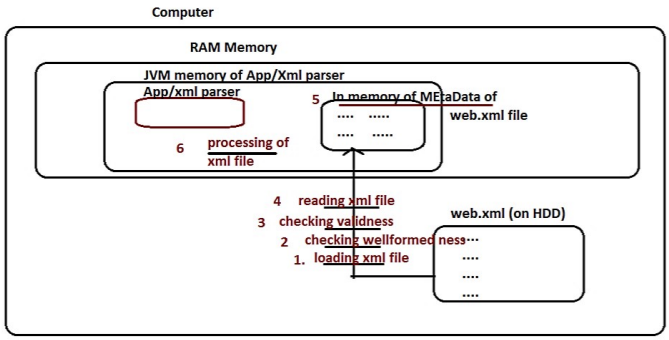
=> Xml parser can check wheather given xml file is wellformed or not , valid or not and also can load xml file, can read xml file and can process the xml file.
eg: DOM parser , SAX parser , DOM4J Parser , JDOM parser and etc..

DOM ::Document Object Model
SAX :: simple API for Xml processing
DOM4J :: DOM for Java
JDOM : Java DOM

=> ServletContainer gives one built-in xml parser called SAX parser..

=>Xml parser performs operations on the xml file in the following order

=>Loads the xml file
=>checks for well-formed ness (if not exception raises)
=>check for validness (if not exception raises)
=>reads xml file
=> Prepares InMemory MetaData of xml file In the Memory where current Xml parser(java app) is running (JVM of Memory of RAM)
=>process the Xml document
for multiple times using the InMemory data of xml file directly.

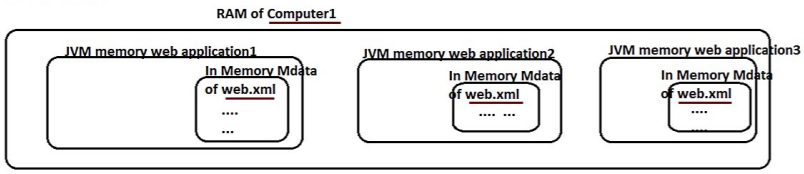


MetaData :: data about data
i.e it contains xml file content and more content on it. since that is created in the Memeory where App/xml parser is running (JVM memory of RAM)
So it is called InMemory MetaData

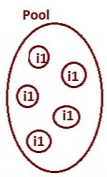
=> if the web App or servlet container needs web.xml file content for multiple times.. instead of loading xml files , checking wellformed-ness , checking validness , reading xml file for multiple times.. the ServletContaienr uses Xml parser performs all these operations for 1 time and creates InMemory MetaData for 1 time and uses that InMemory data of web.xml file for multiple times which actually improves the performence.

=> This JVM memory created for the web application having inMemory MetaData web.xml file will be vanished onc the web application is stopped or reloaded or undeployed or Server is stopped or crashed.

=> The JVM memory in RAM will be created seperately for every web application containing InMemory of its own web.xml file

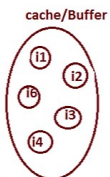


What is difference b/w pool and cache?



(Set of same items)
[Gives the reusability of same items]

=>To implement pool the best collection is List Collection



(set of different items)
[Gives the reusability of different items]

=>To implement Cache the best collection is Map Collection

ServletContainer keeps our servlet comp class objects references in the internal cache of SevletContainer which will be created on 1 per web application basis.. The servlet comp class obj ref kept in the internal cache will be used across the multiple requests.

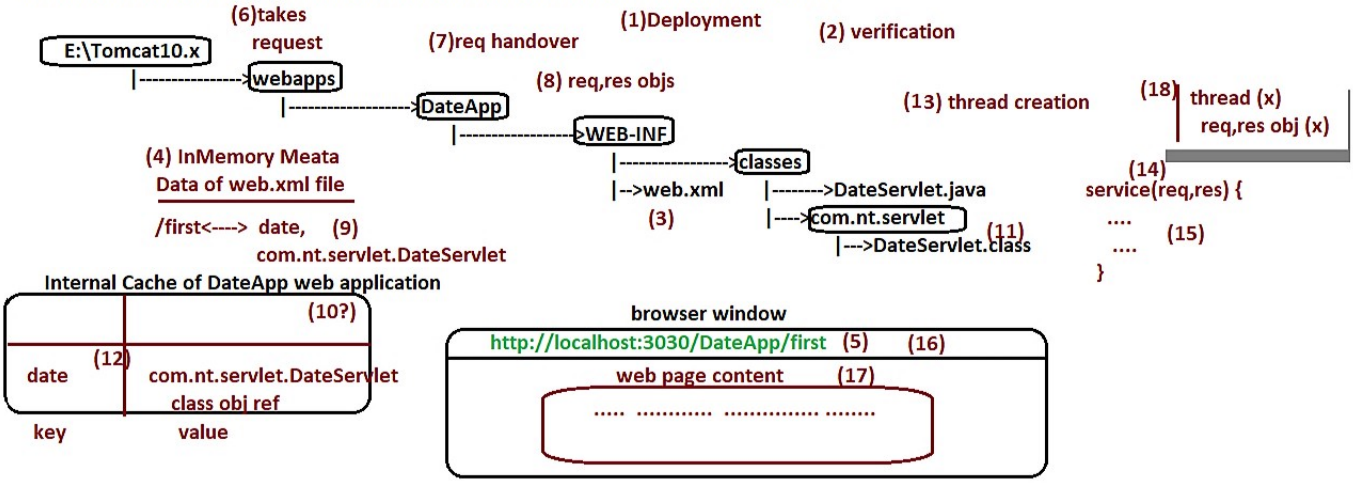
Internal cache of DateApp web application

keys	values
date	DateSevlet class obj ref
test	Testservlet class obj ref

keys == Servlet logical names given in <sevlet-name>tags
values =our servlet class obj references

End to End flow of servlet comp based java web application with the aid of Example App

Oct 09 SEcond sErvlet App Flow of execution



(1) Deployment of DateApp web application in Tomcat server

(2) The Tomcat Web server verifies the deployment directory structure of the web application

(3) ServletCotainer loads the web.xml and checks its well-formedness and validness .. if not well-formed or valid then it throws Exception.

(4) ServletContainer of Tomcat server creates InMemory MetaData of web.xml file in the Memory where the java web application is runnnig (The JVM memory of the RAM)

(5) Enduser generates the request by using the URL typed in browser address bar

(6) Tomcat server traps and takes the request

(7) Tomcat server hander the request to ServletContainer

(8) ServletContainer creates 1 set of req,res objs for the current request

(9) ServletContainer maps the reuqest with "DateServlet" of DateApp web application based on "DateApp" , "first" info collected from the request url and gets other deteails like servlet logical name ("date"), and fully qualified class name "com.nt.servlet.DateServlet"

"DateApp" ⇩ name of the webapp

"first" ⇩ request path (url pattern)

(10) ServletContainer takes the logical name "date" and search es in the Internal cache of "DateApp" web application to get SErvlet class object ..but it is not avaialble

(11) ServletContainer loads com.nt.servelt.DAteServlet class from WEB-INF/classes folder and perform instantiation (object creation) and intialization activities.

(12) ServletContainer keeps the above created Servlet class object in the Internal cache of DateApp web application for reusability of Servlet class obj .. having logical name as the key and our servlet class obj ref as the value.

13) ServletContainer creates 1 thread representing the current request.

14) The thread of current request calls service(-,-) method on our servlet class obj having req,res objs as the values.

15) The request processing logic placed in service(-,-) process the request and generates the output

16) The generated output comes browser as respons through res object, ServletContainer , webServer.

17) the browser displays the recieved response as the web page content

18) parellel to 16 ,17 th steps, The ServletContainer destroys current request thread, req, res objs

=> While overriding super class methods in sub class .. we can use either same modifer or strong moidifer.. (Not applicable for private methods).

Oct 11 Third Web App improvement

=> For example protected service(-,-) of HttpServlet class can be overridden in the sub class (our servlet class) either same protected modifier or with stroing public modifier.

```
public class HtmlServlet extends HttpServlet
{                                          This is actually protected service(-,-)method overidden with strong public modifier.
    public void service(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException{
      //get PrintWriter
          PrintWriter pw=res.getWriter();
          //set response content type
          res.setContentType("text/html");
          //write messages to res obj
          pw.println("<table border='1' bgcolor='cyan' align='center'>");
          pw.println("<tr><th> Game </th> <th> Player </th>  <th> Medal </th> </tr>");
          pw.println("<tr><td> Javelin Throw </td> <td> Neeraj chopra </td>  <td> Gold </td> </tr>");
          pw.println("<tr><td> Batminton </td> <td> PV sindhu </td>  <td> Bronze </td> </tr>");
          pw.println("<tr><td> weight Lifting </td> <td> MiraBai </td>  <td> Siver </td> </tr>");
          pw.println("<tr><td> Wrestling </td> <td> Ravi kumar </td>  <td> Siver </td> </tr>");
          pw.println("<tr><td> Wrestling </td> <td> bajarang punia </td>  <td> bronze </td> </tr>");
          pw.println("</table>");

          //close stream
          pw.close();
    }//service(-,-)
}//class
```

note:: Do not identify service(-,-) method with its public or protected modifier becoz protected service(-,-) of HttpServlet class can be overridden in our servlet comp class having the strong public modifier.

note:: Identify service(-,-) methods based on their signatures

like 1st service(-,-) and 2nd service(-,-)

1st service(-,-) :: public void service(ServletRequest req, ServletResponse res)throws SE,IOE

2nd service(-,-) :: protected void service(HttpServletRequest req, HttpServletResponse res)throws SE,IOE

How enable auto refesh on the web page generated by the servlet comp?

=>In the source code of our servlet comp class that extends from HttpServlet class which gives HttpServletRequest , HttpSevletResponse objs we need to place the following code..

```
res.setHeader("refresh","4");  (or)
res.setIntHeader("refresh",4);
```

→ makes browser to refresh webpage automatically for 4 seconds

=>contentType,refresh and etc.. are the response headers that gives instructions to browwser along wth response to perform different activites

=> "contentType" response header gives instruction to browwser to display web page in the required format

=> "refresh" response header gives instruction to browser to refresh the web page at regular intervals..

=> while dipslaying live game scores, stock market share values, weather report .. we need to use this auto refresh option.

When HttpServletRequest ,HttpServletResponse are the interfaces .. how can u say req, res are the objects?

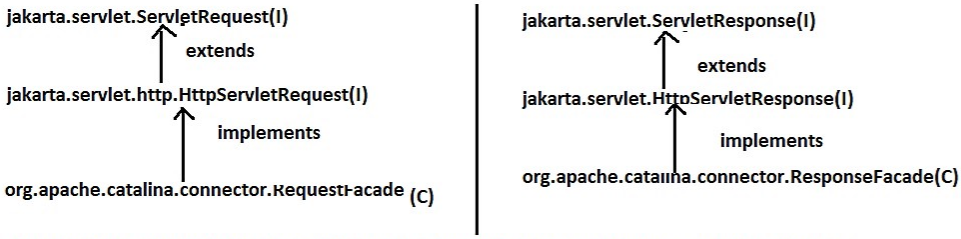
=> req ,res are not the objects HttpServletRequest , HttpServletResponse interfaces respectively .. they are the objects underlying servlet container supplied java class implemeting HttpServletRequest, HttpServletResponse interfaces respectively. The actual class names of req,res objs change server to server or contianer to container .. So we do not highligth the real class names of req, res objs.. we try to refere them using common interfaces names given servlet api.

```
pw.println("<br> req obj class name ::"+req.getClass()+"</b>");
```

In Tomcat server req object class name :: org.apache.catalina.connector.RequestFacade

```
pw.println("<br> res obj class name ::"+res.getClass()+"</b>");
```

In Tomcat server res object class name :: org.apache.catalina.connector.ResponseFacade



=> if any object name is taken with interface name ,then it is not the object of interface.. It is the object of java class that implements in the interface directly or indirectly

eg:: req obj means it is the object underlying servlet container supplied java class implementing jakarta.servlet.http.HttpServletRequest(I)

eg:: res obj means it is the object underlying servlet container supplied java class implementing jakarta.servlet.http.HttpServletResponse(I)

eg: Serializable object means it is the object of java class that implements java.io.Serializable(I)

eg: List object means it is the object of java class that implements java.util.List(I)

HTML to Servlet communication
==========

Oct 12 Html to SErvlet communication

=> HTML files generate static webpages .. Servlet comps generate dynamic web pages .. To make static webpages interacting with dynamic webpages we need HTML to servlet communication

=> Giving request to servlet comp by typing request url directly in the browser address bar is very complex.More over non-technical endusers can not type those URLs perfectly in the browser address bar. In that situation enduser needs one GUI env.. like html page (static page) to generate the request..to servlet comp. this also needs HTML to Servlet communication.

Different ways to HTML to servlet communication
==========

a) Using hyerlink (To send request to servlet comp with out having enduser supplied data).
eg:: getAll emps ' gettingAlltrendingjobs Inbox

b) Using forms (To send reqeust to servlet comp having eduser supplied data as the form data)
eg: new email Id registration form , login form , enquirty form

c) Using Java script (To send reqeust to servlet comp using java script events raised in web page , browser and also on the form comps.

eg: Loading countries to select box (combo box) when browser loads the web page (onload event)
eg: sending request to servlet comp for onblur event ..to check wheather choosen/given email id is avaiable or not
⇩
raises when the form comp looses the focus.

eg: Getting suggestions for onkeydown,onkeyup events while typing content is text boxes
⇩ raises when keyboard key is pressed
⇩ raises when keyboard key is released

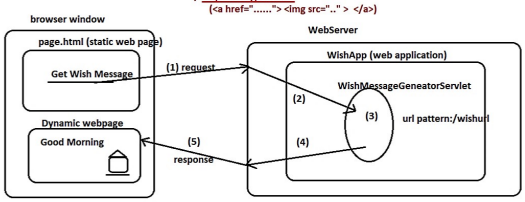
note:: Here the request will be given to servelt comp with out using hyperlinks and with out using forms.

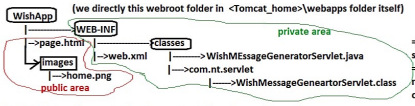
Html to servlet communication using hyperlinks
==========

=>For this we need to place the servelt comp's request url as href url of the <a> tag (gives hyerplink)

```
<a href="<servlet comp's url>"> <text>/<image> </a>
```

hyperlinks are two types
a) Textual hyperlinks
(<text>)
b) Graphical hyperlinks
(<img src=".." >)





=>WEB-INF,classes folder names are standard names..we can not change them
=>WishApp, images folder names are non-standard names.. so we can change them

Html to html interaction :: static web page to static web page interaction
Html to servlet interaction :: static web page to Dynamic web page interaction
servlet to html interaction :: Dynamic web page to static web page interaction

page.html (b)
==========

```
<h1 style="color:red;text-align:center"> Html to servlet communication using hyperlinks </h1>
                                                    (c)
<h1 style="text-align:center"> <a href="http://localhost:3535/WishApp/wishurl"> Get wish Message </a> </h1>
                                                    request url of servlet comp
```

web.xml
==========

```
<web-app>
  <servlet>
    <servlet-name>wish</servlet-name>
      <servlet-class>com.nt.servlet.WishMessageGeneratorServlet</servlet-class>
    </servlet>
    <servlet-mapping>
      <servlet-name>wish</servlet-name>
      <url-pattern>/wishurl </url-pattern>
    </servlet-mapping>
</web-app>
```

(d) Uses InMemory Mdata info

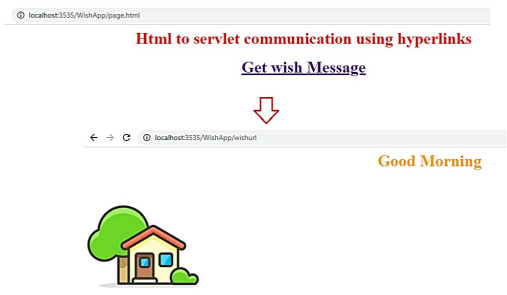
WishMessageGeneratorServlet.java
==========

```
//WishMessageGeneratorServlet.java
package com.nt.servlet;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.*;
import java.time.*;  //java 8 date and time api
                    (e) keeps servlet class obj ready
public class WishMessageGeneratorServlet extends HttpServlet
{
                    (f)
  public void service(HttpServletRequest req, HttpServletResponse res)throws ServletException,IOException{
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //write b.logic to generate wish messsage bbased on the current date and time
          //get system date and time
          LocalDateTime ldt=LocalDateTime.now();  //static method
            //get current hour of the day
            int hour=ldt.getHour();  // 24 hrs format  0 to 23
            //generate the wish message
             if(hour<12)
                  pw.println("<h1 style='color:orange;text-align:center'> Good Morning </h1>");
             else if(hour<16)
                  pw.println("<h1 style='color:red;text-align:center'> Good Afternoon </h1>");
             else if(hour<20)
                  pw.println("<h1 style='color:cyan;text-align:center'> Good Evening </h1>");
             else
                  pw.println("<h1 style='color:green;text-align:center'> Good Night </h1>");

            //add home hyperlink
              pw.println("<br><a href='http://localhost:3535/WishApp/page.html'> <img src='images/home.png'> </a>");
    }//service(-,-)                    html file/page request url
}//class
```

response to browser

request url from the browser
==========

http://localhost:3535/WishApp/page.html (a)



=>The Servletcontainer takes the request path (last word of request url) of request url .. first searches in in the private area through web.xml file for web comps cfg that is mapped with given request path.. if find then execute that web comp.. if not find then it goes public area and searches for request path info based web comp.

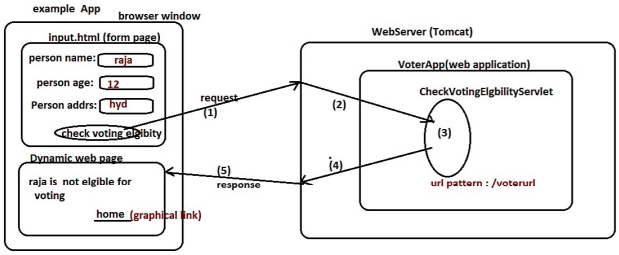
=>What happens if the html file name is taken as servlet comp's url pattern?
Ans) the request given html file also goes to servlet comp..becoz private area comps gets higher priority in the execution order by servlet container.

Assignment:: Develop web applicaiton having html to servlet interaction to display season name based on the current month of year.

HTML to servlet communication using forms — Oct 19 Html to SErvlet communication Using forms

=> if the requests given to servlet comp wants to carry enduser supplied data then go for HTML to servlet communication using forms..

eg:: email id registration , custom registration , adding product , online credit/debit card payment and etc..

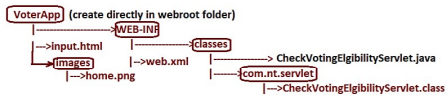


For html form to servlet communication we need to place the servlet comp's request url as the form page action url (action attribute value of <form> tag)

=>The html form page can send to modes/methodologies of requests

1.GET mode (default) ---> can carry only limited amount of data along with the request (2kb to 8kb)
(In Max browsers it is 2kb)

2.POST mode ----> can carray unlimited amount of data along with the request



input.html

```
<body bgcolor="pink">

<h1 style="color:red;text-align:center"> html to servlet communication using forms  </h1>

<form action="http://localhost:3535/VoterApp/voterurl"  method="GET">
    Person name :: <input type="text" name="pname"> <br>
    Person age :: <input type="text" name="page"> <br>
    Person address :: <input type="text" name="paddrs"> <br>
     <input type="submit"  value="check voting elgibility">
</form>

</body>
```

=>only WEB-INF,classes,web.xml folder,file names standard names (fixed names) the remaing folder, file names are not fixed

action url — request mode/method/methodology

The names of form comps (pname, page, praddrs) which later becomes request param names in the query string having form comp values as the req param values.

?pname=raja&page=12&paddrs=hyd

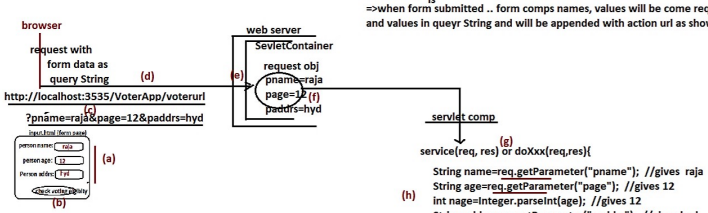
this query string will be appended with action url in "GET" mode request

http://localhost:3535/VoterApp/voterurl ?pname=raja&page=12&paddrs=hyd

action url is — query string having req param names and values.

req param name, req param value

=>when form submitted .. form comps names, values will be come req param names and values in queyr String and will be appended with action url as shown above.



=> while developing HttpServlet class based servlet comp .. it is not recomanded to place request processing logic in service(-,-) method..it is recomanded to place in one doXxx(-,-) based on the request mode or method.

=>For example , if the request mode is GET then use doGet(-,-) method.. to place request processing logic.. similarly if the request mode is POST then use doPost(-,-) method to place the request processing logic.

=> Though there are 7 doXxx(-,-) methods like doGet(-,-), doPost(-,-),doPut(-,-),doDelete(-,-),doOptions(-,-), doTrace(-,-) ,doHead(-,-)) .. we generally use only two methods doGet(-,-) ,doPost(-,-) becoz browser can send only "GET" ,"POST" mode requests.

signature of doXxx(-,-) methods
protected doXxx(HttpServletReqeust req, HttpServletResponse res)throws SE,IOE

=>using single service(-,-) we can not differentiate logics for "GET" ,"POST" requests..
where as using doGet(-,-) ,doPost(-,-) methods we can differentiate logics for "GET" , "POST" mode requests

Servlet comp

```
//CheckVotingElgibilityServlet.java
package com.nt.servlet;

import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.*;
                    (7) creates of locates the servlet class obj
public class  CheckVotingElgibilityServlet extends HttpServlet
{
                (8)  internally calls through service(-,-) method
      public  void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException{
            // get PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read form data /req param values from request object
            String name=req.getParameter("pname");
            String age=req.getParameter("page");
            int nage=Integer.parseInt(age);
            String  addrs=req.getParameter("paddrs");
            //write b.logic  or request processing
            if(nage>=18)
          pw.println("<h1 style='color:green;text-align:center'> Mr/Miss/Mrs."+name+"  u r  eglible for voting </h1>");
            else
                  pw.println("<h1 style='color:red;text-align:center'> Mr/Miss/Mrs."+name+"  u r not eglible for voting </h1>");

            //add home hyperlink
            pw.println("<br><a href='http://localhost:3535/VoterApp/input.html'><img src='images/home.png'> </a>");

            //close stream
            pw.close();
      }//method
}//class
```

(9) response to browser

input.html (2)

```
<body bgcolor="pink">

<h1 style="color:red;text-align:center"> html to servlet communication using forms  </h1>
                                                    (5)
<form action="http://localhost:3535/VoterApp/voterurl"  method="GET">
    Person name :: <input type="text" name="pname"> <br>
    Person age :: <input type="text" name="page"> <br>
    Person address :: <input type="text" name="paddrs"> <br>
     <input type="submit"  value="check voting elgibility">
</form>                                              (4)

</body>
```

(3)

web.xml

```
<web-app>
    <servlet>
        <servlet-name>voting</servlet-name>
            <servlet-class>com.nt.servlet.CheckVotingElgibilityServlet</servlet-class>
        </servlet>
        <servlet-mapping>
        <servlet-name>voting</servlet-name>
        <url-pattern>/voterurl </url-pattern>
        </servlet-mapping>

</web-app>
```

(6)

request url in browser address bar :: http://localhost:3535/VoterApp/input.html (1)

Write a web application that collects person name, age, gender (grouped radion buttons) to check wheather that person is eligible for marriage or not?

Procedure to develop web application using eclipse IDE (MarriageApp)
==========

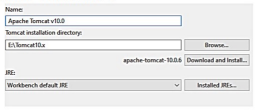
Oct 20 Html to SErvlet communication Using forms

step1) make sure that eclipse JEE (latest 2021-06) installed in u r sytem

step2) lauch eclipse IDE by choosing workspace (NTAJ115) folder
(G:\Worskpaces\advjava\NTAJ115)

step3) confiugure (link) Tomcat server with Eclipse IDE

window menu —>preferences —> server —> runtime —> add —>
apache tomcat —> tomcat 10.x —>



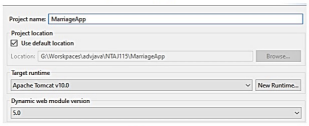
—>apply —>ok

Go to servers tab —>

select apache tomcat10.x —> next —>finish.

step4) create Dynamic web Project in eclipse ide choosing Tomcat server as the target run time env..

File menu —> new —>Dynamic web project —>



—>next —>next —>finish.

step5) understand Project directory strcture given by eclipse

=>The above structure is not deployment directory structure.. Infact the Deployment directory structure will be created dynamically when we run the applicaton or project.

step5) add input.html file in src—>main—>webapp folder and home.png to images folder of src—>main—>webapp

input.html

```
<h1 style="color:red;text-align:center"> Html to Servlet communication using forms</h1>

<form action="http://localhost:3535/MarriageApp/marriageurl" method="GET">
   <table border="0" bgcolor="cyan" align="center">
     <tr>
      <td>Person name:: </td>
      <td><input type="text" name="pname"> </td>
     </tr>
     <tr>
      <td>Person age:: </td>
      <td><input type="text" name="page"> </td>
     </tr>
     <tr>
      <td>Person gender:: </td>
      <td>
       <input type="radio" name="gender" value="M"> Male    
       <input type="radio" name="gender" value="F" checked> FeMale    
      </td>
     </tr>

     <tr>
      <td> <input type="submit" value="check marriage elgibility"> </td>
      <td> <input type="reset" value="cancel"> </td>


     </tr>

   </table>

</form>
```

=>By giving same name to multiple radio buttons we can group them into single unit and we can select only one radion button at a time from that group.

=>when we select radio button , the label of radio button will not go to server as req param value .. the data kept in the "value" attribute of selected radio button goes to server as req param value.

```
<td>
    <input type="radio" name="gender" value="M"> Male    
    <input type="radio" name="gender" value="F"> FeMale    
</td>
```

if male radio button is selected then the value "M" goes to server as request param value.

step6) create package "com.nt.servlet" in "src/main/java" folder and develop servlet comp in that folder.

```
//MarriageElgibilityCheckServlet.java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class MarriageElgibilityCheckServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read form data
        String name=req.getParameter("pname");
        int age=Integer.parseInt(req.getParameter("page"));
        String gen=req.getParameter("gender");

        //write b.logic
        if(gen.equalsIgnoreCase("M")) {
            if(age>=21) {
                pw.println("<h1 style='color:green;text-align:center'>Mr."+name+" u r eglible for marriage but think twice </h1>");
            }
            else {
                pw.println("<h1 style='color:red;text-align:center'>Mr."+name+" u r not eglible for marriage ..Be happy </h1>");
            }
        }
        else {
            if(age>=18) {
                pw.println("<h1 style='color:green;text-align:center'>Miss."+name+" u r eglible for marriage but think twice </h1>");
            }
            else {
                pw.println("<h1 style='color:red;text-align:center'>Miss."+name+" u r not eglible for marriage ..Be happy </h1>");
            }//else
        }//else
        //add home hyperlink as the graphical link
        pw.println("<br><br><a href='http://localhost:3535/MarriageApp/input.html'><img src='images/home.png'></a>");

        //clsoe stream
        pw.close();
    }//method

}//class
```

step7) cfg servlet comp in web.xml file having "/marriageurl" as the url pattern

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <servlet>
    <servlet-name>marriage</servlet-name>
    <servlet-class>com.nt.servlet.MarriageElgibilityCheckServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>marriage</servlet-name>
    <url-pattern>/marriageurl</url-pattern>
  </servlet-mapping>

</web-app>
```

step8) run the web application

right click on project —>run as —> run on server —>select Tomcat server..

This step8 performs
a) compilation of .java files into .class files
b) creating deployment directory structure
c) statring server
d) deployment of the web application
e) opening browser and showing the basic url

http://localhost:3535/MarriageApp/input.html

Absolute URL and Relative URL

Oct 20 Html to SErvlet communication Using forms

While developing web application it is recomanded to use relative URL .. not absolute URLS

```
<a href="http://localhost:3535/WishApp/wishurl"> get WishMessage </a>
```

absolute url (bad pratice)

=>when we move the applicaton to another server of same machine or different machine the port no or port no and host name in url will change.. So we need to modify the source code explicitly.. This kills WODA behaviour of the web application

WODA :: Write Once Deploy Any Where

```
<a href="wishurl"> get wish Message </a>
```

relative url (good pratice)

=>The relative URL supports to continue WODA behaviour of the web application..

```
<form action="http://localhost:3535/MarriageApp/marriageurl" method="POST">
   .....
   ....
</form>
```

absolute url (bad pratice)

=>In absolute URL we provide every detail to locate the destination comp .. So the the dest comp can be located from any source comp of same or diferent web application it is like postal address

```
<form action="marriageurl" method="POST">
   ....
   ....
</form>
```

relative URL (good pratice)

=>In relative URL we provide only request path(the last word of request url) to locate the destination comp .. So the the dest comp can be located only from source comp of same web application . It like telling address of dest location with respect to current location(go straight and take left)

note:: Stop using absolute urls .. prefer working with relative URLS

---

note :: While executing Html to Servlet communication web application.. do not give direct requests to servlet comp.. always form web page and generate request to servlet comp from that web page.

welcome files cfg

=>The first page of web application that comes automatically when given request to web application is called home page or welcome page or main page of the web application.

=>we can cfg servlet comps or jsp files or html files as the welcome files /pages of the web application by specifying their details in web.xml file

=> If multiple welcome files/pages are configured in web.xml file then the welcome file will be picked based on order of confuguration and availability.

in web.xml

```
<welcome-file-list>
  <welcome-file>input.html</welcome-file>
  <welcome-file>home.html</welcome-file>
  <welcome-file>input.jsp</welcome-file>
</welcome-file-list>
```

=> In Most of the servers like Tomcat , jetty , undertow and etc.. the index.html or index.jsp file will be taken the default welcome file if not welcome files are cfg... index.html gets high prority if both index.html and index.jsp files are available.

are

Q) if all explicitly cfg welcome files not availble physically in the web application .. but index.html or index.jsp files are avaiable ..Tell me what happens if run the web application

a) Application takes index.html as the welcome file
b) Application runs with out welcome file (correct)
c) Exception will be raised

Configuring servlet comp as the welcome

a) develop servlet comp having welcome page content

```
// HomePageServlet .java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class HomePageServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //write content to resp obj
        pw.println("<h1 style='color:blue;text-align:center'>welcome page from servlet </h1>");
        //close stream
        pw.close();

    }//doGet(-,-)


}
```

step2) cfg servlet comp in web.xml file

in web.xml

```
<servlet>
  <servlet-name>home</servlet-name>
  <servlet-class>com.nt.servlet.HomePageServlet</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>home</servlet-name>
  <url-pattern>/homeurl</url-pattern>
 </servlet-mapping>
```

step3) cfg servlet comp as the welcome file in web.xml

In web.xml

```
<welcome-file-list>
  <welcome-file>homeurl</welcome-file>
  <welcome-file>home.html</welcome-file>
  <welcome-file>input.html</welcome-file>
  <welcome-file>input.jsp</welcome-file>
</welcome-file-list>
```

Generally servlet comp is not recomanded to take as welcome file/page.. So prefer taking html file or jsp file as the welcome file

---

=>when we configure Tomcat server with eclipse IDE .. The Eclipse will not use orgnial tomcat server..it will created another copy to Tomcat server in the eclipse workspace folder and all deployment and executions of the web application takes place in that copy Tomcat server.

In our eclipse IDE the copy Tomcat server location is

G:\Worskpaces\advjava\NTAJ115\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\

- G:\Worskpaces\advjava\NTAJ115\ — workspace folder
- wtpwebapps — like webapps folder
- In this folder we can see the real deployed web application with proper deployment directory structure.

Oct 20 Html to SErvlet communication Using forms

How to make our servlet comp flexible to process both GET,POST mode requests?
i.e if change form page request mode from GET to POST or POST to GET .. our servlet comp still should be in a position to process the request.

Approach1:(bad) keep requst processing logic in service(-,-) method
==========
=>service(-,-) method can process both GET,POST mode requests..

```
public class MarriageServlet extends HttpSErvlet{

  public void service(req,res)throws SE,IOE{
      ....

       ....
    }
  }
```

Limitations of service(-,-) method
==========
=>There two service(-,-) methods confusing the programmers
=>1st service(-,-)/public service(-,-) gives ServletRequest,ServletResponse objs as the args but they can not be process to work with protocol http features
=> In servlet comp we can not write two different request processing logics for both "GET" , "POST" mode requets.
=>service(-,-) is not indudstry standard to use.

Approach2 : (working with doXxx(-,-) methods) (Good approach)
==========

Version1:: if GET,POST requests wants to execute same request processisng logic

** Override both doGet(-,-) ,doPost(-,-) methods keep request processing logic in one method and call the method from other method.

```
public class MarriageServlet extends HttpSErvlet{

  public void doGet(req,res)throws SE,IOE{
      ....
      ....    //request processing logic
      ....
    }
  public void doPost(req,res)throws SE,IOE{
      doGet(req,res);
    }
}
```

Version 2:: if GET,POST requests wants to execute s~ two different logics

*** override both doGEt(-,-),doPost(-,-) methods. write "GET" mode request proesing logic in doGEt(,-,-) and write "POST" mode reuqest prossing doPost(-,-)

```
public class MarriageServlet extends HttpSErvlet{

  public void doGet(req,res)throws SE,IOE{
      ....
      ....    //request processing logic
      ....          for GET mod requests
    }
  public void doPost(req,res)throws SE,IOE{
      ....
      ....    //request processing logic
      ....          for POST mod e requets.
    }
}
```

Approach3 (Good ... but not recomanded)
==========

Version1:: if GET,POST mode requests wants to execute same request processisng logic

*** keep request processing in user-defined methods whose parameter types are doXxx(-,-) types and call that user-defined method in both doGet(-,-) .doPost(-,-)methods.

```
public class MarriageServlet extends HttpSErvlet{
  public void process(req,res)throws SE,IOE{
      ....
      ... //requst processing logic
      ...
    }
  public void doGet(req,res)throws SE,IOE{
      process(req,res);
    }
  public void doPost(req,res)throws SE,IOE{
      process(req,res);
    }
}
```

Version 2:: if GET,POST requests wants to execute s~ two different logics

*** take "GET" mode request processing logic in user-fined method and call that mehtod from doGet(req,res) method ..
take "POST" mode request processing logic in user-fined method and call that mehtod from doPost(req,res) method ..

```
public class MarriageServlet extends HttpSErvlet{
  public void process(req,res)throws SE,IOE{   // GET
      ....
      ... //requst processing logic   (GET)
      ...
    }
  public void process(req,res)throws SE,IOE{   // POST
      ....
      ... //requst processing logic   (POST mode)
      ...
    }
  public void doGet(req,res)throws SE,IOE{
      processGet(req,res);
    }
  public void doPost(req,res)throws SE,IOE{
      processPOST(req,res);
    }
```

advantages of doXxx(-,-) methods
==========
a) allows to seperate request processing logic for both "GET" ,"POST" requests
b) gives HttpServletRequest , HttpServletResponse objs as the aguments .. so we can use protocl Http features
c) 7 doXxx(-,-) are given to process the 7 modes of request

GET ----------doGet(-,-)
POST ---------->doPost(-,-)
Head ------> doHeader(-,-)
TRACE ---> doTrace(-,-)
PUT --->doPut(-)
DELETE --->doDelete(-,-)
Option$_s$->doOptions(-,-)

d) The industry methods to place request processing is doXxx(-,-) methods.

Html to Servlet communication using Java Script **Oct 23 Html to servlet communication uising Java Script**

=> It is all about sending request to servlet comp for the event raised in Java script on web comps like text boxes, radion buttons , select boxes and etc...

=>Text box loosing focus raises onBlur Event
=> Selecting item from the select box raises onchange event
=> Cicking on submit Button raises onSubmit event
=> Cicking on standard Button raises onClick event
=> Loading of web page in browser raises onLoad event
and etc..

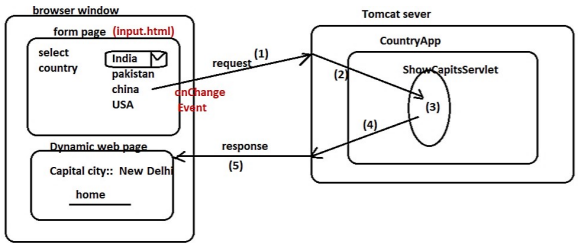
=>Java script support DOM programming (Document Object Model) ..which treats every tag as an object and gives properties and methods to invoke on those objects.

=>Accroding DOM programming <form> is an object , <b> tag is an object <input >tag an object
=>Apart of tags as objects.. Java script is also having some built-in objects like document , window ,navigator ,history and etc..

=>In Java script we can call "formobject".submit() method to submit the request with out using Submit Button

=>Url typed in the browser address bar , hyperlinked generated requests are GET mode requests

=> Form page can generate either "GET" mode or POST mode requests.

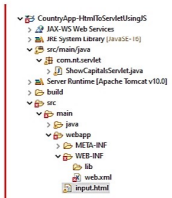


input.html (c)

```
<h1 style="color:red;text-align:center">Html to servlet communication using Java Script</h1>
(g)
<form action="capitalsurl" method="POST" name="frm">
  <table>
    <tr>
      <td> select Country :: </td>
      <td>                                                    (e)
      <select name="country" onchange="sendRequest()">
          <option value="">-----Select a value-----</option>
          <option value="0"> India </option>
          <option value="1"> Pakistan </option>
          <option value="2"> USA </option>
          <option value="3">China </option>
          <option value="4"> Srilanka </option>
      </select>
      </td>
    </tr>
  </table>
  <script language="JavaScript">
    function sendRequest(){ (f)
      frm.submit();
    }
    (g)
  </script>

</form>
```

(d) (select an item)

when we select an item in the select box .. then the selected will not go to server as req param value only the data kept in value attribute of the selected item goes to server as req param value.



web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
 xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
(b)  <welcome-file-list>
       <welcome-file>input.html</welcome-file>
     </welcome-file-list>

     <servlet>
      <servlet-name>capitals</servlet-name>
      <servlet-class>com.nt.servlet.ShowCapitalsServlet</servlet-class>
(h)  </servlet>
     <servlet-mapping>
       <servlet-name>capitals</servlet-name>
       <url-pattern>/capitalsurl</url-pattern>
     </servlet-mapping>
</web-app>
```

ShowCapitalsServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
                    (i) creates or locates Servlet class obj
public class ShowCapitalsServlet extends HttpServlet {

    @Override      (l)
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        String capitals[]=new String[] {"New Delhi","Islamabad","WashingtonDC","Beging","Columbo"};
        //general settings
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read form data
        int countryCode=Integer.parseInt(req.getParameter("country"));
        //display capital
        pw.println("<h1 style='color:red;text-align:center'>Capital city name is ::"+capitals[countryCode]+" </h1>");
        //add home hyperlink
        pw.println("<a href='input.html'> home </a>");
        //close stream
        pw.close();
    }
    @Override      (j)   (called through super class (HttpServlet) service(-,-) method)
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       doGet(req,res);  (k)
    }

}
```

(m) response to browser

request url in browser address bar
==========
(No req path in the URL ,so looks for welcome file cfg)

http://localhost:3232/CountryApp-HtmlToServletUsingJS (a)

Capital city name is ::WashingtonDC
(n)
home

improved input.html
==========

```
<h1 style="color:red;text-align:center">Html to servlet communication using Java Script</h1>

<form action="capitalsurl" method="POST" name="frm">
  <table>
    <tr>
      <td> select Country :: </td>
      <td>
      <select name="country" onchange="frm.submit()">
          <option value="">-----Select a value-----</option>
          <option value="0"> India </option>
          <option value="1"> Pakistan </option>
          <option value="2"> USA </option>
          <option value="3">China </option>
          <option value="4"> Srilanka </option>
      </select>
      </td>
    </tr>
  </table>
  </script> -->

</form>
```

After launching the web page of website on to the browser .. if modify the source code that web page using "view source option" Can u tell me the web page content will be modified or not?

Working with multple hyperlinks sending request to Servlet comp

Oct 25 Working with mutple hyperlinks and submit buttons

```
<a href="wishurl">go </a>
```

This hyperlink generated request is not carrying any data along with request

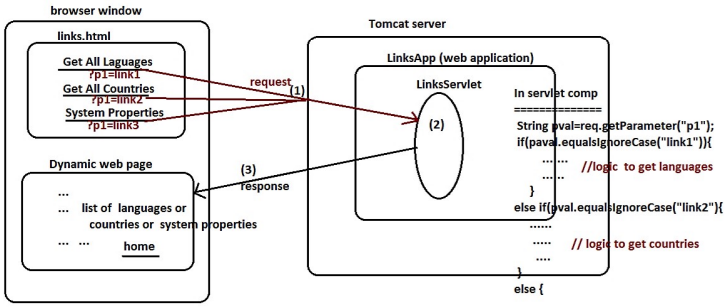
```
<a href="wishurl?p1=val1&p2=val2"> go </a>
```

This hyperlink generated request carries p1=val1, p2=val2 as the request param values along with the request.

By appending query string to the url of href attribute we can send data along with the request generated by the hyperlink

giving

How to differentiate logics for multiple hyperlinks in servlet comp when they are request same servlet comp?



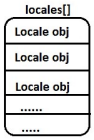
Make multiple hyperlinks sending request to servlet comp having addtional param with same name and different values .. use that addtional request param value as the criteria value to differentiaate the logics in the servlet comp

locale means language + country

example::

en-US :: English as it speaks in USA
fr-FR :: French as it speaks in France
hi-IN :: hindi as it speaks in India
te-IN :: telugu as it speaks in India
or-IN :: oria as it speaks in India
ande etc..

=>In java java.util.Locale class obj represents each locale of the outside world and provides methods to get all languages and countries.



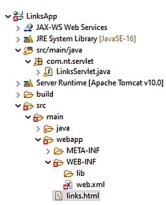

```
Locale locales[]=Locale.getAvaiableLocales();

for(Locale l: locales){                                    y
    System.out.println(l.getDisplayLanguage()+"    "+l.getDisplayCountr());
}
```

---

links.html

```
<h1 style="color:red;text-align:center">Working with multiple hyperlinks</h1>

<h1 style="color:red;text-align:center"> <a href="linksurl?p1=link1"> Get Languages</a> </h1>
<br>
<h1 style="color:red;text-align:center"> <a href="linksurl?p1=link2"> Get  countries</a> </h1>
<br>
<h1 style="color:red;text-align:center"> <a href="linksurl?p1=link3"> System Properties</a> </h1>
<br>
```

example app



LinksServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Locale;
import java.util.Properties;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class LinksServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read the value of addtional req param
        String pval=req.getParameter("p1");
        Locale locales[]=Locale.getAvailableLocales();
        //differentiate logic for different requests
        if(pval.equalsIgnoreCase("link1")) {
            pw.println("<h1> All Languages are </h1>");
            for(Locale l:locales) {
                pw.println(l.getDisplayLanguage()+",");
            }
        }
        else if(pval.equalsIgnoreCase("link2")){
            pw.println("<h1> All Countries are </h1>");
            for(Locale l:locales) {
                pw.println(l.getDisplayCountry()+",");
            }
        }
        else {
            pw.println("<h1>System Properties are </h1>");
            Properties props=System.getProperties();
            for(Object k:props.keySet()) {
                pw.println(k+"="+props.getProperty((String) k)+"<br>");
            }//for
        }//else
            //add home hyperlink
        pw.println("<a href='links.html'>home </a>");

        //close steam
        pw.close();
    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
      doGet(req,res);
    }

}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <welcome-file-list>
        <welcome-file>links.html</welcome-file>
    </welcome-file-list>

    <servlet>
      <servlet-name>links</servlet-name>
      <servlet-class>com.nt.servlet.LinksServlet</servlet-class>
    </servlet>

    <servlet-mapping>
      <servlet-name>links</servlet-name>
      <url-pattern>/linksurl</url-pattern>
    </servlet-mapping>
</web-app>
```

the web page content will be modified or not?

Ans) The chanages may reflect temporarly at client side.. but once we say refresh page the orginal code from server comes to browser .. Due to this the changes will not be reflected..perminently..

Using view Soruce option .. we can view the code. maximum.. but the perminent modification of the code is not possible.

How to differentiate logics in servlet comp when multiple submit buttons are giving request to servlet comp?

working with multiple submit buttons

Oct 26 Working with mutple hyperlinks and submit buttons

case1:

```
<form action="...." method="...">
    .....
    .....
  <input type="submit" value="send">
</form>
```

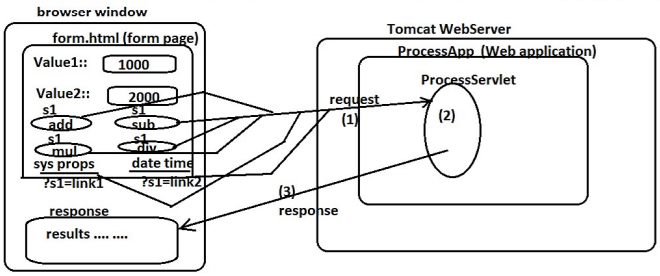
The caption of submit button does not go to server as req param value.

case2:

```
<form action="...." method="...."
    ....
    ...
  <input type="submit"  name="s1" value="send">
</form>
```

The caption of the submit goes to server along with the request as request param name and vlaue (s1=send)

How to differetiate logics in servlet comp if multiple submit buttons and hyperlinks are giving request same servlet comp?

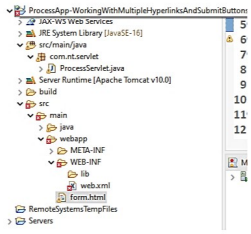



```
String pval=req.getParameter("s1");
 if(pval.equalsIgnoreCase("add")){
      ...
  }
 else if (pval.equalsIgnoreCase("sub")){
   ....
 }
 else if(pval.equalsIgnoreCase("mul")){
   ....
 }
 else if(pval.equalsIgnoreCase("div")){
   ...
  }
 else if(pval.equalsIngoreCase("link1")){
   ....
 }
 else{
   ....
 }
```

=>Take multiple submit buttons with same name and different values and also take multiple hyperlinks having addtional request param and vlaues ..make sure that the name of the addtional req param for hyperlinks is matching with the name of submit buttons.

use that addtional request param value in servlet comp as the criteria value to differetiate the logics

=>Though we have added hyperlinks in form page , the hyperlinks generated requests do not carry form generated data..



form.html

```
<h1 style="color:red;text-align:center"> Working with multiple submit Buttons and hyperlinks</h1>

<form action="processurl" method="GET">
 <table border="1" align="center" bgcolor="cyan">
   <tr>
    <td> value1:: </td>
    <td> <input type="text" name="t1"></td>
   </tr>
   <tr>
    <td> value2:: </td>
    <td> <input type="text" name="t2"></td>
   </tr>
   <tr>
    <td><input type="submit" name="s1" value="add"> </td>
    <td> <input type="submit" name="s1" value="sub"></td>
   </tr>
   <tr>
    <td><input type="submit" name="s1" value="mul"> </td>
    <td> <input type="submit" name="s1" value="div"></td>
   </tr>
   <tr>
    <td><b><a href="processurl?s1=link1">System properties</a></b>  </td>
    <td> <b><a href="processurl?s1=link2">System date and time</a></b></td>
   </tr>


 </table>

</form>
```

ProcessServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class ProcessServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response cotent type
        res.setContentType("text/html");
        //read addtional param value
        String pval=req.getParameter("s1");
        //read form data and convert  that in to numeric values only when hyperlinks are not generating requests
        float val1=0.0f, val2=0.0f;
        if(!pval.equalsIgnoreCase("link1") && !pval.equalsIgnoreCase("link2")) {
            val1=Float.parseFloat(req.getParameter("t1"));
            val2=Float.parseFloat(req.getParameter("t2"));
        }
        //write request processing logic for submit Buttons and hyerplinks
        if(pval.equalsIgnoreCase("add")) {
            float result=val1+val2;
            pw.println("<h1 style='color:red;text-align:center'>Add ::"+result+"</h1>");
        }
        else if(pval.equalsIgnoreCase("sub")) {
            float result=val1-val2;
            pw.println("<h1 style='color:red;text-align:center'>sub::"+result+"</h1>");
        }
        else if(pval.equalsIgnoreCase("mul")) {
            float result=val1*val2;
            pw.println("<h1 style='color:red;text-align:center'>Mul::"+result+"</h1>");
        }
        else if(pval.equalsIgnoreCase("div")) {
            float result=val1/val2;
            pw.println("<h1 style='color:red;text-align:center'>DIV::"+result+"</h1>");
        }
        else if(pval.equalsIgnoreCase("link1")) {
            pw.println("<b>System properites ::</b>"+System.getProperties());
        }
        else {
            pw.println("<h1 style='color:red;text-align:center'> Date And time :: "+new java.util.Date()+"</h1>");
        }

        //add hyperlink
        pw.println("<br> <h1 style='color:red;text-align:center'> <a href='form.html'>home </a> </h1>");

        //close PrintWriter
        pw.close();
    }//doGet(-,-)

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
      doGet(req,res);
    }//doPost(-,-)

}//class
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <welcome-file-list>
        <welcome-file>form.html</welcome-file>
    </welcome-file-list>

    <servlet>
    <servlet-name>process</servlet-name>
    <servlet-class>com.nt.servlet.ProcessServlet</servlet-class>
     </servlet>
     <servlet-mapping>
       <servlet-name>process</servlet-name>
       <url-pattern>/processurl</url-pattern>
     </servlet-mapping>
</web-app>
```

Form validations

=>Verifying the pattern and format of the form data is called form validations

=>This logic verifies form data format and pattern before it gives it gives form data to b.logic as input values

What is differecen b/w form validation logic and b.logic?

=>The form validation verifies the pattern and format of form data .. where b.logic uses form data as inputs and generates the results.

=> veryifying wheather text box values are given as numberic values or not is form validation logic taking text box values and performing addition ,subtraction is called b.logic.

=> Checking wheather given email id having "@" ,"." symbols is called form validation logic. checking given email is available in the Db s/w or not is b.logic

=>Checking credit card number is numberic value or not and having 16 digits or not is form validation logic performing payment using credit card is called b.logic

We can perform Form validations in two ways

1.Using Client side form validation logics

=>These logics are written in scripting languages like java script(best), vb script and etc..

=>These logics goes to browser along html form page code and executes in browser (client side). So these logics are called client side form validation logics

=>To execute JAva script code we need Java script engine.. which is part of browser (client)

2.Using Server Side form validation logics

=>These logics will be written as java code in servet,jsp comps before executing the b.logic or request processing logic

=>these logics are serverside logics becoz they execute in the server..

=>Decide wheather form validation logics are client side or server side based on the place where they are executing.. not based on the place where they are residing..

Oct 27 Form Validations

We can perform Form validations in two ways

1.Using Client side form validation logics

=>These logics are written in scripting languages like java script(best), vb script and etc..

=>These logics goes to browser along html form page code and executes in browser (client side). So these logics are called client side form validation logics

=>To execute JAva script code we need Java script engine.. which is part of browser (client)

2.Using Server Side form validation logics

=>These logics will be written as java code in servet,jsp comps before executing the b.logic or request processing logic

=>these logics are serverside logics becoz they execute in the server..

=>Decide wheather form validation logics are client side or server side based on the place where they are executing.. not based on the place where they are residing..

note: if form data is used in the b.logic directly with out validations.. then the b.logic may raise exeception or may generate results. To overcome the problems take the support of form validations..

Different approaches of performing form validations

Approach1: Write only Server Side Form Validation logics

Limitation:: if the form page is rejected for multiple times by server side form validation logics then there will be multiple network round trips between browser and server.

Approach2: Write only Client Side form validation logics using java script

advantage:: Since form validations are happening at client side using Java Script there will not be network rountrips between browser and server..

Limitation :: Using bowser settings , there is possibility of disabling JAva script code execution then no form validations takes place.

To disabe java script in chrom browser

... (menu) -->settings -->search for Javascript ---> go to java script

site settings-->

Don't allow sites to use Javascript

To disable java script in firefox browser

type about:config in address bar -->

accepting risk -->

javascript.enabled	false

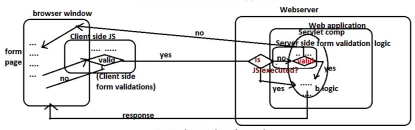
Approach3: Write both client side and server side form validations ..

advantages :: Even though client side Java script is disabled .. the server form valdations takes place to validate form data..

Limitaton :: if both client side and server side form validation logics are executed there is possibility of getting performance issue. Validating the form data at server side that already been validated at client side gives the performance issue.

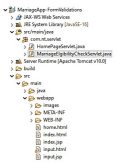
Approach4:: (Best) Write both Client side and server side form validations.. Enable server side form validations only when Client side form validations are not done.

advantage:: =>As long as client form side are performed the network roundtrips between browser and server will be reduced.
=>Since the server side form validations execute only when the client side form validations are not done.. there will not any performence issue.



Approach no:4 Flow of execution..

In order to display response content /error content in the same webpage from whom the request is generated we need to advanced java script called ajax.




```
//MarriageElgibilityCheckServlet.java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class MarriageElgibilityCheckServlet extends HttpServlet {

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        System.out.println("MarriageElgibilityCheckServlet.::doGet(-,-)");
     //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read form data
        String name=req.getParameter("pname");
        String tage=req.getParameter("page");
        String gen=req.getParameter("gender");
        // Server Form validation logics
        int age=0;
        List<String> errorsList=new ArrayList();
        if(name==null || name.length()==0 || name.equals(""))  //required rule
            errorsList.add("Person name is required");
        else if(name.length()<5)
            errorsList.add("Person name must have minimum of 5 characters");  //min length rule

        if(tage==null || tage.length()==0 || tage.equals("")) //required rule
            errorsList.add("Person age is required");
        else {
            try {
                age=Integer.parseInt(tage);
                if(age<=0 || age>125) {  // range rule
                    errorsList.add("Person age must be in the range of 1 through 125");
                }
            }//try
            catch(NumberFormatException nfe) {
                errorsList.add("Person  age must be numeric value");  // must be numeric value rule
            }
        }

        if(gen.equalsIgnoreCase("") || gen==null || gen.length()==0)
            errorsList.add("Person Gendor must be selected");

        //display form validation error messages
        if(errorsList.size()!=0) {
            pw.println("<ul>");
            for(String msg:errorsList) {
                pw.println("<li style='color:red'>"+msg+"</li>");
            }//for
            pw.println("</ul>");
            return;  //return stmt with out value returns the control from curent method definitation to caller.
        }//if

        //write  b.logic
        if(gen.equalsIgnoreCase("M")) {
            if(age>=21) {
                pw.println("<h1 style='color:blue;text-align:center'>Mr."+name+" u r eglible for marriage but think twice </h1>");
            }
            else {
                pw.println("<h1 style='color:red;text-align:center'>Mr."+name+" u r not eglible for marriage ..Be happy </h1>");
            }
        }
        else {
            if(age>=18) {
                pw.println("<h1 style='color:green;text-align:center'>Miss."+name+" u r eglible for marriage but think twice </h1>");
            }
            else {
                pw.println("<h1 style='color:red;text-align:center'>Miss."+name+" u r not eglible for marriage ..Be happy </h1>");
            }//else
        }//else
        //add home hyperlink as the graphical link
        pw.println("<br><br><a href='input.html'><img src='images/home.png'></a>");

        //clsoe stream
        pw.close();
    }//method

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        System.out.println("MarriageElgibilityCheckServlet.doGet()");
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //write logic
        Date d=new Date();
        pw.println("<h1 style='color:red;text-align:center' >"+d.toString()+"</h1>");
        //cose stream
        pw.close();
    }

}//class
```

---

what is the difference between java and java script?

The "word" Java is kept in java script only to take the advnatage of Java popularity

JAva	Java Script
a) It is programming language	a) It is scritpng language
b) This code can be executed directly with out embedding its code to other techologies code	b) For this code execution , the java script code must be embedded html code
c) It is Object oriented Programming language	c) It is objects based scripting laguage (Does not support inheritence and polymorphism)
d) To execute Java code we need JRE/JVM	d) To execute Java script code we need Java script engine
e) Generally used as server side webtechnology	e) Generally used as the cleint side web technology
f) It is Scrictly typed code	f) It is also..
g) can be used to develop all kinds of applications including web applicationss	g) Can be used only in web applications..
h) given by Sun ms (oracle corp)	h) Given by netscape + Sun Ms

VB script is given by MicroSoft and works only in MicroSoft browsers
JAva script is given by NEtScape and works in browser s/w.

Client side Form validations using Java script

Oct 29 Form Validations

java script events are onSubmit , onLoad , onBlur,onChange,onClick ,onDblClick and etc..

To create variable in JS: var <varname>; (or) let <var name> ;

No data types in JS .. based on the data that is assigned .. the data data type will be decied dynamically.

To create function in JS ::

```
function <function-name>( <param1>,<param2>,<param3> ,...){
  ....
  ...
  ...
}
```

JS supports DOM Programming (Document Object Model Programming) i.e every tag will be treated as object having properties and methods.

=>To make java script reusable across the multiple form pages of the web application.. it is better take that java script code in seperate .js file and include that content multuple .html files. (form pages)

validation.js

```
                    (a4)  (b4)
function validate(frm){

    //read form data
    var name=frm.pname.value;
    var  age=frm.page.value;                    (a5)  (b5)
    var gender=frm.gender.value;
    var flag=true;
    //Client Side form validaton logics
    if(name==""){  //required rule   (a6?:: true)   (b6:false)
        alert("Person name is required");
        flag=false;                                  (a7)
    }                          (b7:false)
    else if(name.length<5){ //min length rule        //lenght pre-defined js property
        alert("Person  name must have minimum of 5 characters");
        flag=false;
    }

    if(age==""){          (a8:true)   (b8:false)
        alert("Person age is required");
        flag=false;                                  (a9)
    }
    else if(isNaN(age)){    // isNaN pre-defined Js function .. isNaN (is Not a Number)   (b9:false)
        alert("Person age must be numeric value");
        flag=false;
    }
    else if(age<=0 || age>125){     (b10:false)
        alert("Person age must be there in the range of 1 through 125");
        flag=false;
    }

    if(gender==""){     (a10::true)     (b11:false)
        alert("Person gender  must be SELECTED");
        flag=false;
    }
     if(flag==false)                  (a11:true)   (b12:false)
   frm.pname.focus();
    return flag;  (a12)      (b13 --returns true)
}//function
```

input.html

```
<h1 style="color:red;text-align:center"> Html to Servlet communication using forms</h1>

<script language="JavaScript" src="js/validation.js">
                                                    (a3)  (b3)
</script>

<form action="marriageurl"  method="POST"  onsubmit="return validate(this)">
    <table border="0"  bgcolor="cyan" align="center">
      <tr>
        <td>Person name:: </td>
        <td><input type="text" name="pname"><span style="color:red">* </span> </td>
      </tr>
      <tr>
        <td>Person age:: </td>
        <td><input type="text" name="page"><span style="color:red">* </span> </td>
      </tr>
      <tr>
        <td>Person gender:: </td>
        <td>
          <input type="radio" name="gender"  value="M"> Male    
          <input type="radio" name="gender"  value="F" checked> FeMale    
          <span style="color:red">* </span>
        </td>
      </tr>

      <tr>                                          (a1)  (b1)
        <td> <input type="submit" value="check marriage elgibility"> </td>
        <td> <input type="reset" value="cancel"> </td>
      </tr>
    </table>
</form>
```

(a13) here return is statement mandatory to take return value of validate(this) function call that true/false and to pass that value browser .. to make browser to send the request to server or to block the request in browser self. if browser gets "false" then it indicates there are form validation errors and the requested will be blocked in browser itself. if browser gets "true" then it indidates there are no form validation errors and the request will go to server.

(b14) (a2) (b2)

a14:: browser gets flase so, the request is blocked in browser itself.

b15. browser gets true .. so request goes to server.

(a1) to a14 :: if person name,age values are given not

(b1) to b15: if person name,age values are given

note1:: alterboxes are old fashion to dispaly error messages .. prefer displaying error messages besides the text boxes using dynamic content generation technique with the support of innerHTML

=>altert dialogbox is bad becoz it blocks user activities until users clicks on ok button .. It is just good in debugging..

=> tag is given to apply styles on one line or small amount of text

=> tag is give to apply styles on multiple lines of text.

read these JS properties innerHTML, outerHTML

Oct 30 Form Validations

=>innerHTML , outerHTML are the properties of every html tag in JS to place the content to tag dynamically "innerHTML" replaces just tag body with dynamic content.. where as outerHTML replaces the total tag itsef new dyanmic content,.

```
<span id="id1" stye="color:red"> hello </span>

In JS
=====
  document.getEelementById("id1").innerHTML="abc";
```

replaces "hello" with "abc" i.e still <span> tag styles will be on "abc"

```
<span id="id1" stye="color:red"> hello </span>

In JS
=====
  document.getEelementById("id1").outerHTML="abc";
```

replaces total <span> tag including body with "abc" i.e no <span> tag styles will be on "abc"

improvised validation.js file
=========================

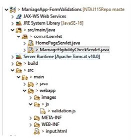
```
function validate(frm){

      //Empty the previous error messages
      document.getElementById("pnameErr").innerHTML="";
      document.getElementById("pageErr").innerHTML="";
      document.getElementById("genderErr").innerHTML="";

      //read form data
      var name=frm.pname.value;
      var  age=frm.page.value;
      var gender=frm.gender.value;
      var flag=true;
      //Client Side form validaton logics
      if(name==""){  //required rule
            document.getElementById("pnameErr").innerHTML="Person name is required";
            flag=false;
      }
      else if(name.length<5){ //min length rule
            document.getElementById("pnameErr").innerHTML="Person  name must have minimum of 5 characters";
            flag=false;
      }

      if(age==""){
            document.getElementById("pageErr").innerHTML="Person age is required";
            flag=false;
      }
      else if(isNaN(age)){
            document.getElementById("pageErr").innerHTML="Person age must be numeric value";
            flag=false;
      }
      else if(age<=0 || age>125){
            document.getElementById("pageErr").innerHTML="Person age must there 1 through  125";
            flag=false;
      }

      if(gender==""){
            document.getElementById("genderErr").innerHTML="Person Gender must be selected";
            flag=false;
      }
       if(flag==false)
     frm.pname.focus();

      return flag;
}
```

we can alter enduser to enable java script by using <noscript> tag

```
<h1 style="color:red;text-align:center"><noscript> Enable  Java Script</noscript> </h1>
```

This tag content will be displayed on the browser only when JAva Script code execution is disabled.

The best pratice of form validations is .. enable server side form validations only when Client side form validations are not done. For this we need to take one hidden box in form page having value "no" .. change that value to "yes" in Java script code which means "no" --> client form validations are not done.. "yes" -- client side form validations are done Read this hiddent box value in servlet comp as req param value and enable server side form validations only if the the recieved value is "no".



input.html (c) (3)
=========

```
<h1 style="color:red;text-align:center"> Html to Servlet communication using forms</h1>
 <h1 style="color:red;text-align:center"><noscript>  Enable  Java Script</noscript> </h1>
<script language="JavaScript" src="js/validation.js">

</script>
                         (h)  (5)                                  (d)
 <form action="marriageurl"  method="POST"  onsubmit="return validate(this)">
   <table border="0"  bgcolor="cyan" align="center">    (g)
     <tr>
       <td>Person name:: </td>
       <td><input type="text" name="pname"><span style="color:red"  id="pnameErr">* </span> </td>
     </tr>
     <tr>
       <td>Person age:: </td>
       <td><input type="text" name="page"><span style="color:red" id="pageErr">* </span> </td>
     </tr>
     <tr>
       <td>Person gender::  </td>
       <td>
         <input type="radio" name="gender"  value="M"> Male    
         <input type="radio" name="gender"  value="F" checked> FeMale    
         <span style="color:red" id="genderErr">* </span>
        </td>
     </tr>

     <tr>                      (c)    (4)
       <td> <input type="submit" value="check marriage elgibility"> </td>
       <td> <input type="reset" value="cancel"> </td>
     </tr>
   </table>
    <input type="hidden" name="vflag"  value="no">
 </form>
```

validation.js
==========   (e)

```
function validate(frm){
      alert("Client side form validations");

      //Empty the previous error messages
      document.getElementById("pnameErr").innerHTML="";
      document.getElementById("pageErr").innerHTML="";
      document.getElementById("genderErr").innerHTML="";

      //read form data
      var name=frm.pname.value;
      var  age=frm.page.value;
      var gender=frm.gender.value;
      var flag=true;
      //Client Side form validaton logics
      if(name==""){  //required rule
            document.getElementById("pnameErr").innerHTML="Person name is required";
            flag=false;
      }
      else if(name.length<5){ //min length rule
            document.getElementById("pnameErr").innerHTML="Person  name must have minimum of 5 characters";
            flag=false;
      }

      if(age==""){
            document.getElementById("pageErr").innerHTML="Person age is required";
            flag=false;
      }
      else if(isNaN(age)){
            document.getElementById("pageErr").innerHTML="Person age must be numeric value";
            flag=false;
      }
      else if(age<=0 || age>125){
            document.getElementById("pageErr").innerHTML="Person age must there 1 through  125";
            flag=false;
      }

      if(gender==""){
            document.getElementById("genderErr").innerHTML="Person Gender must be selected";
            flag=false;
      }
       if(flag==false)
     frm.pname.focus();
    //change hidden box value to "yes" indicating client side java script validations are done
     frm.vflag.value="yes";

         return flag;   (f) //return true
  }
```

i2ee/web-app_2_

```
//MarriageElgibilityCheckServlet.java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
              (j) Sevlet object creating or locating  (7)
public class MarriageElgibilityCheckServlet extends HttpServlet {

    @Override   (k)    (8)
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
          System.out.println("MarriageElgibilityCheckServlet.::doGet(-,-)");
    //get PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read form data
            String name=req.getParameter("pname");
            String tage=req.getParameter("page");
            String gen=req.getParameter("gender");
            String  vstatus=req.getParameter("vflag");
           //  enable Server side Form validation logics only when client side form validations are not done
            int age=0;
      if(vstatus.equalsIgnoreCase("no")) {      (l?)  (9?)
            List<String> errorsList=new ArrayList();
            System.out.println("Server side form validations ");
             if(name==null || name.length()==0 || name.equals(""))  //required rule
                  errorsList.add("Person name is required");
             else if(name.length()<5)
                  errorsList.add("Person name must have minimum of 5 characters");   //min length rule

             if(tage==null || tage.length()==0 || tage.equals("")) //required rule
                  errorsList.add("Person age is required");
             else {
                    try {
                           age=Integer.parseInt(tage);
                           if(age<=0 || age>125) {   // range rule
                                  errorsList.add("Person age must be in the range of 1 through 125");
                           }
                    }//try
                    catch(NumberFormatException nfe) {
                           errorsList.add("Person  age must be numeric value");  // must be numeric value rule
                    }
             }

(10)         if(gen.equalsIgnoreCase("") || gen==null || gen.length()==0)  //required
                  errorsList.add("Person Gendor must be selected");

             //display form validation error messages
             if(errorsList.size()!=0) {
                   pw.println("<ul>");
                  for(String msg:errorsList) {
                         pw.println("<li style='color:red'>"+msg+"</li>");
                  }//for
                   pw.println("</ul>");
                   return;   //return stmt with out value returns the control from curent method definitation to caller.
             }//if
      }//if
      else {
            age=Integer.parseInt(tage);      (m)
      }

            //write  b.logic
            if(gen.equalsIgnoreCase("M")) {
                   if(age>=21) {
                        pw.println("<h1 style='color:blue;text-align:center'>Mr."+name+" u r eglible for marriage but think twice </h1>");
                   }
                   else {
                        pw.println("<h1 style='color:red;text-align:center'>Mr."+name+" u r not eglible for marriage ..Be happy </h1>");
                   }
            }
(n)         else {
                  if(age>=18) {
(11)                    pw.println("<h1 style='color:green;text-align:center'>Miss."+name+" u r eglible for marriage but think twice </h1>");
                  }
                   else {
                        pw.println("<h1 style='color:red;text-align:center'>Miss."+name+" u r not eglible for marriage ..Be happy </h1>");
                   }//else
            }//else
            //add home hyperlink as the graphical link
            pw.println("<br><br><a href='input.html'><img src='images/home.png'></a>");

            //clsoe stream
            pw.close();
    }//method

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
          System.out.println("MarriageElgibilityCheckServlet.doGet()");
          //get PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //write logic
            Date d=new Date();
            pw.println("<h1 style='color:red;text-align:center' >"+d.toString()+"</h1>");
            //cose stream
            pw.close();
    }

}//class
```

web.xml
==========

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

     <welcome-file-list>
        <welcome-file>input.html</welcome-file>
(b)    <welcome-file>homeurl</welcome-file>
(2)     <welcome-file>home.html</welcome-file>
        <welcome-file>input.jsp</welcome-file>
     </welcome-file-list>

      <servlet>
        <servlet-name>marriage</servlet-name>
        <servlet-class>com.nt.servlet.MarriageElgibilityCheckServlet</servlet-class>
      </servlet>
(i)   <servlet-mapping>
(6)     <servlet-name>marriage</servlet-name>
        <url-pattern>/marriageurl</url-pattern>
      </servlet-mapping>


</web-app>
```

(1)
request url :: http://localhost:MarriageApp-FormValidations (a)

w.r.diagram (a) to n :: request1 giving perfect form data by enabling client side JS

(1) to (11) :: request2 giving perfect form data by not enabling client side JS

Working with different types of form comps

Nov 01 Woring with different types of form comps

textbox
passwordbox
radio button
grouped radio buttons
check box
grouped checkboxes
text area
hidden box
select box/combo box /drop down box
file uploading
List box / Multi Select box
Standard button
Reset button
Submit button

tradtional
html form comps

Html 5 form comps

"color"
"date"
"datetime-local"
"datetime"
"email"
"image"
"month"
"number"
"radio"
"range"
"search"
"tel"
"time"
"url"
"week"

These are advanced comps with built-in validators

<input type="email"> Gives text box having built-in validator support.

=>If form comp is giving single value as the req param value then use req.getParameter(-)(return type is String) to read that value
(For most of the comps we can use that method)
=>if form comp is giving multiple values as the req param values then use req.getParameterValues(-) (return type is String[]) to read those values (For grouped checkboxes, select box , list box comps)

note: By giving same name to multiple radio buttons we can group them in to single unit...
If multiple radions are kept in a group .. then we can select one radio button at a time..
If multiple checkboxes are kept in a group then we can select 0 or more chekcboxes.

In form page

eg:: Gender :: <input type="radio" name="gen" value="M">Male
<input type="radio" name="gen" value="F" checked >FeMaile

Gender :: Male FeMaile

In servlet comp

String gender=req.getParameter("gen"); //gives F

In form page

Hobies : <input type="checkbox" name="hb" value="reading">ReadingBooks
<input type="checkbox" name="hb" value="TV" checked> Watching TV
<input type="checkbox" name="hb" value="playing" checked> Playing cricket
<input type="checkbox" name="hb" value="airforce"> Travelling

Hobies : [ ] ReadingBooks [x] watching TV
[x] PlayingCricket [ ] Travelling

In servlet comp

String hobies[]=req.getParameterValues("hb");

hobies[]
TV 0
palying 1

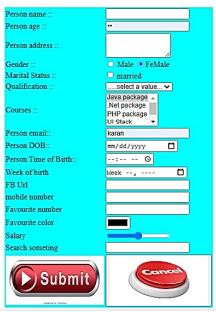
example App

AllFormCompsApp
JAX-WS Web Services
JRE System Library [JavaSE-16]
src/main/java
com.nt.servlet
FormCompServlet.java
Server Runtime [Apache Tomcat v10.0]
build
src
main
java
webapp
images
cancel.jpeg
submit.jpeg
META-INF
WEB-INF
lib
web.xml
person_details.html

person_details.html

```
<h1 style="color:red;text-align:center"> Working all form comps in form page</h1>
<form action="formurl" method="POST">
 <table align="center" bgcolor="cyan">
  <tr>
   <td> Person name :: </td>
   <td> <input type="s" name="pname"> </td>
  </tr>
  <tr>
   <td> Person age :: </td>
   <td> <input type="password" name="page"> </td>
  </tr>
  <tr>
   <td> Person address :: </td>
   <td> <textarea rows="3" cols="14">

        </textarea> </td>
  </tr>
  <tr>
   <td> Gender :: </td>
   <td> <input type="radio" name="gender" value="M"> Male
        <input type="radio" name="gender" value="F" checked>FeMale
   </td>
  </tr>
  <tr>
   <td> Marital Status :: </td>
   <td> <input type="checkbox" name="ms" value="married"> married
   </td>
  </tr>
  <tr>
   <td> Qualification :: </td>
   <td>
       <select name="qlfy" >
          <option value="" selected> .....select a value... </option>
          <option value="engineer">B.E/B.Tech </option>
          <option value="medico">MBBS/MD </option>
          <option value="arts">BA/B.Btech </option>
          <option value="science">B.sc /M.sc </option>
       </select>
   </td>
  </tr>
  <tr>
   <td> Courses :: </td>
   <td>
      <select name="crs" multiple >
         <option value="Java" selected> Java package </option>
         <option value=".net">.Net package </option>
         <option value="php">PHP package</option>
         <option value="UI">UI Stack </option>
         <option value="spring" selected>Spring STack </option>
      </select>
   </td>
  </tr>
  <!-- advanced comps --->
  <tr>
   <td> Person email:: </td>
   <td> <input type="email" name="email"> </td>
  </tr>
  <tr>
   <td> Person DOB:: </td>
   <td> <input type="date" name="dob"> </td>
  </tr>
  <tr>
   <td> Person Time of Birth:: </td>
   <td> <input type="time" name="tob"> </td>
  </tr>
  <tr>
   <td> Week of birth </td>
   <td> <input type="week" name="wb"> </td>
  </tr>
  <tr>
   <td> FB Url </td>
   <td> <input type="url" name="fbUrl"> </td>
  </tr>
  <tr>
   <td> mobile number </td>
   <td> <input type="tel" name="mobile No"> </td>
  </tr>
  <tr>
   <td> Favourite number </td>
   <td> <input type="number" name="favNumber"> </td>
  </tr>

  <tr>
   <td> Favourite color </td>
   <td> <input type="color" name="favColor"> </td>
  </tr>

  <tr>
   <td> Salary </td>
   <td> <input type="range" name="salary" min="20000" max="100000"/>
  </tr>

  <tr>
   <td> Search someting </td>
   <td> <input type="search" name="itemSearch"/>
  </tr>

  <tr>
   <td> <!-- <input type="sumbit" value=" process Request"> -->
        <input type="image" src="images/submit.jpeg" width="200" height="100" >
   </td>
   <td> <!-- <input type="reset" value="cancel" /> -->
        <button type="reset"><img src="images/cancel.jpeg" width="200" height="100"></button>
   </td>
  </tr>

 </table>

</form>
```

Working all form comps in form page



Nov 02 Woring with different types of form comps

between

What the differecne tradtional html comps and HTML5 comps

=>raditional comps does not provide built-in form validations and these are basic form comps

=>Html5 comps comps advanced comps and most of the comps are providing built-in validations

=>The check box (single or group) and List box (Mutliselect box) does not form request parameters if their items are not slected .. So we must non-selected from them to avoid NullPointerException, NumberFormatException from the Application.. With remaining comps that problem is not there becoz they form reqparams atleast with empty "" though they not selected or filledup.

```
String ms=req.getParameter("ms");
String courses[]=req.getParameterValues("crs");

    //provide non-select state for cheboxes and list boxes
        ms=(ms==null)?"single":ms;
        courses=(courses==null)?new String[] {"No courses are seelted"}:courses;
```

Ternary operator is used here.

example App

Person_details.html

```
<h1 style="color:red;text-align:center"> Working all form comps in form page</h1>
<form action="/formurl" method="GET">
  <table align="center" bgcolor="cyan">
    <tr>
      <td> Person name ::  </td>
      <td> <input type="x" name="pname"> </td>
    </tr>
    <tr>
      <td> Person age ::  </td>
      <td> <input type="password" name="page"> </td>
    </tr>
    <tr>
      <td> Person address ::  </td>
      <td> <textarea name="paddrs" rows="3" cols="14">

          </textarea> </td>
    </tr>
    <tr>
      <td>  Gender ::  </td>
      <td> <input type="radio" name="gender" value="M"> Male
           <input type="radio" name="gender" value="F" checked>FeMale
      </td>
    </tr>
    <tr>
      <td> Marital Status ::  </td>
      <td> <input type="checkbox" name="ms" value="married"> married
      </td>
    </tr>
    <tr>
      <td> Qualification ::  </td>
      <td>
          <select name="qlfy" >
             <option value="" selected> .....select a value... </option>
             <option value="engineer">B.E/B.Tech </option>
             <option value="medico">MBBS/MD </option>
             <option value="arts">BA/B.Btech </option>
             <option value="science">B.sc /M.sc </option>
          </select>
      </td>
    </tr>
    <tr>
      <td> Courses ::  </td>
      <td>
          <select name="crs" multiple >
             <option value="Java" selected> Java package </option>
             <option value=".net">.Net package </option>
             <option value="php">PHP package</option>
             <option value="UI">UI Stack </option>
             <option value="spring" selected>Spring STack </option>
          </select>
      </td>
    </tr>
    <!-- advanced comps --->
    <tr>
      <td> Person email::  </td>
      <td> <input type="email" name="email"> </td>
    </tr>
    <tr>
      <td> Person DOB::  </td>
      <td> <input type="date" name="dob"> </td>
    </tr>
    <tr>
      <td> Person Time of Birth::  </td>
      <td> <input type="time" name="tob"> </td>
    </tr>
    <tr>
      <td> Week of birth  </td>
      <td> <input type="week" name="wb"> </td>
    </tr>
    <tr>
      <td> FB Url  </td>
      <td> <input type="url" name="fbUrl"> </td>
    </tr>
    <tr>
      <td> mobile number  </td>
      <td> <input type="tel" name="mobileNo"> </td>
    </tr>
    <tr>
      <td> Favourite number  </td>
      <td> <input type="number" name="favNo"> </td>
    </tr>

    <tr>
      <td> Favourite color  </td>
      <td> <input type="color" name="favColor"> </td>
    </tr>

    <tr>
      <td> Salary  </td>
      <td> <input type="range" name="salary" min="20000" max="100000"/>
    </tr>

    <tr>
      <td> Search someting  </td>
      <td> <input type="search" name="itemSearch"/>
    </tr>

    <tr>
      <td> <!-- <input type="sumbit" value=" process Request"> -->
           <input type="image" src="images/submit.jpeg" width="200" height="100" >
      </td>
      <td> <!-- <input type="reset" value="cancel" /> -->
           <button type="reset"><img src="images/cancel.jpeg" width="200" height="100"></button>
      </td>
    </tr>
  </table>
```

sevlet comp

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Arrays;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class FormCompsServlet extends HttpServlet {
  @Override
  public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        // get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response cotent type
        res.setContentType("text/html");
        //read form data
        String name=req.getParameter("pname");
        int age=Integer.parseInt(req.getParameter("page"));
        String addrs=req.getParameter("paddrs");
        String ms=req.getParameter("ms");
        String gender=req.getParameter("gender");
        String  qlfy=req.getParameter("qlfy");
        String courses[]=req.getParameterValues("crs");
        String mail=req.getParameter("email");
        String dob=req.getParameter("dob");
        String tob=req.getParameter("tob");
        String  wob=req.getParameter("wb");
        String fbUrl=req.getParameter("fbUrl");
        long mobileNo=Long.parseLong(req.getParameter("mobileNo"));
        int favNo=Integer.parseInt(req.getParameter("favNo"));
        String favColor=req.getParameter("favColor");
        float salary=Float.parseFloat(req.getParameter("salary"));
        String item=req.getParameter("itemSearch");

        //provide non-select state for cheboxes and list boxes
        ms=(ms==null)?"single":ms;
        courses=(courses==null)?new String[] {"No courses are seelted"}:courses;

        //write b.logic
        if(gender.equalsIgnoreCase("M")) {
              if(age<5) {
                  pw.println("<h1 style='color:red;text-align:center'>Master."+name+"  u  r  baby boy </h1>");
              }
              else if(age<12) {
                  pw.println("<h1 style='color:red;text-align:center'>Master."+name+"  u  r small boy </h1>");
              }
              else if(age<19) {
                  pw.println("<h1 style='color:red;text-align:center'>Mr."+name+"  u  r teenage boy </h1>");
              }
              else if(age<35) {
                  pw.println("<h1 style='color:red;text-align:center'>Mr."+name+"  u  r young man </h1>");
              }
              else if(age<50) {
                  pw.println("<h1 style='color:red;text-align:center'>Mr."+name+"  u  r middle-aged man </h1>");
              }
              else {
                  pw.println("<h1 style='color:red;text-align:center'>Mr."+name+"  u  r old man </h1>");
              }
        }
        else {
              if(age<5) {
                  pw.println("<h1 style='color:red;text-align:center'>Master."+name+"  u  r  baby girl </h1>");
              }
              else if(age<12) {
                  pw.println("<h1 style='color:red;text-align:center'>Master."+name+"  u  r small girl </h1>");
              }
              else if(age<19) {
                  if(ms.equalsIgnoreCase("married"))
                    pw.println("<h1 style='color:red;text-align:center'>Mrs."+name+"  u  r teenage married girl </h1>");
                  else
                        pw.println("<h1 style='color:red;text-align:center'>Miss."+name+"  u  r  teenage  girl </h1>");
              }
              else if(age<35) {
                  if(ms.equalsIgnoreCase("married"))
                    pw.println("<h1 style='color:red;text-align:center'>Mrs."+name+"  u  r young  married  woman </h1>");
                  else
                        pw.println("<h1 style='color:red;text-align:center'>Miss."+name+"  u  r young   woman </h1>");
              }
              else if(age<50) {
                  if(ms.equalsIgnoreCase("married"))
                  pw.println("<h1 style='color:red;text-align:center'>Mrs."+name+"  u  r middle-aged  married woman </h1>");
                  else
                        pw.println("<h1 style='color:red;text-align:center'>Miss."+name+"  u  r middle-aged   woman </h1>");
              }
              else {
                  if(ms.equalsIgnoreCase("married"))
                    pw.println("<h1 style='color:red;text-align:center'>Mrs."+name+"  u  r married dadi </h1>");
                  else
                        pw.println("<h1 style='color:red;text-align:center'>Miss."+name+"  u  r buddi </h1>");
              }
        }//else
              //display form data
              pw.println("<h1 style='color:red;text-align:center'> Form data is  </h1>");
              pw.println("<b> name :: "+name+"</b><br>");
              pw.println("<b> age :: "+age+"</b><br>");
              pw.println("<b> addrs :: "+addrs+"</b><br>");
              pw.println("<b> qualification :: "+qlfy+"</b><br>");
              pw.println("<b> maritial status :: "+ms+"</b><br>");
              pw.println("<b> courses :: "+Arrays.toString(courses)+"</b><br>");
              pw.println("<b> email :: "+mail+"</b><br>");
              pw.println("<b> DOB :: "+dob+"</b><br>");
              pw.println("<b> TOB :: "+tob+"</b><br>");
              pw.println("<b> WOB :: "+wob+"</b><br>");
              pw.println("<b> mobileNo :: "+mobileNo+"</b><br>");
              pw.println("<b> faviourite No :: "+favNo+"</b><br>");
              pw.println("<b> faviourite Color :: "+favColor+"</b><br>");
              pw.println("<b> Salary :: "+salary+"</b><br>");
              pw.println("<b> Search Item :: "+item+"</b><br>");

              //add home hyperlink
              pw.println("<a href='person_details.html'>home</a>");


  }

  @Override
  public void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        doGet(req,resp);
  }
}
```

web.xml

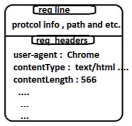
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <welcome-file-list>
        <welcome-file>person_details.html</welcome-file>
    </welcome-file-list>
    <servlet>
      <servlet-name>form</servlet-name>
      <servlet-class>com.nt.servlet.FormCompsServlet</servlet-class>
    </servlet>
    <servlet-mapping>
      <servlet-name>form</servlet-name>
      <url-pattern>/formurl</url-pattern>
    </servlet-mapping>
</web-app>
```

Nov 03 Http Method GET vs POST

request given by protocol "http" will have standard structure .
The GET mode request structure does not contain body becoz , the
request inputs (form data representing req params) goes to server as query String data appended to
the url

http://localhost:3030/MarriageApp?pname=raja&page=30&ms=married

GET mode request structure



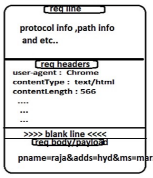
Request headers and values are automatically
added addtional information to request about the browser
s/w and client machine.

Request header names are fixed and values
will be generated based on browser s/w and client machines we use.

No request body becoz
form data is going to server
as req param value

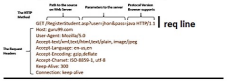
=>The POST mode request carries the request inputs (form data) in the form of request body .. So
no query String appears in the browser address bad appended to the request url

POST mode request structure



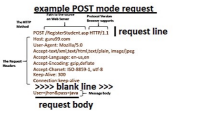
=>Since form data goes to server
as request body .. So no queery
appears in browser's address bar

example GET method request



---

example POST mode request



How to find out the browser s/w name from which
the request is given to servlet comp/web comp?

Ans) Take the support the request header "user-agent" in servlet comp
as shown below

```
String brName=req.getHeader("user-agent");
```

What is the difference b/w request parameters and request headers?

request parameters	request headers
a) generaally represents form data given in the request by enduser	a) represents browser supplied addtional data having more info browser and client machine
b) These are optional in the request	b) will be added automatically to request (indirecly mandatory)
c) request params are developer choice name	c) req header names are fixed names given by protocol "http" standards
d) request param names can be taken as duplicate names	d) request header names are unique names
e) One request param can have multiple value	e) one request header can having only one value
f) In GET mode request these are in the form of query String .. In Post mode request these are in the form of request body	f) Always part of the request structure in the middle.
g) To read request param value we can use ServletRequest object or HttpServletRequest object	g) we must need HttpServletRequest object
h) To read req param values use req.getParameter(-) method	h) To read req header values use req.getHeader(-) method

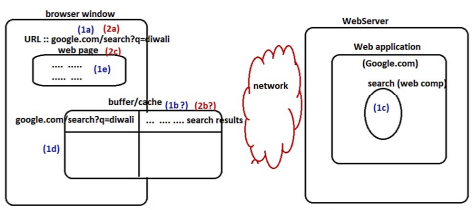
---

What is the difference GET mode request and POST mode request?

GET	POST
a) it is default mode of the request	a) It is not a default mode of the request
b) It is given to get or to query the data from server	b) It is given to send/submit data to server
c)Can carry limitted amout of data along with the request (2k to 8kb) (In most of browsers it is 2kb)	c) Can carry unlimited amount of data along with the request
d) Not suitable for file uploading	d) Suitable for file uploading
e) form data goes to server as query string appended to the url	e) Form data goes to server as request body /pay load
f) The Http Request strcuture does not request body	f) The http request structure contains request body
g) There is no data secracry as form data appears the in the address bar as query string	g) There is data secracy with POST mode request as form data does not appear in the browser address bar
h) Hacking is possible	h) hacking can be prevented
i) Can carry only text data along with the request	i) can carry both text data and binary data (images, audio,vedio ,exe files and etc..)
j) GET mode request is ReadOnly request	j) POST mode request is update request is
k) GET mode request is idem potent (safe to repeat the request)	k) POST mode request non-idempotent (Not saft to repeat request)
l) GET mode requests support book marking	l) Does not support book marking effectivley
m) Supports caching (browser level)	m) does not caching support effectively.
n) To process this mode request use doGet(-,-) method	n) To process this mode rquest use doPost(-,-) method
o) To generate GET mode request a) type url in the browser address bar b) using hyperlink ... c) using <form> with out any mode <form action="...." > ... <input type="submit" ....> </form> d) Using <form> with GET mode <form action="....." method="GET"> .... <input type="submit" > </form>	o) To generate POST mode request <form action="....." method="POST"> .... <input type="submit" > </form>

Cache/Buffer is a tempory memory that holds data for temporary period and uses that data for multiple times

=>The cache that is avalable with browser holds server supplied data for urls +queryStrings and uses that
data across the multiple requests.. This reduces network round trips b/w browser and server(web application)



1a to 1e :: 1st request
2a to 2e : 2nd request

Both requets are
same having same url and query String.

=>To take full benifit buffer /cache we need to empty the browser cache at requral intervals or we need to
disable browser buffer or cache for web comps like Live Game score comps, Stock share value comps and etc..

To disable browser cache/buffer being from servlet comp

```
res.setHeader("cache-control", "no-cache");  // In http 1.1 env.. (latest)
        (or)
res.setHeader("pragma", "no-cache");  // In http 1.0 env.. (old)
```

To empty the cache at regular intervals

```
res.setDateHeader("expries",20);  // for every 20 secs
                                  cache content will be emptied.
```

if the response content of web comp
is changing at regular intervals then prefer
disabling browser cache.
eg:: live game scores ,
stock market share values and etc..

if the response content of web comp
is not changing at requalar internal then prefer not
disabling browser cache
eg:: google search results, history of toursit places and etc..

Nov 06 ServletLifeCycle

Generally 1st request given servlet comp participates in
->Servlet loading
->Servlet instantiation
->Servlet initialization
->request processing and response to browser
other than 1st request given servlet comp participates in
->request processing and response to browser

-> The response time(The time take to process the request and to generate the response) 1st request is bit more compare to other than 1st request.

=>By enabling pre-instanation of servlet using <load-on-startup> we make Servletcontainer performing servlet loadig, servlet instanitation and servlet initialization activities either during server startup and during the deployment of web application.

In web.xml

```
<servlet>
  <servlet-name>lc</servlet-name>
  <servlet-class>com.nt.servlet.LcTestServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>lc</servlet-name>
  <url-pattern>/lcurl</url-pattern>
</servlet-mapping>
```
<load-on-startup> priority value

[the <load-on-startup> priority value can be 0 or positive number .Enabling <load-on-startup> by taking -ve value as the priority value Ignores the <load-on-startup>]

For the above sevlet comp
==========
During server startup or during the deployment of the web application
---
-> servlet loading
-> servlet instantiation
-> servlet initialization

When 1st request is given
->request processing and response generation to browser
When other than 1st request is given
->request processing and response generation to browser

Here 1st response time is reduced and equalized with other than 1st request.

=>By default servletcontainer creates Servlet comp class object through lazy instanitation process becoz it creates servlet comp class object lazily only when 1st request is given

=>By enabling <load-on-startup> on servlet comp , the servletcontainer creates Servlet comp class object through eager/pre-instanitation process becoz it creates servlet comp class object eagely either during server startup or during the deployment of the web webapplication.

=>It is not mandatory that every servlet comp must be enabled with <load-on-startup> ..It is recomanded to enable <load-on-startup> only on that servlet comps which will be requested immediately after deployment.
eg: servlet comp giving home page /first page
eg: servlet comp giving main menu page

=>if multiple servlet comps of a web application are enabled with <load-on-startup> then their order pre-instantiation either during server startup or during the deployment of web application will be decided based on the <load-on-startup> priority value.

High value indicates low priority
Low value indicates high priority
-ve value ignores the enabled <load-on-startup>

Servlet1,Servlet2,Servlet3,Servlet4 are the 4 web comps of the web application

	<load-on-startup> value	<load-on-startup> value	<load-on-startup> value	
Servlet1	10 (IV)	5 (III)	10	
seervlet2	5 (I)	0 (I)	10	Server decides the order using its own algorithm
servlet3	7 (III)	1 (II)	10	
Servlet4	6 (II)	-10 (Ignores the <load-on-startup>)	10	

=>No default value for <load-on-startup> priority .... <load-on-startup> tag must taken with priority value.

In web.xml
==========

```
<servlet>
  <servlet-name>lc</servlet-name>
  <servlet-class>com.nt.servlet.LcTestServlet</servlet-class>
  <load-on-startup>1</load-on-startup>   (2)
</servlet>
<servlet-mapping>
  <servlet-name>lc</servlet-name>
  <url-pattern>/lcurl</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>html</servlet-name>
    <servlet-class>com.nt.servlet.HtmlServlet</servlet-class>
    <load-on-startup>0</load-on-startup>   (1)
</servlet>
<servlet-mapping>
  <servlet-name>html</servlet-name>
  <url-pattern>/htmlgames</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>plain</servlet-name>
    <servlet-class>com.nt.servlet.PlainServlet</servlet-class>
    <load-on-startup>10</load-on-startup>   (5)
</servlet>
<servlet-mapping>
  <servlet-name>plain</servlet-name>
  <url-pattern>/plaingames</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>word</servlet-name>
    <servlet-class>com.nt.servlet.WordServlet</servlet-class>
    <load-on-startup>4</load-on-startup>   (3)
</servlet>
<servlet-mapping>
  <servlet-name>word</servlet-name>
  <url-pattern>/wordgames</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>excel</servlet-name>
    <servlet-class>com.nt.servlet.ExcelServlet</servlet-class>
    <load-on-startup>7</load-on-startup>   (4)
</servlet>
<servlet-mapping>
  <servlet-name>excel</servlet-name>
  <url-pattern>/excelgames</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>xml</servlet-name>
    <servlet-class>com.nt.servlet.XmlServlet</servlet-class>
    <load-on-startup>-1</load-on-startup>   (Ignores <load-on-startup> )
</servlet>
<servlet-mapping>
  <servlet-name>xml</servlet-name>
  <url-pattern>/xmlgames</url-pattern>
</servlet-mapping>
```

s.o.p message on server console

```
HtmlServlet: static block
HtmlServlet:: 0-param constructor
HtmlServlet::init(cg)
LcTestServlet:: static block
LcTestServlet:: 0-param constructor
LcTestServlet: init(-) method
WordServlet: static block
WordServlet:: 0-param constructor
WordServlet::init(cg)
ExcelServlet: static block
ExcelServlet:: 0-param constructor
ExcelServlet::init(cg)
PlainServlet: static block
PlainServlet:: 0-param constructor
PlainServlet::init(cg)
```

=>Serlet container is using init(-) to create our servlet comp class obj?

Ans) No.. Servlet Container internally uses its own reflcation api based logic to create our servlet comp class object .. init(-) life cycle method is just avaiable only to place initialization logics of servlet comp

=>Does Servlet container is using destroy() method to destroy our servlet class object?
Ans) No .... Servletcontainer its own GarbageCollector to destroy our sevlet comp class object.

init(cg), destroy() are the two life cycle methods that can be overridden in servlet comp to place the programmer choice initializiation and uninitialization logics .. the logics of both methods are not resposible to create create or destroy our servlet class object.

when servlet container creates our servlet comp class object?

if <load-on-startup> is not enabled
---
a) For the first request given to servlet comp
b) For the first request given to servlet comp after reloading ,redeploying and restarting the web application
c) For the first requst given to servlet comp after restarting the server
when <load-on-startup> is enabled
==========
a) Eigher during server startup and during the deployment of web application

When ServletContainer destroys our servlet class obj?

a) if the web application is stopped/reloaded /undeployed
b) if the underlying server is restarted /crashed
c) if our servlet class obj is continuosly idle from long time

note: No Special priority towards dustruction of servlet calss objs even <load-on-startup> is enabled on those servlet comps.

Nov 08 ServletLifeCycle

=> Is init(-) method is creating our servlet class object or not?

Ans) No .. init() method is not creating our servlet class object.. SErvletcontainer intenrally uses reflection api based newInstance() method for instantiation(object creation) .. init(-) is useful to place servlet initialization logic like creating jdbc con object

=> Is destroy() method is destroying our servlet class object or not?

Ans) No ..destroy() method is not destroying our servlet class object.. SErvletcontainer intenrally uses its own garbage collector for destroying our servlet class object .. destroy(-) is useful to place servlet unitialization logic like closing jdbc con object.

What happens if call destroy() method from service(-) method?

Ans) our servlet class object will not be destroyed.. but the logics destroy() method executes along with service(-,-) method..

->When Event raised ,the event handling method will be executed automatically.. but the even will not be raised since we have called event handling method manually.

->when ServletContainer is about to destroy our servlet class object.. the destruction event will be raised Since we have called destroy() method manually .. the ServletCotainer does not raise destruction event .

what happens if we call init(-) from service(-) method?

Ans) our servlet class object will not be created for every request.. but the logics of init(-) method exeecutes along with service(-) method.

Can we place main(-) method in our servlet comp?

Ans) yes , we can place .. but will not be called automatically.. becoz Servletcontainer executes servlet comp through life cycle methods and main(-) is not a life cycle method of servlet comp.

How does the servlet comp is executing with out main(-) method?

Ans1) JVM directly begins the execution of standalone java app by calling main(-) method .. since Servlet comp is not standalone java app .. and it is web comp of java web application.. So do not main(-) method in servlet comp ..

Ans2) ServletContainer executes the Servlet comp automatically for the given request .. by calling life cycle methods. Since main(-) is not life cycle method of servlet comp.. So we do not main(-) method ins servlet comp..

Ans3) When webSErver started class with main(-) executes ---> main(-) starts servletcontainer as deamon process and this servlet container loads ,instantiates our servlet comp class and calls various life cycle methods in the execution. In any application /project only one class will have main(-) method and remaining helper classes or supporting classes do not need to have main(-) method..

In tomcat server
==========

When we run <tomcat_home>\bin\tomcat<ver>.exe file .. it runs internally <Tomcat_home>\bin\boostrap.jar file .. In the process it executes org.apache.catalina.startup.Boostrap class by calling its main(-) method .. This main(-) method interally starts servlet container as daemon process ---->this servletContainer executes the servlet comp by calling various life cycle methods on its object.

What is singleton java class .. Is our servlet comp is singleton java class or not?

Ans) => the java class that allows to create only one object in any given situation is called singleton Java class i.e we can not create more than 1 object for singleton java class.. if we attempt to create more objs then it returns existing object.

=>Though class is allowing to create more objects.. if u r just happy with one object creation then it is not called singleton java class.

=> generally Servlet container creates only 1 object for our servlet comp class .. and we do no create any object for servlet comp class though servlet comp class normal class and allows us to create multiple objects. Based on this disccussion we can say our servlet comp is not a singleton java class.

=>ServletConfig obj is right hand object to our servlet class obj .. It is very useful object for programmers to know info about servlet comp and to gather info to servlet comp from outside.

=> ServletConfig object will be created right after our servlet class object and will be intialized/assigned to our servlet class object using init(ServletConfig cg) method.

=>Hardcoding inputs direcly in servlet comp is bad pratice ..prefer soft coding (collecting from outside)

=> SoftCoding in servlet programming

a) using request parameters (form data)

--> if the input values are non-technical and expected from endusers then prefer to get them as requiest parameters (form data)

--> These values will be stored automatically in req object and we can use req.getParameter(-) method to read the values.

eg:: name, addrss, email id , age , course and etc..

b) using servlet init parameters

->if the input values are technical input values and expected from the programmers then this concept use

eg:: jdbc driver class name, jdbc url, db user, db pwd and etc..

->these values will be placed in web.xml .. So programmer can palce technical inputs here

-> these values will be stored automatically in ServletConfig object.. the moment that object is created.

-> These are init parameter values and these values can be read using cg.getInitParameter(-) method.

Nov 10 ServletConfig obj

ServletConfig object

=>ServletConfig obj is right hand object to our servlet class obj .. It is very useful object for programmers to know info about servlet comp and to gather info to servlet comp from outside.

=> ServletConfig object will be created right after our servlet class object and will be initialized/assigned to our servlet class object using init(ServletConfig cg) method.

=>Hardcoding inputs direcly in servlet comp is bad pratice ..prefer soft coding (collecting from outside)

=> SoftCoding in servlet programming

a) using request parameters (form data)

--> If the input values are non-technical and expected from endusers then prefer to get them as requlest parameters (form data)

--> These values will be stored automatically in req object and we can use req.getParameter(-) method to read the values.

eg:: name, addrss, email id , age , course and etc..

b) using servlet init parameters

->if the input values are technical input values and expected from the programmers then this concept

eg:: jdbc driver class name, jdbc url, db user, db pwd and etc..

->these values will be placed in web.xml .. So programmer can palce technical inputs here

-> these values will be stored automatically in ServletConfig object.. the moment that object is created.

-> These are init parameter values and these values can be read using cg.getInitParameter(-) method.

In web.xml

```
<servlet>
  <servlet-name>lc</servlet-name>
  <servlet-class>com.nt.servlet.LcTestServlet</servlet-class>
  <init-param>
    <param-name>driverClass</param-name>
    <param-value>oracle.jdbc.driver.OracleDriver</param-value>
  </init-param>
  <init-param>
    <param-name>dbuser</param-name>
    <param-value>system</param-value>
  </init-param>
  <load-on-startup>10</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>lc</servlet-name>
  <url-pattern>/lcurl</url-pattern>
</servlet-mapping>
```

ServletConfig object

driverClass: oracle.jdbc.driver.OracleDriver
dbuser:system
(Init params)

LcTestServlet.java

```
public class LcTestServlet extends HttpServlet {
    static {
        System.out.println("LcTestServlet:: static block");
    }

    public LcTestServlet() {
        System.out.println("LcTestServlet:: 0-param constructor");
    }

    @Override
    public void init(ServletConfig cg) throws ServletException {
    System.out.println("LcTestServlet: init(-) method");
    String driver=cg.getInitParameter("driverClass");
    String user=cg.getInitParameter("dbuser");
    System.out.println(driver+"  "+user);
    }

    @Override
    public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {
        System.out.println("LcTestServlet.service(req,res)");

        destroy();
    //get PrintWriter
        PrintWriter pw=res.getWriter();
        res.setContentType("text/html");
        //write content to response object
        pw.println("<h1 style='color:red;text-align:center'> Date and time ::"+new java.util.Date()+"</h1>");
        //close stream
        pw.close();
    }

    public static void main(String[] args) {
        System.out.println("main(-) method");
    }

    @Override
    public void destroy() {
        System.out.println("LcTestServlet.destroy()");
    }

}
```

After our servlet class object.. the SerletContainer creates 1 ServletConfig object having servlet init param names and values collected from the web.xml file and that ServletConfig object is handed over to ServletClass object through intialization process by calling init(-) method having ServletConfig obj as the arg value. So by cg.getInitParameter(-) method we can read and display init param values.

=>When request object is created for every request .. the recived req param values(form data) will be stored into request object automatically.

=>When Servletconfig is created for every Servletclass obj .. the init param values colelcted web.xml's servlet cfgs will be stored into the ServletConfig object automatically.

Understanding two init methods of servlet api

=>servlet api is having two init() methods

a) 1st init(-) method / init with parameter method

=> signature :: public void init(ServletConfig cg)throws ServletException

=> It is servlet life cycle method for instantiation event

=> this method is orginally declared in jakarta.servlet.Servlet(I) and implemented jakarta.servlet.GenericServlet(AC).

=>jakarta.servlet.http.HttpServlet(AC) does not contain any init(-) methods .. inherits and uses init methods from jakarta.servlet.GenericServlet(AC).

=>The init(ServletConfig cg) method definitation of GenericServlet class contains logic to initialize the recived ServletConfig object with instance variable of GenericServlet class.

```
public abstract class GenericServlet implenetns Servlet(I){
  private ServletConfig config;
  public void init(ServletConfig config) throws Exception{
      this.config=config;  //initialization of ServletConfig object.
      .....
  }
  ....
}
```

b) init() /init no parameter method

=> It is not servlet life cycle method for instantiation event .. It is convience method given to Programmers to override in servlet comp having programmer's choice initialization logic.

=> signature :: public void init() throws ServletException

=>This is direct concrete method jakarta.servelt.GenericServlet (AC) having no logic i.e method definitation with no logic in body (Null Method Definitation)

=>GenericServlet (AC) class contains ServletConfig obj's initialization logic the init(-)method definitation and also calls init() method as shown below.

```
public abstract class GenericServlet implements Servlet{
  private ServletConfig config;
     //1st init method
  public void init(ServletConfig config)throws ServletException{
       this.config=config;  //initialize Servletconfig obj
        init();
  }      //2nd init method
  public void init() throws ServletException{
  }

  public  ServeltConfig getServletConfig(){
     return config;
  }
  ....
  .... //other methods
  ...
  }
```

By calling this method outside of class or in the sub class we can get access to ServletConfig object in outside the class or in sub class.

```
class  Test{                          (c)
  public void y(){
    z();  (d)
  }
  public void z(){
    ..
  }
}
```

```
class Demo extends Test{
  public void z(){                    (b?)
    ...   (e)
  }
}
```

```
Demo d=new Demo();
d.y();  (a)
```

=>if the super class method defintation is executing having the sub class as the invoking object.. so any other method invocation from aslo searches in sub class // then it invokes sub class mehod (refer above example)

Approach1 of keeping init mehdos

//GnericServlet.java

```
public abstract class GenericServlet implements Servlet{
  private ServletConfig config;
     //1st init method      (5)
  public void init(ServletConfig config)throws ServletException{
      this.config=config;  //initialize Servletconfig obj
        init();  (6)
  }      //2nd init method
  public void init() throws ServletException{
  }

  public  ServeltConfig getServletConfig(){
     return config;
  }
  ....
  .... //other methods
  ...
  }
```

By calling this method outside of class or in the sub class we can get access to ServletConfig object in outside the class or in sub class.

Q) How init() method placed in servelt comp is executing from our servlet comp though it is not life cycle method

(Ans) :: GenericServlet's init(Servletconfig) method executed as life cycle method and that is internally calling init() method.. In that process the overridden init() method of our servlet comp class executes

TestSErvlet (our servlet comp)

(1)First reqqest (2) servlet loading and instatiation (3) ServletConfig object creation (stores init params into it)

(4) Raising of instantiation event and Servletcontainer calling init(ServletConfig) method on our servlet class obj

```
public class Testservlet extends GenericeServlet/HttpServlet{
     void
  public init() {        (7)
     ....  //our initialization logic
     ...                    like creating jdbc con obj
     ....    ServletConfig cg=getServletConfig();
     ....    //use cg  here.. to read param value..
     ...
  }
```

init

Nov 11 Two Service methods and 7 doXxx(-,-) mehods

Understanding two init methods of servlet api
==

=>servlet api is having two init() methods

a) 1st init(-) method / init with parameter method

=> signature :: public void init(ServletConfig cg)throws ServletException
=> It is servlet life cycle method for instantiation event
=> this method is orginally declared in jakarta.servlet.Servlet(I) and implemented jakarta.servlet.GenericServlet(AC).
=>jakarta.servlet.http.HttpServlet(AC) does not contain any init(-) methods .. inherits and uses init methods from jakarta.servlet.GenericServlet(AC).

=>The init(ServletConfig cg) method definitation of GenericServlet class contains logic to initialize the recived ServletConfig object with instance variable of GenericServlet class.

```
public abstract class GenericServlet implenetns Servlet(I){
 private ServletConfig config;
 public void init(ServletConfig config) throws Exception{
     this.config=config; //initialization of ServletConfig object.
      .....
 }
 ....
}
```

b) init() /init no parameter method

=> It is not servlet life cycle method for instantiation event .. It is convience method given to Programmers to override in servlet comp having programmer's choice initialization logic.

=> signature :: public void init() throws ServletException

=>This is direct concrete method jakarta.servelt.GenericServlet (AC) having no logic i.e method definitation with no logic in body (Null Method Definitation)

=>GenericServlet (AC) class contains ServletConfig obj's initialization logic the init(-)method definitation and also calls init() method as shown below.

```
public abstract class GenericServlet implements Servlet{
  private ServletConfig config;
    //1st init method
  public void init(ServletConfig config)throws ServletException{
      this.config=config;  //initialize Servletconfig obj
        init();
  }   //2nd init method
  public void init() throws ServletException{
  }

  public ServeltConfig getServletConfig(){
     return config;
   }
   ....
   .... //other methods
   ...
  }
```

By calling this method outside of class or in the sub class we can get access to ServletConfig object in outside the class or in sub class.

```
class Test{ (c)
  public void y(){
    z(); (d)
  }
  public void z(){
    ..
  }
}

class Demo extends Test{
  public void z(){       (b?)
     ...   (e)
   }
}
                Demo d=new Demo();
                d.y(); (a)
```

=>if the super class method defintation is executing having the sub class as the invoking object.. so any other method invocation from aslo searches in sub class // then it invokes sub class mehod (refer above example)

Approach1 of keeping init mehdos
==============================

Q) How init() method placed in servelt comp is executing from our servlet comp though it is not life cycle method

(Ans) :: GenericServlet's Init(Servletconfig) method executed as life cycle method and that is intenrally calling init() method.. In that process the overridden init() method of our servlet comp class executes

```
//GnericServlet.java
==================
 public abstract class GenericServlet implements Servlet{
   private ServletConfig config;
     //1st init method     (5)
   public void init(ServletConfig config)throws ServletException{
       this.config=config;  //initialize Servletconfig obj
         init();  (6)
   }     //2nd init method
   public void init() throws ServletException{
   }

   public ServeltConfig getServletConfig(){
      return config;
    }
    ....
    .... //other methods
    ...
   }
```

By calling this method outside of class or in the sub class we can get access to ServletConfig object in outside the class or in sub class.

```
TestSErvlet (our servlet comp)
========================
(1)First reqqest   (2) servlet loading and instatiation
public class Testservlet extends GenericeServlet/HttpServlet{

   public void init() {     (7)
     .... //our initialization logic
                like creating jdbc con obj
     ...
     ....   ServletConfig cg=getServletConfig();
     ....   //use cg here.. to read init param value..
     ...
   }

   public void service( ServletRequest req, ServletResponse res)throws SE,IOE{
    ....
    ...
   ...    ServletConfig cg=getServletConfig();
   ....   //use cg there.. to read init param value..
  }
}
```

(3) ServletConfig object creation (stores init params into it)
(4) Raiising of instantiation event and Servletcontainer calling init(ServletConfig) method on our servlet class obj

note:: same class mehtods and super class public methods is sub class can be called with out object ref varalble.. as shown for getServletConfig()

What happens if we place both init() methods in our servlet comp?

Ans) init(ServletConfig) method executes...becoz it is life cycle method of SErvlet comp for instantiation event

if do not place any init() method in our servlet comp then what happens?

Ans) The super class method executes GenericService Init(ServletConfig)executes and this method after intializating Servletconfig object it intenrally calls init() method and init() method of Generic Servlet class executes.

What is the recomanded init() method to place in our servlet comp?

Ans) Though init(ServletConfig) is the life cycle method .. It is recomanded to place init() in our servlet comp having our choice initialization.. So for instantiation life cycle event init(ServletConfig) method will be called on our servlet class obj.. since not available the super class (Generic Servlet class) init(ServletConfig) executes , this method contains logic to intilized ServletConfig object and calls init() method, so init() method of our servlet comp class executes.

=>Since we are giving chance to execute init(ServletConfig) method of GenericeServlet class.. It will take care of initializing ServletConfig object and programmer is free from that process.. and by overriding init() method in servlet comp programmer can place his choice intialization logics.

when init(ServletConfig) method is life cycle method .. why the servlet api has given init() method?

Ans) if init() method is not there in servlet api .. then we override init(Servletcofig) method directly in our servlet comp where we are forced to initialize SErvletConfig obj and placing our choice intialization logics like creating jdbc con object .. In the process writing our choice initialzation some time we may forget to intialize servlet config and that may to problems to access ServletConfig object in other methods of sevlet comp...

To overcome this problem they given init() method .. so programmers overrides init() method having his choice initlization logic and init(ServletConfig) method GenericeServlet executes as life cycle method which internallys ini() method..

=>Since we are giving chance to execute init(ServletConfig) method of GenericeServlet class.. It will take care of initializing ServletConfig object and programmer is free from that process.. and by overriding init() method in servlet comp programmer can place his choice intialization logics.

Best pratice :: place init() method in our servlet comp having our choice initialization logics and use getServletConfig() method every where in servlet comp to access to Servletconfig obj that is intilzed by super class (GenericServlet class) init(ServletConfig)method (lfe cycle method)

Understaning 2 service(-,-) methods and 7 doXxx(-,-) methods of servlet api
===

1) 1st service(-,-) / public service(-,-) method
==

=>It is life cycle method of request processing event
=> signature ::
public void service(ServletRequest req,ServletResponse res)throws SE,IOE
=> This method is orginally declared in jakarta.servlet.Servlet(I) and maintained as abstract method in jakarta.servlet.GenericServlet(AC) and implemented in jakarta.servlet.http.HttpServlet(AC)

2) 2nd service(-,-) /protected service(-,-) method
==

=>It is not a life cycle method ..It is convience method given to prorgammers.
=>signature ::
protected void service(HttpServletRequest req, HttpServletResponse res)throws SE,IOE
=>This method is direct concrete method of jakarta.servlet.http.HttpServlet(AC) class.

3) 7 doXxx(-,-) methods (doGet(-,-)/doPost(-,-)/doDelete(-,-)/doPut(-,-)/doTrace(-,-)/doOptions(-,-)/doHead(-,-))
==

=>These are not life cycle methods of servlet comp.. These are convience methods given programmers to handle 7 modes of http requests using 7 doXxx(-,-) methods.
=> signature :: protected void doXxx(HttpServletRequest req,HttpServletResponse res)throws SE,IOE
=>These are direct concrete methods of jakarta.servlet.http.HttpServlet(AC) class.

=>HttpServlet class is having 1st service(-,-), 2nd service(-,-) and 7 doXxx(-,-) methods as concrete methods along with other methods.. 1st service(-,-) intenrally calls 2nd service(-,-) method and 2nd service(-,-) method internally calls one of the 7 doXxx(-,-) method based on the given request mode/method.

```
HttpServlet.java
==============

public class HttpServlet extends GenericServlet {

 //1st service(-,-) or public service(-,-) method
  public void service(ServletRequest req,ServletResponse res)throws SE,IOE {   (6)
    //convert ServletRequest obj to HttpServletRequest obj
    //and  ServletResponse obj to HttpServletResponse obj

      HttpServletRequest request=(HttpServletRequest)req;
      HttpServletResponse response=(HttpServletResponse)res;
    //call 2nd service(-,-) /protected service(-,-) method
          service(request,response);   (7)
    }

 //2nd service(-,-) or protected service(-,-) method
   protected void service(HttpServletRequest req,HttpServletResponse res)throw SE,IOE{   (8)
     //get request method/mode
      String method=req.getMethod();
     if(method.equalsIgnoreCase("GET"))
          doGet(req,res);  (9)
     else if(method.equalsIgnoreCase("POST"))
          doPost(req,res);
     else if(method.equalsIngoreCase("HEAD")
          doHead   (req,res);
       ....
       ....
       ....
     }

  //doXxx(-,-) methods  (7 methods)
   protected  void  doXxx(HttpServletRequest req,HttpServletResponse res)throws SE,IOE{
      ...
      ... //sends 405 error response to browser
      ...
    }
 }
```

=>Do not let to execute the super class(HttpServlet) doXxx(-,-) methods becoz they send 405 error response to browser indicating our servlet comp is not ready to process the given mode request.

```
our servlet comp
==============
(1)GET  (2) req,res objs  (3) creaing/locating servlet obj  (4) thread
(5)calls 1service(-,-) as life cycle method  -- request processing event
public class TestServet extends  HttpServlet{

   public void doPost(HttpServletRequest req,HttpServletResponse res)throws SE,IOE{   (10)
     ....
     .... //request processing logic
     ....        (11)
   }

 }
```

Different usecase.
=================

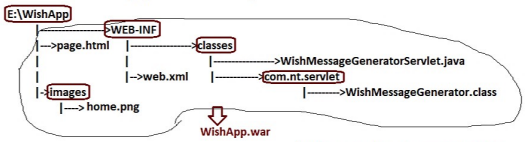
Assume the client makes a HTTP request based on GET method in the following situations

1. **If our Servlet class contains both doGet(-,-) and doPost(-,-)**
   a. public service(-,-) of HttpServlet
   b. protected service(-,-) of HttpServet
   c. doGet(-,-) of our Servlet class
2. **If our Servlet class contains only doPost(-,-)**
   a. public service(-,-) of HttpServlet
   b. protected service(-,-) of HttpServlet
   c. protected doGet(-,-) of HttpServlet(as our class does not contain doGet(-,-))
   d. 405 response back to client.
3. **If our Servlet class overrides public service(-,-) method, and contains doGet(-,-), doPost(-,-).**
   a. public service(-,-) of our Servlet class [doGet(-,-) of our Servlet class will not get invoked, because the control did not pass on to protected service(-,-) of HttpServlet].
4. **If our Servlet class overrides public service(-,-) method, and does not contain doGet(-,-)**
   a. public service() of our Servlet class
   No 405 response back to client.
5. **If our class overrides public service(-,-) method and it makes a call to super.service(-,-) method, and contains doGet(-,-), doPost(-,-)**
   a. public service(-,-) of our Servlet class
   b. public service(-,-) of HttpServlet(because of super.service(-,-))
   c. protected service(-,-) of HttpServlet
   d. doGet(-,-) of our Servlet class
6. **If our class overrides public service() method, and it makes a call to super.service(-,-) method, and contains only doPost(-,-)**
   a. public service(-,-) of our Servlet class
   b. public service(-,-) of HttpServlet(because of super.service(-,-))
   c. protected service(-,-) of HttpServlet
   d. protected doGet(-,-) of HttpServlet
   e. 405 response back to client

=>The web application will be packaged and released to the market in the form war file
war file :: web application archieve
jar file :: java archieve
war file is also jar file acting as web application..

Nov 12 War file

=>war file will be created using "jar" command having the extension .war file



E:\WishApp>jar cf WishApp.war .

Here every thing of current directory will be added to the war file (WishApp.war)

c-->create archieve (compressed file)
f-> specify archieve file name
"." -->indicates current directly..

Where exactly war file will be used in real projects?
=>For deployment of the java web application
=>For releasing web application to clietns
=> For hosting java web application on to the internet
=> To pack the web application and its web comps to a single file
and etc..

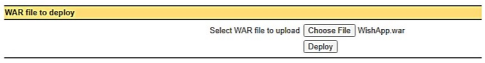
Different ways of deploying Java Web application in servers
==========

a)Console Deployment
[By using console window or Manager window of web server]
b) Hard Deployment
[By copying the web app content(either as directory or as war file) to Server Folder structure <Tomcat_home>\webapps folder]
c) Tool based Deployment
[Deployment of web application using tools like IDE, maven and gradle]

Procedure to perform console deployment of java web application in Tomcat server
==========
step1) prepare war file representing the web application.
(As shown above WishApp.war)

step2) start Tomcat server (out side)
use <Tomcat_home>\tomcat10.exe file
step3) Open Tomcat admin console or manager window
Tomcat home page (http://localhost:3535) -----> manager app ----->
submit username : admin , password: admin

step4) Upload the war file by going "Upload War file " section



The deployed war file will be moved to <Tomcat_home>\webapps folder and the war file will be extracted there

step5) Test the web application..
url ::: http://localhost:3535/WishApp/page.html

The war file name becomes the name of the web application or context path or context root

To undeploy the web application from admin console
==========
Tomcat manager screen ----> go WishApp application ---> undeploy

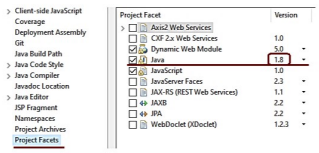
Procedure to perform hard deployment of web application in Tomcat server
==========
=> copy the web application directory(WishApp folder) or war file(WishApp.war) to <Tomcat_home>\webapps folder.

Undeployment of the web appliation if the app is deployed using hard deployment process
==========
a) undeployment from admin console (refer abovel)
(or)
b) delete directory or war file from <Tomcat_home>\webapps folder

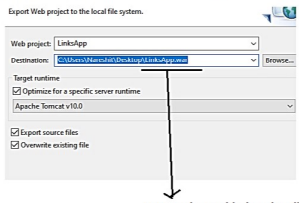
Tool based Deployment
==========
=> The deployment and execution of the java web application using eclipse tool or maven or gradle tool falls under tool based deployment.

Preparing war file using Eclipse IDE on Dynamic web project
==========
step1) Try find the outside tomcat server's jdk compitable version (mostly java8)

step2) For eclipse dynamic Project change java version 1.8
Right click on the project ---> properties ---> Project Facets ---> java --> change version to 1.8



step3) Prepare war file by using "Export" option.
Right click on Eclipse Dynamic Project --->export ---> war--->



Gives LinksApp.war file on the destkop

WE can keep this location directly to <Tomcat_home>\webapps folder

step4) Deploy and the test the webapplication any java bsaed webserer or application server.
(we can use either console deployment or hard deployment)

What is difference b/w HOT Deployment and COLD Deployment?

=> if the web application is deployed when the server is in running mode then it is called HOT Deployment ..

=> if the web application is deployed when the server is in stopped mode then it is called COLD Deployment ..

=> The "Console Deployment process of the web application" is always HOT Deployement
=> The "Hard Deployment and Tool Boased deployment" of the web application can be HOT deployment or COLD Deployment.

Hosting of Java web application
==========

Nov 13 Hosting Web application

=>The process of keeping class room Apps over the internet having the fixed domain name(website name) is called Hosting the Web application.

=>For hosting the website , we need to take the support of Domain registar company like goDaddy , jelastic , i9, indiarocks, indiabricks and etc..

=>Most of domain registars do not support java based hosting becoz they do not provide java web server /Application server where as Jelastic support java based hosting..

The domain registart offers the following services
==========

a) Selling the domain names
-->like nataraz.com , nataraz.in and etc.

b) selling the space in web server /Applicaiton server for hosting our Apps ... This web server or Application will be installed and maintained in static IP address(fixed) machine of the internet..

c) AMC (Annual Maintainance Contract)

d) Digital marketing (SEO)

=>In regular internet connection.. the ip address machine changes time to time

=>Domain registar machine always maintains fixed IP/Static IP address.

Procedure to host class Room Application on to the internet using "jelastic" domain registar
==========

step1) change the java version 8 and prepare war file representing the web application.
LinksApp.war (refer previous class)

step2) Register with jelastic by submitting emailid and get jelastic account having 14 days trail period

=>jelastic.com home page --->submit emaild --> select cloud for developer ---> select daidem india --->accept terms and conditons ---> proceed
Edge cloud or cloudjiffy

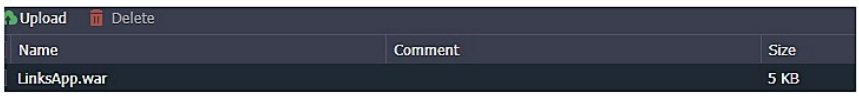
=>Login the sumibitted email id ---> check mail from jelastic (even in spam folders)---> activate now ----> choose password ---> submit mobile no ---> submit code ---> activate..

step3) Create Env.. in the jelastic annount chooise java version and Tomcat server App

go to jelastic account dashboard ---> create new env.. by choosing Tomcat 10 , jdk 8 and NITWEB as the env.. name.

step4) upload the war file...

select env.. (NITWEB) --->upload --->select war file (LinksApp.war)



step5) Host/deploy to the server..

Go to uploaded war file ----> deploy to

Deploy

step6) Test the web applicaiton..

Open browser to run the application..

http://env-0449134.user.edgecloudph.com/LinksApp/

7 Http methods

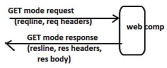
Nov 15 Seven Http methods

=>From different types of client we can give 7 modes/methods requests to web application..
and to process these we can 7 doXxx(-,-) methods in servlet programming

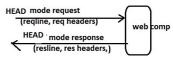
http methods	7 doXxx(-,-) methods
GET	doGet(-,-)
POST	doPost(-,-)
HEAD	doHead(-,-)
PUT	doPut(-,-)
DELETE	doDelete(-,-)
TRACE	doTrace(-,-)
OPTIONS	doOptions(-,-)

From Browser (client) we can given only 2 modes of requests (GET) and (POST) .. So when we develop website , we generally override doGet(-,-) and doPost(-,-) methods to place request processing logic.

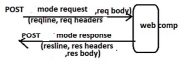
GET
====
=>Given to get Data from server by sending the request
=>can carry limited amount of data along with the request
=> This mode request related response contains all the 3 parts
(response line, response headers and response body)
=>This mode request structure does not contain request body.. becoz inputs goes to server
in the form of query string



HEAD
========
=> same as GET but the HEAD mode request related response does not contain response body
=> Head mode request is very useful to checking wheather certain web comp is available or not
i.e useful for debugging.



POST
=====
=> Given to POST/SUBMIT data to server
=>Can carry unlimited data along with the request
=>Both POST mode reuqest structure and response structure contains
BODY
note: In GET mode form data goes to server as query String .. where POST mode request
the form data goes to server as requeste body.



=>Only GCT,POST mode requests can be given from browser.. The other requests can be given from cpanel (control panel) of Domain Registars like jelastic dashboard

PUT
====
=> Given to place new web comp in the web application or
to upload web application to web server
=>Very useful in the hosting of web application.

DELETE
======
=>Given to delete the web comp of the web application and to delete
the hosted web application from the web server
=>Very useful in the Hosted webisite management

OPTIONS
=======
=> This mode request given web comp determines what are different modes of requests that the web comp allows.

If we give OPTIONS mode request to this servlet comp
We get following response
Allow;GET,HEAD,TRACE,OPTIONS

```
public class TestServlet extends HttpServlet{
  public void doGet(req,res)throws SE,IOE{
    ....
    ...
  }
}
```

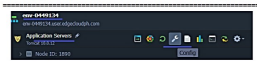
If we give OPTIONS mode request to this servlet comp
We get following response
Allow: POST,TRACE,OPTIONS

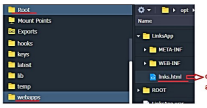
```
public class TestServlet extends HttpServlet{
  public void doPost:(req,res)throws SE,IOE{
    ....
    ...
  }
}
```

TRACE
======
=> This mode request traces all the parties/comps that involved to execute the web comp of the web application
=>Gives various details that involved in the success or failure of request processing and response generation
=>Very useful to know trace reason of web comp execution's failure..
=>TRACE mode requests are useful for debugging and testing..

=>GET,HEAD,TRACE,OPTIONS mode requests are idempotent i.e safe to repeat the requests
=>PUT,DELETE,POST, mode requests are non-idempotent i.e not safe to repeat the requests..

Managing the hosted websiste
==========
1.Modifying the source code of html file in the hosted web applicaiton of jelastic
==========



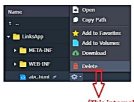


=>Test the web application...

2. Adding the new html file to hosted web application

=>develop html file outside having ur choice code (abc.html)
=> upload file LinksApp folder of deployed web application.

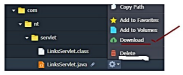
3.Deleting the exiting web comp of the web application.



4. Modifying the soruce code of servlet comp in Hosted java web applicaiton of jelastic
==========

note: Since we can not compile servlet comp from the DashBoard of jelastic .. do that work outside and replace existing .class with new .class file.

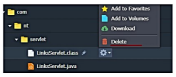
step1) Donwload servlet comp soruce file (LinksServlet.java)



step3) Modify the soruce code in the .java file of "downloads" folder
and recompile there it self using javac -d option.

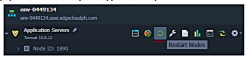
```
C:\Users\Nareshit\Downloads>javac -d . LinksServlet.java
```

step4) delete exsting .java and .class files of same server comp from jelastic
LinkeServlet.java and LinksServlet.class



step5) Upload the modified .java and .class files.
to the LinksApp\WEB-INF\classes\com\nt\servlet folder

step6) Restart the node (LinksApp of Server)



step7) Test the web application (run once)

If more chanages are there to perform in multiple web comps then prefer doing all changees at once prepare war file reflecting all the chnages and redeploy the web application by removing existing war file..

Servlet to Db s/w communication
==========

Nov 16 Servlet to DB communication

Servlet needs to talk Db s/w in the following situations
a) To save the recieved inputs from enduser to Db s/w
eg:: customer registration , profile registrtion in naukari.com , opening FB/Gmail account
b) To collect Data from DB tables and to display for endusers
eg:: sales report , displaying trending jobs , getting and displying fb posts for enduser

c) To collect data from DB tables and to use as the b.logic inputs
eg:: For Login details Verification, Getting Ticket pricess and using them in ticket booking and etc..
d) To store generated results of servlet comp to db table
eg:: After calculating bill details .. storing them in db table
eg:: after calculating HRA,DA,TA,PF and Gross Salary ,NetSalary we store them in db table

For servlet to Db s/w communication , we need to place JDBC code or Hibernat code or spring jdbc code or spring data jpa code or spring orm code or etc.. in servlet comp

For Servlet to Db s/w communication using JDBC there are 3 approaches to follow

Approach1 (Bad)
=========
=>create jdbc con in the init() method
=> use jdbc con in the service(-,-) or doXxx(-,-) methods
=> close jdbc con in the destroy() method

```
public class TestServlet extends HttpServlet{
  private Connection con;
   public void init(){
     ...
     ... //create jdbc con
   }
   public  void  service(-,-)/doXxx(-,-) throws SE,IOE{
     ....
     .... //use jdbc con
     ....
   }
   public void destroy(){
     ....
     ...   //close jdbc con
   }
}
```

=>Here we need to take JDBC con object as instance variable

Limitations:
---

a) jdbc con object is instance variable , So it is not thread safe by default..
To overcome this problem, we need to use synchronization concepts
b) All the requests comming to servlet comp uses same jdbc con object..
if any simultaenous request calls con.rollback() ..then other requests
related persistence opeprations will also be rolledback.

advantages:
---

a) Since all requests are using same jdbc con , we can say the performence of the servlet comp is good.

Approach2) (Good for small apps)
==========
~~=>create JDBC con obj in the service(-,-)/doXxx(-,-) method~~
=>use jdbc con object in the service(-,-)/doXxx(-,-) method
=>close jdbc con object in the service(-,-)/doXxx(-,-) method

=>Here jdbc con object is local variable to service(-,-)/doXxx(-,-) method.. So it thread safe object by default.

advantages
==========
=> Here jdbc con object is thread safe object
=> Since every request is using its own jdbc con object.. so any simultaenous request related con.rollback() method does not effect other requests related persistence operations.

Disadvantage

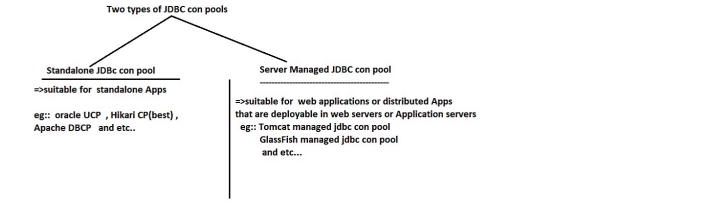
=> Using seperate jdbc con object for every request kills the performence of the Application.

Approach3 (Good for both small and large scale Apps)
=========
=> get jdbc con object from jdbc con pool being form service(-,-)/doXxx(-,-) methods
=> use jdbc con object in service(-,-)/doXxx(-,-) methods
=> return jdbc con object back to jdbc con pool from service(-,-)/doXxx(-,-) methods

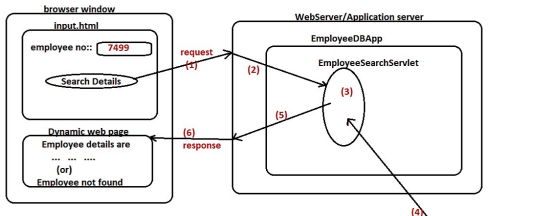
=>JDBc con pool is a factory that contains set of readily available jdbc con obejcts before actually being used.
pros:: -> Reusability of jdbc con objs
-> With minimum jdbc con objs , we can make max clients talking to Db s/w
-> creating jdbc con obj ,managing jdbc con obj and destroying jdbc con object activities will be taken care by jdbc con pool

pros
====
=> all advantages of jdbc con pool can be taken
=> Here the ref variable that points to jdbc con object of jdbc con pool is local variable . So it is thread safe
=> Since the requests comming to Servlet comp are using pooled jdbc objs we can say performence is quite good

cons
===
=>We should choose that web server or App server which supports JDBC con pool



Example App Servet To DB s/w interaction
==========



Deployement Directory Strcuture

```
EmployeeDBApp
  |---------->WEB-INF
  |-->input.html  |------------------->classes
                  |-->lib              |---->EmployeeSearchServlet.java
                  |  |->ojdbc6.jar     |---->com.nt.servlet
                  |                    |---->EmployeeSearchServlet.class
                  |->web.xml
```

lib,classes,WEB-INF.web.xml names standard/fixed names..

=>jar files in CLASSPATH :: servlet-api.jar , ojdbc6.jar

=>The stanalone App's compilation and execution takes place from command prompt , so the jar files added to CLASSPATH will be used at both compile time and runtime.

=>The servlet comp's compilation takes place from cmd promt and execution takes place from the servlet container So , if the servlet comp uses third party api like oracle jdbc driver.. then that third party api related jar files (ojdbc6.jar) must be added to CLASSPATH for compilation and also must be added WEB-INF/lib folder for servlet container's exection

=>Jar files add CLASSPATH will be used by java compiler (javac) during the compilation of servlet comp.. to recgonize the thrid party api where as the jar file added to WEB-INF/lib folder will be used by ServletContaienr during the execution of Servlet comp to recongize and use third party api ..

if the Project Eclipse created Dynamic Web application project then
we need to third party api jar to the BUILD PATH and also to the src/main/webapp/WEB-INF/lib folder.

Deployement Directory Strcuture for Eclipse IDE

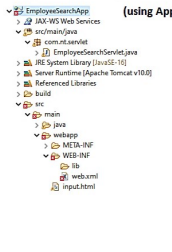
```
EmployeeDBApp
  |---------->src/main/java
  |             |--->com.nt.servlet
  |                    |--->EmployeeSearchServlet.java
  |---->src
  |       |--->main
  |              |-->webapp
  |                    |--->input.html
  |                    |---->WEB-INF
  |                           |-->lib
  |                           |  |->ojdbc6.jar
  |                           |
  |                           |->web.xml
```

jar files in build path
servlet-api.jar , ojdbc6.jar
=>Optional to place , if the target runtime env.. is Tomcat server..

Example app on Servlet to Db Communication

Nov 17 Servlet to DB communication Example App

(using Approach2)



input.html

```
<h1 style="color:red;text-align:center"> Html to Servlet interaction </h1>

<form action="empdburl" method="POST">
 <table align="center" bgcolor="cyan">
 <tr>
  <td>Enter Employee no::</td>
  <td><input type="text" name="eno"> </td>
 </tr>
 <tr>
   <td><input type="submit" value="search"> </td>
   <td><input type="reset" value="cancel"> </td>
 </tr>
 </table>
</form>
```

//EmployeeSearchServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class EmployeeSearchServlet extends HttpServlet {
	private  static final  String  GET_EMP_INFO="SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE EMPNO=?";
	@Override
	public  void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
		// set response content type
		res.setContentType("text/html");
		//get PrintWriter
		PrintWriter pw=res.getWriter();
		//read form data
		 int eno=Integer.parseInt(req.getParameter("eno"));
		 //write  jdbc code
		 Connection con=null;
		 PreparedStatement ps=null;
		 ResultSet rs=null;
		 try {
			//Load jdbc driver class
			Class.forName("oracle.jdbc.driver.OracleDriver");
			//establish the connection
			 con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
			 //create Jdbc Statement object
			 ps=con.prepareStatement(GET_EMP_INFO);
			 //set IN param values
			 ps.setInt(1, eno);
			 //execute the SQL query
			 rs=ps.executeQuery();
			 //process the ResultSet
			 if(rs.next()) {
				pw.println("<h1 style='color:red;text-align:center'>"+eno+" Employee Details are </h1>");
				pw.println("<table border='1' align='center'>");
				pw.println("<tr> <td>Empno:: </td> <td>"+rs.getInt(1)+" </td> </tr>");
				pw.println("<tr> <td>Emp Name:: </td> <td>"+rs.getString(2)+" </td> </tr>");
				pw.println("<tr> <td>Emp Desg:: </td> <td>"+rs.getString(3)+" </td> </tr>");
				pw.println("<tr> <td>Emp Salary:: </td> <td>"+rs.getFloat(4)+" </td> </tr>");
				pw.println("<tr> <td>Emp  DeptNo:: </td> <td>"+rs.getInt(5)+" </td> </tr>");
				pw.println("</table>");
			 }//if
			 else {
				pw.println("<h1 style='color:red;text-align:center'> Employee Details are not found </h1>");
			 }//else


		}//try
		catch(SQLException se) {
			se.printStackTrace();
			pw.println("<h1 style='color:red;text-align:center'> Problem in DB Interaction </h1>");
		}
		catch(Exception e) {
			e.printStackTrace();
			pw.println("<h1 style='color:red;text-align:center'> Unknown Problem  </h1>");
		}
		finally {
			//close jdbc objs
			try {
				if(rs!=null)
					rs.close();
			}//try
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(ps!=null)
					ps.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(con!=null)
					con.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
		}//finally
		//add  home  hyperlink
		pw.println("<h1 style='text-align:center'> <a href='input.html'>home </a> </h1>");

		//close stream
		pw.close();
	}//doGet(-,-)

	@Override
	public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
	doGet(req,res);
	 }//doPost(-,-)

}//class
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

<servlet>
  <servlet-name>search</servlet-name>
  <servlet-class>com.nt.servlet.EmployeeSearchServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>search</servlet-name>
  <url-pattern>/empdburl</url-pattern>
</servlet-mapping>

<welcome-file-list>
	<welcome-file>input.html</welcome-file>
</welcome-file-list>

</web-app>
```

While working with Eclipse Dynamic web project , we can move jar file added (ojdbc6.jar file) to BUILDPATH dynamically to WEB-INF/lib folder java web application by using "Deployment Assembly " operation.

procedure::
Right Click on Project ---> properties ---> deployment Assembly ---> add ---> select build path entries --->select ojdbc6.jar ---> .....

=>Instead of placing thrid party jar files in WEB-INF\lib folder of every web application it is recomanded to place in server library folder only for 1 time .. (eg:: place ojdbc.jar file in <Tomcat_home>\lib folder)

Server	Server Library Folder
a)External Tomcat	<Tomcat_home>\lib folder
b) Eclipse Tomcat	G:\Worskpaces\advjava\NTAJ115\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\lib folder (workspace folder)
c) GlassFish	<GlassFish_home>\domains\<domain-name>\lib\ext folder

"lib" will not be by default.. So create it manually

note:: keeping third party api jar files in Server Library folder is bad pratice .. becoz when move applicaiton to another server or another machine we need to move these jar files seperately and we may forget to move So better to place in WEB-INF/lib folder of web application itself.

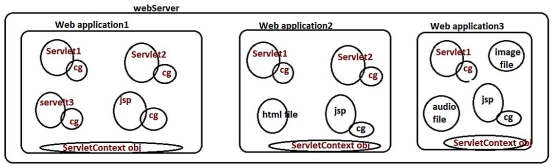
if servlet comp uses third party api , the servlet container tries recongize and use that third party api from the following places and order

a) From the classes of pkgs added to WEB-INF/classes folder of current web application if not avaialble then
b) from the jar files added to WEB-INF/lib fiolder of current web application if not avaiable then
c) from the jar files added to Server Library folder if not available then
throws ClassNotFoundException(if direct class is not avaiable) or NotClassDefFoundException(if dependent class is not avaiable)

Nov 19 ServletConfig and ServletContext objs

What is the diffence between ServletConfig object and ServletContext object?

Ans) ServletConfig object is right hand object to our servlet class obj . It is 1 per our servlet class object
ServletContext object is Global memory of the web application and it is 1 per web application.



cg (Servletconfig obj)
===
=> It is 1 per our servlet class object
=> It is called right hand object to our servlet class obj.. Using this obejct
we can pass inputs servlet comp and we can get details about servlet comp
=>ServletContainer creates ServletConfig object right after our servlet class object creation and
handovers to servlet class object by calling init(-) method (This is called intialization of servlet comp)
=>ServletConfig object means it is the object java class that implements jakarta.servlet.ServletConfig(I)
=>The servlet init parameters palced in web.xml file for servlet cfg will be stored automatically in
ServletConfig object , so we can use ServletConfig object to read init param values.

=>Using this object , we can get details about servlet comp like servlet name(logical name), its init param names
and values

=> Since ServletConfig object is specific to one Servlet comp , So its init param values are also specific
to the same Servlet comp.
=> The ServletContainer destroys the ServletConfig object just before destroying Servlet class obj

ServletContext obj (Also called as "application" object)
==========
=> It is one per each web application
=> It is called global memory of the web application ..becoz the data kept in this web application
is visible and accessible in all the web comps of the web application.
=> The Servletcontainer creates this object either during deployment of the web application (HOT deployment)
or during the server startup (COLD Deployment)
=> It is the object ServletContainer supplied java class that implements jakarta.servlet.ServletContext(I)
=> Using this object we can get multiple details like
a) The servlet,jsp comp names of the web application
b) The underlying server name and version
c) The Servlet api version supported by the server
d) Content type of the given file
e) Location /absolute path of the given file
f) InputStream pointing to the given file (To read the content)
g) For Dynamic Servlet,Filter,Listener Registration (Programatic approach)
h) To write log messages to current days log file
i) can used to read context param(global init params) names and vlaues
and etc..
=>ServletContainer destroys the ServletContext object automatically when the web application is stopped or reloaded or
undeployed

Q) In a web server 10 web applications are deployed .. In that 6 web applications are in running mode
and other 4 web applications are in stopped mode. Can we tell me total how servlet context objs are
currently available in that web server?
a) 10 b) 1 c) 6 d) 4

Ans : (c) 10 -4 : 6

Q) In a deployed java web application , 10 servlet comps are placed , In that 3 servlet comps are already requested
and other 3 servlet comps are enabled with <load-on-statup> , Can u tell me total how many ServletConfig objects
are currently avaiable?
a) 10 b) 6 c) 3 (d) 1

Ans) 3 + 3 : 6 (b)

Q) In a web application 10 servlet comps are placed and not servlet comp is requested.Can u tell me how many s
Servletconfig objects are created?
Ans) 0 (Zero object)

Q) In web server 10 java web applications are deployeed and no servlet comps are requested can u tell me how
many ServletContext objects are avaiable?

Ans) 10 Servletcontext objects

=>we do not create the following objects becoz they will created by servlet container at different phases of
execution
a) our servlet class obj
b) SerlvetConfig obj
c) ServletContext obj
d) request object
e) response obj
and etc..

=>To access our servlet comp class obj being from servlet comp use "this" operator
=>To access req,res objs use the parameters of service(-,-) , doXxx(-,-) methods
Different ways of accessing ServletConfig object
==========

Approach1) (Bad) By initializing ServletConfig object manually in the init(-) method our servlet comp

```
public class TestServlet extends HttpServlet{
  private ServletConfig cg;
    public void init(ServletConfig cg){
        this.cg=cg; //ServletConfig object initialization
       .... ..... .... //our intialization logic
    }
    public void doXxx(-,-)throws SE,IOE{
      ....
      ....
       //use cg here
    }
}
```

note:: In the process writing our initialization like creating jdbc con object.. we may forget to initialize servletConfig obejct.. So it bad pratice.

Approach2) (Best) accesssing and using ServletConfig object that is initialzed by init(ServletConfig) method of GenericServlet class
==========

```
public class TestServlet extends HttpServlet{

  public void init(){
      //get Access to ServletConfig obj
      ServletConfig cg=getServletConfig ();
     ....
   }

public void doXxx(-,-)throws SE,IOE{
       //get Access to ServletConfig obj
       ServletConfig cg=getServletConfig ();

      ....
      ...
      ..
      }
}
```

Different ways of accessing ServletContext obj in our servlet comps
==========

Approach1) By accessing ServletConfig object

```
//access ServletConfig obj
ServletConfig cg=getServletConfig();
//access ServletContext obj
Servletcontext sc=cg.getServletContext();
```

Approach2) (Good) Using convience method given by GenericServlet class

```
ServletContext sc=getServletContext();
```

→ Direct public mehtod of GenericServlet class which internally Approach code..

[Same class methods and super class public methods in sub class can be called with out object]

In GenericServlet.java

```
@Override
  public ServletContext getServletContext() {
    return this.getServletConfig().getServletContext();
  }
```

Approach3: Using request object (froms servlet 3.0)

```
ServletContext sc= req.getServletContext();
```

Hard coding inputs in servlet comp is bad pratice
so prefer softcoding (collect from out side of the servlet comp)

Softcoding can be in 3 ways
==========

a) request parameters (form data)
(if input values are non-technical values and expected from enduser like name ,age ,address and etc.)
=>These values are stored in request object ,So we use req.getParameter(-) method to read these values
b) servlet init parameters
(if input values are technical values , specific to one servelt comp and expected from Programmers like driver class name, url , db user,db pwd and etc..)

Nov 20 ServletConfig and ServletContext objs

Hard coding inputs in servlet comp is bad pratice
so prefer softcoding (collect from out side of the servlet comp)

Softcoding can be in 3 ways
==========================

a) request parameters (form data)
(if input values are non-technical values and expected from enduser like
name ,age ,address and etc.)
=>These values are stored in request object ,So we use req.getParameter(-) method
to read these values

b) servlet init parameters
(If input values are technical values , specific to one servelt comp and expected from Programmers
like driver class name, url , db user,db pwd and etc..)
=>These values are stored in ServletConfig object , so we use cg.getInitParameter(-)
method to read them eg: jdbc properties in one servlet comp

c) global init parameters /context parameters
(If input values are technical values , required in multiple servlet comps and expected from Programmers
like driver class name, url , db user,db pwd and etc..)

=>These values are stored in ServletContext obj , so we can use sc.getInitParameter(-)
method to read them eg: jdbc properties required in mulitple servlet comps

note1: form data stores in req object , So its scope is request scope i.e specific to each request
note2: servlet init params store in ServletConfig object , So they are specific to one servlet comp
note2: servlet context params store in ServletContext object , So they can be used in multiple servlet comps of a web application.

=>jdbc driver class name , url, db username, password details are called jdbc properites

Procedure to Servletconfig obj based init parameters to make jdbc propeties required in the servlet comp as flexible values to change?

step1 .) add jdbc properties as the init param names and values in web.xml file

In web.xml
=========

```
<servlet>
        <servlet-name>search</servlet-name>
        <servlet-class>com.nt.servlet.EmployeeSearchServlet</servlet-class>
        <init-param>
          <param-name>driver</param-name>
          <param-value>oracle.jdbc.driver.OracleDriver</param-value>
        </init-param>
        <init-param>
           <param-name>jdbcurl</param-name>
           <param-value>jdbc:oracle:thin:@localhost:1521:xe</param-value>
        </init-param>
        <init-param>
          <param-name>dbuser</param-name>
          <param-value>system</param-value>
        </init-param>
        <init-param>
          <param-name>dbpwd</param-name>
          <param-value>manager</param-value>
        </init-param>

      </servlet>
      <servlet-mapping>
        <servlet-name>search</servlet-name>
        <url-pattern>/empdburl</url-pattern>
      </servlet-mapping>
```

step2) read init param values of Servlet config object in the servlet comp as shown below

```
public class EmployeeSearchServlet extends HttpServlet {
        private static final String GET_EMP_INFO="SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE EMPNO=?";
        @Override
        public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
                // set response content type
                res.setContentType("text/html");
                //get PrintWriter
                PrintWriter pw=res.getWriter();
                //read form data
                 int eno=Integer.parseInt(req.getParameter("eno"));
                 //write jdbc code
                 Connection con=null;
                 PreparedStatement ps=null;
                 ResultSet rs=null;
                 try {
                        // get ServletConfig object
                        ServletConfig cg=getServletConfig();
                        String driver=cg.getInitParameter("driver");
                        String url=cg.getInitParameter("jdbcurl");
                        String dbuser=cg.getInitParameter("dbuser");
                        String dbpwd=cg.getInitParameter("dbpwd");

                        //Load jdbc driver class
                        Class.forName(driver);
                        //establish the connection
                         con=DriverManager.getConnection(url, dbuser,dbpwd);
                         //create Jdbc Statement object
                         ps=con.prepareStatement(GET_EMP_INFO);
                         //set IN param values
                         ps.setInt(1, eno);
                         //execute the SQL query
                         rs=ps.executeQuery();
                         //process the ResultSet
                         if(rs.next()) {
                                pw.println("<h1 style='color:red;text-align:center'>"+eno+" Employee Details are </h1>");
                                pw.println("<table border='1' align='center'>");
                                pw.println("<tr> <td>Empno::  </td> <td>"+rs.getInt(1)+" </td> </tr>");
                                pw.println("<tr> <td>Emp Name::  </td> <td>"+rs.getString(2)+" </td> </tr>");
                                pw.println("<tr> <td>Emp Desg::  </td> <td>"+rs.getString(3)+" </td> </tr>");
                                pw.println("<tr> <td>Emp Salary::  </td> <td>"+rs.getFloat(4)+" </td> </tr>");
                                pw.println("<tr> <td>Emp  DeptNo::  </td> <td>"+rs.getInt(5)+" </td> </tr>");
                                pw.println("</table>");
                         }//if
                         else {
                               pw.println("<h1 style='color:red;text-align:center'> Employee Details are not found </h1>");
                         }//else

                }//try
                catch(SQLException se) {
                        se.printStackTrace();
                        pw.println("<h1 style='color:red;text-align:center'> Problem in DB Interaction </h1>");
                }
                catch(Exception e) {
                        e.printStackTrace();
                        pw.println("<h1 style='color:red;text-align:center'> Unknown Problem  </h1>");
                }
                finally {
                        //close jdbc objs
                        try {
                                if(rs!=null)
                                        rs.close();
                        }//try
                        catch(SQLException se) {
                                se.printStackTrace();
                        }
                        try {
                                if(ps!=null)
                                        ps.close();
                        }
                        catch(SQLException se) {
                                se.printStackTrace();
                        }
                        try {
                                if(con!=null)
                                        con.close();
                        }
                        catch(SQLException se) {
                                se.printStackTrace();
                        }
                }//finally
                //add  home  hyperlink
                pw.println("<h1 style='text-align:center'> <a href='input.html'>home </a> </h1>");

                //close stream
                pw.close();
        }//doGet(-,-)

        @Override
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       doGet(req,res);
        }//doPost(-,-)

}//class
```

note:: . if jdbc properties are chanaged then we need not to modify servlet source code.. we just need to modify the web.xml file content (Gives flexibility of modification)

Can we choose properites file to place jdbc properties required in servlet comp?

Ans) Yes , But not recomanded becoz web.xml file is part of web application's development eacho system
So there is no need of taking seperate properties file from jdbc proprites and place jdbc properites
directly in web.xml file as servlet init param or context params.

=>ServletConfig is the obejct of underlying Servletcontainer supplied java class that implements
jakarta.servlet.ServletConfing(I)..
In Tomcat server that class name is :: org.apache.catalina.core.StandardWrapperFacade

```
            //get Servletconfig object
                      ServletConfig cg=getServletConfig();
                      pw.println("<b> db user init param value ::"+cg.getInitParameter("dbuser")+"</b>");
```

=>Using ServletConfig object we can not only read servlet init param values .. we can also use that
object to get Servlet logical name.
(servlet class obj name that is created internally)

```
          ServletConfig cg=getServletConfig();
                pw.println("<b> Servlet logical name  ::"+cg.getServletName()+"</b>");
```

gives current servlet class obj name
nothing logical name given in <servlet-name> tag.

=>if multiple servlet comps of the web application are looking same technical details like same jdbc proeprties
then instead of writing them as init params for every servlet comp.. we can write only for 1 time as
context params in web.xml file..
=>Context params also called as global init params will be stored in SErvletcontext obj (one per web application)
So they can be read from all servlet,jsp comps of web application

Example code
============
step1) place Context params in web.xml file

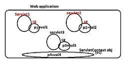
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <context-param>
        <param-name>driver</param-name>
        <param-value>oracle.jdbc.driver.OracleDriver</param-value>
      </context-param>
      <context-param>
        <param-name>jdbcurl</param-name>
        <param-value>jdbc:oracle:thin:@localhost:1521:xe</param-value>
      </context-param>
      <context-param>
        <param-name>dbuser</param-name>
        <param-value>system</param-value>
      </context-param>
      <context-param>
        <param-name>dbpwd</param-name>
        <param-value>manager</param-value>
      </context-param>

     <servlet>
      <servlet-name>search</servlet-name>
      <servlet-class>com.nt.servlet.EmployeeSearchServlet</servlet-class>
     </servlet>
     <servlet-mapping>
      <servlet-name>search</servlet-name>
      <url-pattern>/empdburl</url-pattern>
     </servlet-mapping>

     <servlet>
      <servlet-name>test</servlet-name>
      <servlet-class>com.nt.servlet.TestServlet</servlet-class>
     </servlet>
     <servlet-mapping>
      <servlet-name>test</servlet-name>
      <url-pattern>/testurl</url-pattern>
     </servlet-mapping>

     <welcome-file-list>
          <welcome-file>input.html</welcome-file>
     </welcome-file-list>

   </web-app>
```

step2) use Context param values in Servlet comps using ServletContext object

```
public class EmployeeSearchServlet extends HttpServlet {
        private static final String GET_EMP_INFO="SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE EMPNO=?";
        @Override
        public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
                // set response content type
                res.setContentType("text/html");
                //get PrintWriter
                PrintWriter pw=res.getWriter();
                //read form data
                 int eno=Integer.parseInt(req.getParameter("eno"));
                 //write jdbc code
                 Connection con=null;
                 PreparedStatement ps=null;
                 ResultSet rs=null;
                 try {
                        // get ServletContext object
                        ServletContext  sc=getServletContext();
                        String driver=sc.getInitParameter("driver");
                        String url=sc.getInitParameter("jdbcurl");
                        String dbuser=sc.getInitParameter("dbuser");
                        String dbpwd=sc.getInitParameter("dbpwd");
                         System.out.println("ServletContext  object class name::"+sc.getClass());

                        //Load jdbc driver class
                        Class.forName(driver);
                        //establish the connection
                         con=DriverManager.getConnection(url, dbuser,dbpwd);
                         //create Jdbc Statement object
                         ps=con.prepareStatement(GET_EMP_INFO);
                         //set IN param values
                         ps.setInt(1, eno);
                         //execute the SQL query
                         rs=ps.executeQuery();
                         //process the ResultSet
                         if(rs.next()) {
                                pw.println("<h1 style='color:red;text-align:center'>"+eno+" Employee Details are </h1>");
                                pw.println("<table border='1' align='center'>");
                                pw.println("<tr> <td>Empno::  </td> <td>"+rs.getInt(1)+" </td> </tr>");
                                pw.println("<tr> <td>Emp Name::  </td> <td>"+rs.getString(2)+" </td> </tr>");
                                pw.println("<tr> <td>Emp Desg::  </td> <td>"+rs.getString(3)+" </td> </tr>");
                                pw.println("<tr> <td>Emp Salary::  </td> <td>"+rs.getFloat(4)+" </td> </tr>");
                                pw.println("<tr> <td>Emp  DeptNo::  </td> <td>"+rs.getInt(5)+" </td> </tr>");
                                pw.println("</table>");
                         }//if
                         else {
                               pw.println("<h1 style='color:red;text-align:center'> Employee Details are not found </h1>");
                         }//else

                }//try
                catch(SQLException se) {
                        se.printStackTrace();
                        pw.println("<h1 style='color:red;text-align:center'> Problem in DB Interaction </h1>");
                }
                catch(Exception e) {
                        e.printStackTrace();
                        pw.println("<h1 style='color:red;text-align:center'> Unknown Problem  </h1>");
                }
                finally {
                        //close jdbc objs
                        try {
                                if(rs!=null)
                                        rs.close();
                        }//try
                        catch(SQLException se) {
                                se.printStackTrace();
                        }
                        try {
                                if(ps!=null)
                                        ps.close();
                        }
                        catch(SQLException se) {
                                se.printStackTrace();
                        }
                        try {
                                if(con!=null)
                                        con.close();
                        }
                        catch(SQLException se) {
                                se.printStackTrace();
                        }
                }//finally
                //add  home  hyperlink
                pw.println("<h1 style='text-align:center'> <a href='input.html'>home </a> </h1>");

                //close stream
                pw.close();
        }//doGet(-,-)

        @Override
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       doGet(req,res);
        }//doPost(-,-)

}//class
```

```
public class TestServlet extends HttpServlet {

        @Override
        public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
                 //get PrintWriter
                PrintWriter pw=res.getWriter();
                //set response content type
                res.setContentType("text/html");
                //get Servletconfig object
                ServletConfig cg=getServletConfig();
                pw.println("<br><b> Servlet logical name  ::"+cg.getServletName()+"</b>");
                pw.println("<br><b> db user init param value ::"+cg.getInitParameter("dbuser")+"</b>");
                //get ServletContext  object
                ServletContext sc=getServletContext();
                pw.println("<br><b> db user context param value ::"+sc.getInitParameter("dbuser")+"</b>");

                //close stream
                pw.close();
        }//doGet(-,-)

        @Override
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
                doGet(req,res);
        }//doPost(-,-)

}//class
```

Web application



=>p1 init param can be used only in servet1 comp
=>p2 init param can be used only in servet2 comp
=>p3 init param can be used only in servet3 comp
=>p4 context param can be used servet1,servelt2,servlet3 comps

ServletContext obj is the object of underlying servletContainer supplied java class implementing jakarta.sevlet.ServletContext(I)
In Tomcat server that clss name is :: org.apeche.catalina.core.ApplicationContextFacade

```
ServletContext  sc=getServletContext();
 System.out.println("ServletContext  object class name::"+sc.getClass());
```

Can we give same name for init param of a servlet and context param of a web application?
Ans) Possible becoz init param allocates memory in ServletConfig object and context param
allocates memory in ServletContext obj

What happens ~~if init~~ multiple params of servlet are having same name. with different values?

Ans) The first init param value from that list will taken as final value

```
<servlet>
      <servlet-name>test</servlet-name>
      <servlet-class>com.nt.servlet.TestServlet</servlet-class>
        <init-param>
       <param-name>dbuser</param-name>
       <param-value>system2</param-value>
      </init-param>
       <init-param>
       <param-name>dbuser</param-name>
       <param-value>system1</param-value>
      </init-param>

     </servlet>
```

first init param value will be taken

What happens if multiple context params having same name with different values?

Ans) the Last value in that list will be taken as the final value

```
<context-param>
      <param-name>dbuser</param-name>
      <param-value>system11</param-value>
    </context-param>
    <context-param>
      <param-name>dbuser</param-name>
      <param-value>system</param-value>
    </context-param>
```

The last value will be taken

ServletWrappers

Nov 21 ServletWrappers

=>A Wrapper is layer around existing item .. So we can use existing item facilities and wrapper facilities through wrapper

wrapper

=>Wrapper data types are object versions of simple data types

Integer wrapper class obj

=>ServletWrappers are given to develop Custom request obj and custom response obj as wrappers on the SErvletcontainer created request , response objs.

=> We can use custom request object and custom response object to cutomize the funcationalaties of certain methods..and using existing logics for other methods.

Need of customer request object

email Id nit@gmail.com @gmail.com

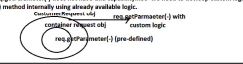
=>if write @gmail.com in the mail id it should take it as it is

if we do not write it should @gmail.com automatically

email Id nit @gmail.com

it takes email id by appending @gmail.com (overall it is nit@gmail.com)

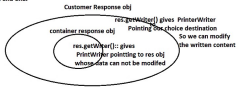
The ready made req.getParameter(-) does not have thid capability...So we need to devleop custom logic for req.getParameter(-) method internally using already available logic.



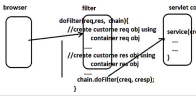
Need of CustomResponse object

=> if we want to add signature(additional contatnt) to every response .. we need to write our choice logics for res.getWriter() method where we create our choice stream object writing out choice destionation , So that we can take that written content to modify it or to add sinature contenet at end end.

Customer Response obj



=>After creating these Custom request and Custom response objects to by using container created request ,response objs To pass to servlet comp we need to take support of Filter by calling chain.doFilter(-,-) with cuustm req , custom res objs



Developing Custom Request and CustomResponse classes becomes easy if we maken them extending Request ,response Wrapper classes becoz they come with container created req,res objs ..So what ever methods we not redefining they will executed directly on container created req,res object.

problem1:

```
public class CustomReequest Implemetns HttpSrvletRequest{
  ....
  ... //need to implements all the methods (burden to the programmers)
  ...
  ..
}
public class CustomResponse implemetns HttpServletResponse{
  ....
  ... //need to implements all the methods (burden to the programmers)
  ...
  ..
}
```

Different ReadymadeWrapper classes

ServletRequestWrapper	Contains definitiation for all methods internally calling same on container created req , res objs.
ServletResponseWrapper	
HttpServletRequestWrapper	
HttpServletResponseWrapper	

Solution1:

```
public class CustomRequest extends HttpServletRequestWrapper{

  ....
  ... //override only those in which u r intrested in (bo burden on the programmers)
  ..
}

public class CustomResponse extends HttpServletResponseWrapper{

  ....
  ... //override only those in which u r intrested in (bo burden on the programmers)
  ..
}
```

Example App

login.html

```
<h1 style="color:red;text-align:center"> Login Page</h1>

<form action="loginurl" method="POST">
  <table align="center" bgcolor="cyan">
    <tr>
      <td> email Id:: </td>
      <td> <input type="text" name="email">@gmail.com</td>
    </tr>
    <tr>
      <td> password :: </td>
      <td> <input type="text" name="pwd"></td>
    </tr>
    <tr>
      <td> <input type="submit" value="Login"> </td>
      <td> <input type="reset" value="cancel"></td>
    </tr>

  </table>
</form>
```

LoginServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/loginurl")
public class LoginServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

        System.out.println("LoginServlet.doGet(-,-)");
        System.out.println("req obj type::"+req.getClass()+"  "+res.getClass());
      //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set content type
        res.setContentType("text/html");
        //read form data
        String mail=req.getParameter("email");
        String pwd=req.getParameter("pwd");
        System.out.println(mail +"     "+pwd);
        //write b.logic
        if(mail.equalsIgnoreCase("raja@gmail.com") && pwd.equalsIgnoreCase("ram")) {
            pw.println("<h1 style='color:green;text-align:center'> Valid credentials </h1>");
        }
        else {
            pw.println("<h1 style='color:red;text-align:center'> inValid credentials </h1>");
        }
        //add home hyperlink
        pw.println("<br><a href='login.html'>home </a>");
        //close stream
        pw.close();

    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        System.out.println("LoginServlet.doPost(-,-)");
      doGet(req, res);
    }

}
```

CustomerRequest.java

```
package com.nt.wrapper;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class CustomRequest extends HttpServletRequestWrapper {
  private HttpServletRequest  request;
      public CustomRequest(HttpServletRequest request) {
            super(request);
            System.out.println("CustomRequest.1-param constructor");
            this.request=request;
      }

      @Override
            public String getParameter(String name) {
            System.out.println("CustomRequest.getParameter(-)");
              String paramVal=request.getParameter(name);
              if(name.equals("email")) {
                    if(!paramVal.endsWith("@gmail.com"))
                          paramVal=paramVal+"@gmail.com";
              }
              return paramVal;
            }

}
```

CustomResponse.java

```
package com.nt.wrapper;

import java.io.CharArrayWriter;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpServletResponseWrapper;

public class CustomResponse extends HttpServletResponseWrapper {
  private HttpServletResponse response;
  private CharArrayWriter  writer;
  public CustomResponse(HttpServletResponse response) {
            super(response);
            System.out.println("CustomResponse:: 1- param  constructor");
            this.response=response;
            writer=new CharArrayWriter(); // internally creates buffer and takes it as destination

      }

    @Override
      public PrintWriter getWriter() throws IOException {
        System.out.println("CustomResponse.getWriter()");
          PrintWriter pw=new PrintWriter(writer);  // Here indirectly  PrintWriter stream is taking buffer as destination
          return pw;
      }

    @Override
      public String toString() {
          System.out.println("CustomResponse.toString()");
            return writer.toString();  //content collected from buffer is returned custom response obj content
      }


}
```

//LoginFilter.java

```
package com.nt.filter;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.FilterChain;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.nt.wrapper.CustomRequest;
import com.nt.wrapper.CustomResponse;

@WebFilter("/loginurl")
public class LoginFilter extends HttpFilter {

    @Override
    public void doFilter(HttpServletRequest req, HttpServletResponse res, FilterChain chain)
                throws IOException, ServletException {
          //create CustomRequest object
          CustomRequest  creq=new CustomRequest(req);
          //create CustomResponse obj
          CustomResponse  cres=new CustomResponse(res);

          chain.doFilter(creq, cres);  //calls next filter or destination comp
          // collect content from custom response obj
          String output=cres.toString();
          //modify the content
          output=output+"<br> <b>Naresh IT</b>";
          //get PrinterWriter  pointing to container created response obj
          PrintWriter pw=res.getWriter();
          pw.println(output); //writes container created response obj --> container --> server --> browser window

          //close stream
          pw.close();

    }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
id="WebApp_ID" version="4.0">
  <display-name>WrappersApp</display-name>
  <welcome-file-list>
    <welcome-file>login.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Using ServletContext object we can get more details about current web application and its web comps

Nov 22 ServletConfig and ServletContext objs

```
//get ServletContext object
ServletContext sc=getServletContext();
pw.println("<br> <b> server name ::"+sc.getServerInfo()+"</b>");
                || server name ::Apache Tomcat/10.0.11
pw.println("<br> <b> servlet api version ::"+sc.getMaiorVersion()+"."+sc.getMinorVersion()+"</b>");
                                || servlet api version ::5.0
pw.println("<br> <b> context path ::"+sc.getContextPath()+"</b>");
            || context path ::/EmployeeSearchApp-contextparams
pw.println("<br><b> MIME type of input.html ::"+sc.getMimeType("input.html")+"</b>");
        || MIME type of input.html ::text/html
pw.println("<br><b> path of input.html ::"+sc.getRealPath("input.html")+"</b>");
    || path of input.html ::G:\Worskpaces\advjava\NTAJ115\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\EmployeeSearchApp-
       contextparams\input.html
pw.println("<br><b> path of web root folder ::"+sc.getRealPath("/")+"</b>");
    || path of web root folder ::G:\Worskpaces\advjava\NTAJ115\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\EmployeeSearchApp-
       contextparams\
```

=> when constructor is there to place our intialziation logic in servlet comp .. why do we prefer placing the same logic in init() method of the servlet comp?

Ans) The SErvletCofig is visible and accessible in init() method of servlet comp and not visible and accessiable in the consturtor becoz Servletconfig object created after the execution of constructor and before the execution of init() .. with out servletConfig or request object we can not access SevletContext object in our servlet comp .. both ServletCofig object and request object are not visiable and accessible in the constructor i.e we can not also access ServletContext obj from the constructor

=> The intialization logic kept constructor can not work with servlet init param values and context param values becoz SeervletConfig and servletContext objs are not accessible in the constructor where as the intialization logic kept in init() method can work with servlet init param values and context param values becoz both ServletConfig and servletContext objs are accessible in init() method of servlet comp

Example code

```
public class TestServlet extends HttpServlet {

    public TestServlet() {
      ServletConfig cg=getServletConfig();
        System.out.println("Init parma vlaue:"+cg.getInitParameter("dbuser"));
        ServletContext sc=getServletContext();
        System.out.println("contextt param vlaue:"+sc.getInitParameter("dbuser"));
    }

    @Override
    public void init() throws ServletException {
        ServletConfig cg=getServletConfig();
        System.out.println("Init parma vlaue:"+cg.getInitParameter("dbuser"));
        ServletContext sc=getServletContext();
        System.out.println("contextt param vlaue:"+sc.getInitParameter("dbuser"));
    }
```

this code gives NullPointException becoz there is no visibility for SErvletConfig object in the constructor

this code executes successfully..

Can write message to browser from init method ,costructor of servlet comp?

Ans) Not possible becoz to create stream called PrintWriter we need to have res object and the res obj is local to service(-,-)/doxxx(-,-) methods and not visiable to init() method and constructor.

=> PrintWriter can be used only in service(-,-)/doXxx(-,-) methods of servlet comp becoz res object is available only in the service(-,-) or doXxx(-,-) methods.

Pointing to res object we can get two types of stream using the "res" obj itself

1)PrintWriter
2) ServletOutputStream

(1)PrintWriter

=>Use PrintWriter pw=res.getWriter();
=>PrintWriter is a character stream .. So it is good only to write text content to browser through res object.
=> PrintWriter pw=res.getWriter();
pw.println(" hello ");

note:: This stream can not be to used if servlet /jsp comp wants to read and send binary fle content (images,auido ,vedio and etc ..files) to browser as reponse becoz PrintWriter is charcter stream and can deal with only character data,

(2) ServletOuputStream

=> use ServletOutputStream sos=res.getOuptuStream();
=>ServletOutputStream is byte stream.. So it can be used to write both text data and bindary data to browser through response object.

```
ServletOutputStream sos=res.getOutputStream();
  sos.println("<b> hello </b>");
  sos.println(" binary data read from img file....");
```

note:: This stream can t be to used if servlet /jsp comp wants to read and send binary fle content (images,auido ,vedio a to browser as reponse becoz this stream is byte r stream and can deal with both character data and text data.

note:: Pointing res object we can crate only one stream at a time i.e if PrintWriter stream is created the Sevletoutputstream creation process throws error and vice-versa.

In Sevlet comp servlet's service(-,-) method

```
@Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            //get PrintWriter
            ServletOutputStream sos=res.getOutputStream();
            //set response content type
            res.setContentType("text/html");
            //get Servletconfig object
            ServletConfig cg=getServletConfig();
            sos.println("<br><b> Servlet logical name ::"+cg.getServletName()+"</b>");
            sos.println("<br><b> db user init param value ::"+cg.getInitParameter("dbuser")+"</b>");
            //get ServletContext object
            ServletContext sc=getServletContext();
            sos.println("<br><b> db user context param value ::"+sc.getInitParameter("dbuser")+"</b>");
        sos.println("<br> <b> server name ::"+sc.getServerInfo()+"</b>");
        sos.println("<br> <b> servlet api version ::"+sc.getMajorVersion()+"."+sc.getMinorVersion()+"</b>");
        sos.println("<br> <b> context path ::"+sc.getContextPath()+"</b>");
        sos.println("<br><b> MIME type of input.html ::"+sc.getMimeType("input.html")+"</b>");
        sos.println("<br><b> path of input.html ::"+sc.getRealPath("input.html")+"</b>");
        sos.println("<br><b> path of web root folder ::"+sc.getRealPath("/")+"</b>");

            //close stream
            sos.close();
        }//doGet(-,-)
```

Can we use both SErvletOutputstream and Printer steam objs in one Sevlet component?

Ans) No, Not possible the moment the other model is called exeption will come indicating already othe object stream is created pointing to response object

(java.lang.IllegalStateException: getOutputStream() has already been called for this response)

Nov 23 GlassFish Server

GlassFish server
==========

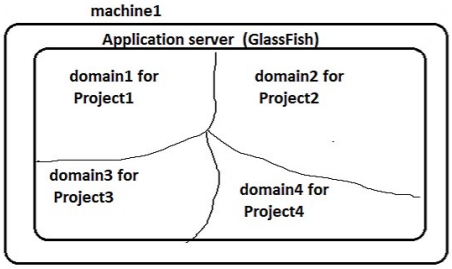
Application server = web server++

type:: Application server
version :: 6 .x (compitable with jdk 10+)
vendor :: sun Ms /oracle corp /Eclipse
Default port numbers :: http operations :: 8080 (changable)
admin console :: 4848 (changable)
Open source (free)
Allows to create domains .. each domain acts as instance/copy of Application server

Default domain name :: domain1
default domain username :: admin
default domain password : adminadmin
To download GlassFish s/w :: dowload as zip file from www.glassfish.org website
To install GlassFish s/w :: extract the zip file

In GlassFish server two important port numbers will be there
a) admin port :: To launch admin console screen pages for deployment and other activities
b) http port :: To send requst and to response for deployment web applications

In one physical installation of GlassFish server s/w we can create any no.of logical domains and it will be done on 1 per Project basis.



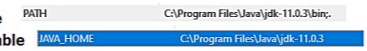
=>Every domain will be treated as an instance/copy of Application server GlassFihs

GlassFish s/w (class)
|---> domain1 (obj1)
|---->domain2 (obj2)
|---->domain3 (obj3)

=> if 1 company is using GlassFish server for 10 projects.. then GlassFish s/w will not be installed for 10 time .. that will be installed only for 1 time in a common computer but 10 domain will be created for 10 projects on 1 per project basis as shown in the diagram

Procedure to create user-defined domain in GlassFish server s/w
==========

step1) make sure that GlassFish server s/w is installed in our computer and also activate jdk10+ (jdk 11 is stable)
donwload glassfish-6.2.zip file and extract it
add <jdk10+_hime>\bin directory to PATH env.. variable
add <jdk10+_hime> directory to JAVA_HOME env.. variable



step2) create user-defined domain (make sure java 10+ is kept ready from command prompt)

```
<Glassfish_home> \bin>asadmin create-domain --adminport=4545 --user=testuser GFNTAJ115Domain
Enter the admin password [Enter to accept default of no password]>
Enter the admin password again>
Using port 4545 for Admin.
Default port 8080 for HTTP Instance is in use. Using 61898
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Using default port 9009 for JAVA_DEBUGGER.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=DESKTOP-IUDAAVL,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=DESKTOP-IUDAAVL-instance,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Domain GFNTAJ115Domain created.
Domain GFNTAJ115Domain admin port is 4545.
Domain GFNTAJ115Domain admin user is "testuser".
Command create-domain executed successfully.
```

step3) change port number of the above created domains server

Go to <GlassFish_home>\domain\NTAJ115Domain\domain.xml file and modify first <network-listener> tag's port attribute value.

New Number

```
<network-listener port="5656" protocol="http-listener-1" transport="tcp" name="http-listener-1"
thread-pool="http-thread-pool"></network-listener>
```

Procedure to perform console deployment in GlassFish domainserver
==========

step1)prepare war file representing the web application
(Take java version 10+ while creating that war file)
LinksApp.war file (change java version from Project facets of the Project properties)

step2) Start the above GlassFish domain server

<GlassFish_home> \bin>asadmin start-domain GFNTAJ115Domain

step3) open admin console of GlassFish server

http://localhost:4545 ---> submit username:testuser
passwod: testuser ----> submit

step5) Deploy the war file by uploading the war file

Admin console screen ----> applications --- > deploy --->select war file (LinksApp.war) ---> ok

step6) Test the web application

http://desktop-iudaavl:5656/LinksApp/
(or)
http://localhost:5656/LinksApp

To stop GlassFish domain server

```
E:\GlassFish6.2\glassfish6\glassfish\bin>asadmin stop-domain GFNTAJ115Domain1
Waiting for the domain to stop .
Command stop-domain executed successfully.
```

Nov 24 27 GlassFish Server -wildfly server

Procedure to undeploy the java web application from admin console

Go to admin console screen —> applications —>select app (LinksApp) —> undeploy

Procedure to peform hard deployment of java web application GlassFish server

step1) prepare war file
LinksApp.war

step2) copy the war file or deployment directory structure to <GlassFish_home>\domains\GFNTAJ115Domain1\autodeploy folder.

step3) TCat the web application
http://localhost:5656/LinksApp/

=>To undeploy the web application that is deployed hard deployment process just delete war file or directory from <GlassFish_home>\domains\GFNTAJ115Domain1\autodeploy

Types of Java web applications deployment

a) Hard Deployment
b) console deployment
c) Tool based deployment (Best —> deployment using IDEs, Ant tool or maven tool or gradle tool)

Procedure to cfg GlassFish server with Eclipse IDE

=> As of now (in latest and 2021-xx) Eclipse IDE versions there is no provoision to cfg GlassFiserver with eclipse IDE

Wildfly server

type :: Applicaiton server
vendor :: RedHat
Latest version : 25.x (compatable with jdk 10+)
open source
default ports :: 9990 (for admin console)
8080 (for req,res opeations)
To download wildfly server :: Download as zip file from
https://www.wildfly.org/downloads/

=>No provision to create domains
=> As of now it is not supporting hard deployment.
=>To install wildfly server :: Extract the zip file

After Installation activities in wildfly server

1) creeate admin username and password.

```
What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): (a)

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : nitraja123
Password recommendations are listed below. To modify these restrictions edit the add-user.properties con

 - The password should be different from the username
 - The password should not be one of the following restricted values {root, admin, administrator}
 - The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alpha
)
Password : 123rajanik*
Re-enter Password : 123rajanik*
```

2) change the Http Port number where we give request and response.

=>search for standalone.xml file in <Wildfly_home> diretory —>open file —>
search for the following tag and change port number

```
<socket-binding name="http" port="${jboss.http.port:9990}"/>
                                                  6677
```

3) To start Wildfly server

a) make sure that JAVA_HOME env.. variable is having jdk10+ home directory as the value

b) use <Wildfly_home>\bin\standalone.bat file to start server..

Procedure to perform console deployment of java web application in wildfly server ?

step1) prepare war file having java10+ java compiler
LinksApp.war

step2) start wildfly server (refer above)

step3) open admin console and deploy the web application..
http://localhost:9990 > submit username :: nitraja123
password : 123rajanik* —> deployments —>start —>
—>(+) deploy the web application —> select the war file (LinksApp.war) —>next —>next —>finish.

step4) Test the web application
http://localhost:6677/LinksApp (or)
http://127.0.0.1:6677/LinksApp

To undeploy the webapplication from admin console

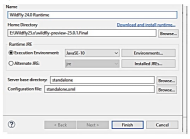
Procedure to cfg Wildfly server with eclipse IDE

step1) install wildfly server adapter to eclipse IDE
window menu —>preferences —> servers —>runtime —> add —>
Redhat middlware —> wildfly ,jboss and other server tools —> next —> accept terms and conditions—>

Front End technologies	middleware technologies	Backend technologies
html ,css, java script ajax, jquery, angular , react js and etc..	servlet, jsp, EJB, webservices, Java mail , jmdi, jdbc and etc... spring , spring boor, hibernate and etc.. Tomcat, weblogic,glassfish, wildfly and etc..	Db s/w, mail server , jndi registries , DNS servers and etc..

server side technologies= middlware technologies

step2) cfg wildfly server in eclipse IDE..
window menu —>preferences —> servers —>runtime —> add —>
Jboss Community —> select wildfly 24+ —>



step3) add wildfly server to "servers" tab
right click —>new server —> jboss community —> wildfly 24+ —> ....

step4) Run the web application by choosing Wildfly server..
a) change Project web module version 4.0 from 5.0
Right click on Project —>properties —>

b) run on server
Right click on project —> run as — run on server

jar file :: jar archive (represents ejb comp as the deployable comp in the server)
jar file also represtnes standalone project, apis/libraries , jdbc drivers and etc..
war file :: web application archive file

ear file ::(enterprise application archieve) jar file + war file (or) war file +war file (or) jar file + jar file

Application server = web server ++

web server	applicaton server
a) Given to manage and execute only web applications (war file)	a) Given to manage and execute web applicaitons(war file) ejb comps (jar file) and enterprise applications (ear file)s
b) contains only servlet, jsp containers	b) contains servlet container, jsp container , CJD container and etc..
c) Developed using servlet api, jsp api	c) developed using servlet api,jsp api, ejb api, jms api, and etc..
d) allows only browser as the clients	d) allows browser , deskop apps, mobile apps and etc.. as the clients
e) does not provide mail server to store mail messages	e) Provide mail servers
f) gives less built-in middlware services like security and etc..	f) Gives more built-in middleware services..
g) most of web servers do not allow domains creation	g Allows the domains creation
h) supports only http, https protocols	h) supports http ,https, iiop , jrmp ,smtp, imap and etc.. protocols https: http over ssl iiop ::internet inter ORB protocol smtp :: simple mail transfer protocol Imap :: internet mail access protocol
i) Bit small is size	i) Bit heavy in size
j) Bit affordable (cost wise)	j) Bit costly
k) Tomcat (upto 6) , jws , jetty , jons and etc.. undertow	k) Tomcat (from 7) , glassfish , weblogic, websphere,wildfly ,jboss and etc..

=>Upto 6 version :: Tomcat is web server s/w
=> from 7 version becoz more facilities are added in terms of middleware services and support to webservices.. so we call it as application server...

Jndi registry
==========

JNDI :: Java Naming And Directory Interface

Nov 29 Server Managed JDBC Con pool

=> It is special registry where java objects references can be placed to provide global visbility and accessibility to objects and object references.

=> The object created in method definitation is local to method
=> The object created as instance variable is specific to object
=> The object created as static variable is specific to class
=>The object placed in Jndi registry is visible and accessible globally ..
Generally the more reusable distributed comps/Entrise comps/ b.comp will be placed Jndi regsitry to provide global visibility and accessbility for them
usecases :: placing payTM object ref in Jndi registry
placing payPall object ref in jndi registry

example Jndi registries

RMI Registry ,COS Registry and etc..
note:: Every Java web server or Application server gives one built-in jndi registry
eg:: Tomcat supplied Jndi registry
GlassFish supplied jndi registry and etc..

Jndi registry maintains the object ref having nicknames or alias names or jndi names

Jndi Registry

keys (jndi names)	values (Object references)
PayTM	PayTM object ref
payPall	PayPall object ref
gpay	GooglePay object ref

Java App ----------Jdbc api----------> DB s/w

Java App ----------JNDI api----------> Jndi regsuy s/w

jdbc api pkgs :: java.sql , javax.sql and their sub pkgs (part of JSE module)
jndi api pkgs :: javax.naming and its sub pkgs

=>java App uses jdbc api to peform jdbc operations (CURD Operations) on DB s/w
=>java App uses jndi api to peform jndi operations on Jndi registry s/w
The Jndi opeartions are
---
bind :: Placing object ref in the jndi registry having nickname/alias name/jndi name
rebind: Replacing the existing object ref with new object ref in the jndi registry
unbind: Removing existing object ref from jndi registry
Lookup :: searching and getting object ref from jndi registry based on the given nick name/alias name
s

=> In jdbc app , the jdbc con object represents connectivity with Db s/w and provide base to create other jdbc objects and to develope persistence logic.

=>In Jndi app (java app with jndi api code) , the InitialContext object represents connectivity with Jndi registry and provides base to perform jndi operations

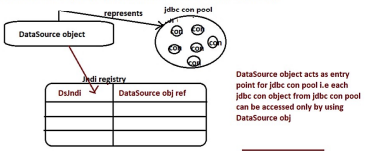
---

of
Two types jdbc con pools

1.Standalone JDBC con pools
=>suitable for standalone Apps
eg:: apache dbcp,
c3P0,
hikari cp(best)
and etc..

2.Server Managed JDBC con pool
=>Created and managed in web server/Application server
=> suitable for web applications, Restfull Apps that are deployable in the servers
eg:: Tomcat managed jdbc con pool
GlassFish managed jdbc con pool
Wildfly managed jdbc con pool
and etc..

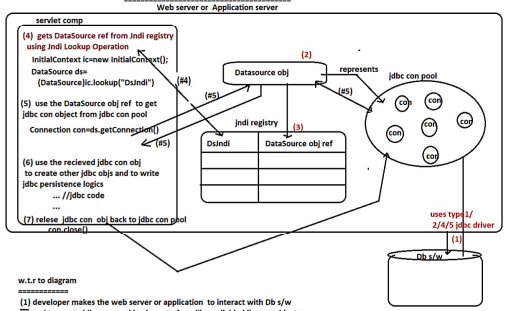
=>Every JDBC Con pool will be represented by DataSource object and that DataSource object ref will be placed in Jndi registry having nick name or alias name for global visibility.



DataSource object acts as entry point for jdbc con pool i.e each jdbc con object from jdbc con pool can be accessed only by using DataSource obj

=>DataSource object means it is the object java class that implements javax.sql.DataSource(I)

understanding server managed jdbc con pool
==========
Web server or Application server



w.r.t r to diagram
==========
(1) developer makes the web server or application to interact with Db s/w and to create jdbc con pool having set of readily available jdbc con objects

(2) developer creates JDBC DataSource obj represting jdbc con pool

(3) Developer keeps DataSource obj ref in the Jndi registry having nick name or alias name..

Procedure to create data soruce object representing jdbc con pool for oracle in GlassFish 6.x server
==========

step1) place ojdbc6/7/8.jar file in <GlassFish_home>\domains\GFNTAJ115Domain1\lib\ext folder (also in "lib" , applibs folders)

step2) start Glassfish domains server

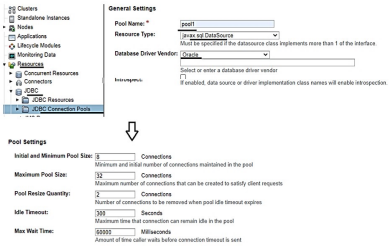
```
E:\GlassFish6.2\glassfish6\glassfish\bin>asadmin start-domain GFNTAJ115Domain1
```

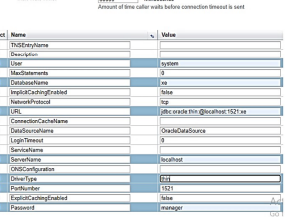
step3) Open admin console by submitting username and password

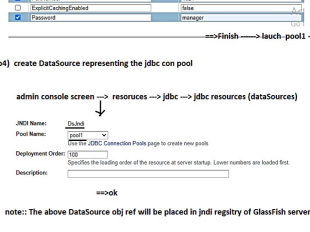
http://localhost:4646 --->
username:testuser
password :testuser
-->submit

step4) create JDBC con pool for oracle

open --> admin screen --->







==>Finish ------> lauch -pool1 --> ping

step4) create DataSource representing the jdbc con pool

admin console screen --> resoruces --> jdbc --> jdbc resources (dataSources)

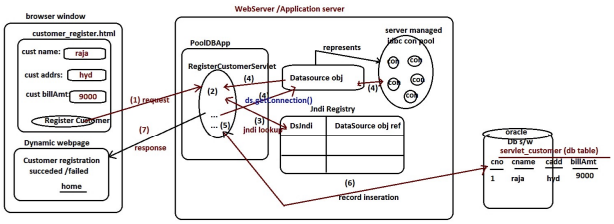
==>ok

note:: The above DataSource obj ref will be placed in jndi regsitry of GlassFish server automatically

Example App on servlet to Db interaction using apporach3

Nov 30 Server Managed JDBC Con pool

Approach3: get Pooled jdbc con being from service(-,-)/doXxx(-,-) methods and create other jdbc objs and write persistene logic in the same service(-,-)/doXxx(-,-) methods.

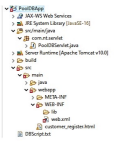


servlet_customer1

	COLUMN_NAME	DATA_TYPE
1	CNO	NUMBER(10, 0)
2	CNAME	VARCHAR2(20 BYTE)
3	CADD	VARCHAR2(20 BYTE)
4	BILLAMT	FLOAT

```
CREATE TABLE "SYSTEM"."SERVLET_CUSTOMER1"
(    "CNO" NUMBER(10,0) NOT NULL ENABLE,
     "CNAME" VARCHAR2(20 BYTE),
     "CADD" VARCHAR2(20 BYTE),
     "BILLAMT" FLOAT(126),
     CONSTRAINT "SERVLET_CUSTOMER1_PK" PRIMARY KEY ("CNO"))
```

```
CREATE SEQUENCE  "SYSTEM"."CNO1_SEQ"  MINVALUE 1 MAXVALUE 10000 INCREMENT BY 1
START WITH 1 CACHE 20 NOORDER  NOCYCLE ;
```



customer_register.html

```
<h1 style="color:blue;text-align:center">Html to servlet communication  using Server managed jdbc con
pool </h1>


<form action="poolurl" method="POST">
  <table border="1"  align="center" bgcolor="cyan">
    <tr>
     <td> customer name :: </td>
     <td> <input type="text" name="cname"> </td>
    </tr>
    <tr>
     <td> customer addrs :: </td>
     <td> <input type="text" name="cadd"> </td>
    </tr>
    <tr>
     <td> customer Billamt :: </td>
     <td> <input type="number" name="billAmt"> </td>
    </tr>
    <tr>
     <td> <input type="submit"  value="Register Customer"> </td>
     <td> <input type="reset" value="cancel"> </td>
    </tr>
  </table>

</form>
```

PoolDBServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;

import javax.naming.InitialContext;
import javax.sql.DataSource;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class PoolDBServlet extends HttpServlet {
	private static final String  INSERT_CUSTOMER_QUERY="INSERT INTO SERVLET_CUSTOMER1 VALUES(CNO1_SEQ.NEXTVAL,?,?,?)";

	@Override
	public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
		//get PrintWriter object
		PrintWriter pw=res.getWriter();
		//set response content type
		res.setContentType("text/html");
		//read form data
		String name=req.getParameter("cname");
		String addr=req.getParameter("cadd");
		float billAmt=Float.parseFloat(req.getParameter("billAmt"));
		//get JDBC con object
		try(Connection con=getPooledJDBCConnection();
			PreparedStatement ps=con.prepareStatement(INSERT_CUSTOMER_QUERY);
			){
			//set query param values
			ps.setString(1,name);
			ps.setString(2, addrs);
			ps.setFloat(3,billAmt);
			//execute the query
			int result=ps.executeUpdate();
			//process the Result
			if(result==0)
				pw.println("<h1 style='color:red;text-align:center'> Problem in Customer  registration </h1>");
			else
				pw.println("<h1 style='color:red;text-align:center'>  Customer  registration completed </h1>");

			pw.println("<br> <a href='customer_register.html'> regisetr customer </a>");
			//close stream
			pw.close();

		}//try
		catch(Exception e) {
			e.printStackTrace();
			pw.println("<h1 style='color:red;text-align:center'> Intenal Problem  -Try again </h1>");
		}


	}

	@Override
	public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
	doGet(req,res);
	}


	//helper method to use only in the current class
	private  Connection getPooledJDBCConnection() throws Exception{
			//create InitialContext object to establish the connection with  jndi registry
			InitialContext ic=new InitialContext();
			//perform jndi lookup operation and get DataSource obj ref from jndi registry
			DataSource ds=(DataSource) ic.lookup("DsJndi");
			//get  Pooled jdbc con object
			Connection con=ds.getConnection();
			return con;
	}

}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
xmlns="http://java.sun.com/xml/ns/j2ee"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

      <servlet>
       <servlet-name>pool</servlet-name>
       <servlet-class>com.nt.servlet.PoolDBServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>pool</servlet-name>
    <url-pattern>/poolurl</url-pattern>
  </servlet-mapping>

      <welcome-file-list>
            <welcome-file>customer_register.html</welcome-file>
      </welcome-file-list>

</web-app>
```

Deploy the above web application in GlassFish server by generating the war file using export option

a) change project's java version to java10
   right click on project ---> properties ---> Project facets ---> java compiler version ---> 10

b) Export the war file to <GlassFish_home>\domains\GFNTAJ115Domain\autodeploy folder

c) Start the domain server..
   E:\GlassFish6.2\glassfish6\glassfish\bin>asadmin start-domain GFNTAJ115Domain1

d) Test the web application..
   url in browser :: http://localhost:5656/PoolDBApp

10.x

Procedure to create jdbc con pool for oracle and JDBC dataSource in tomcat server

step1) make sure that following <Resource> tag is added to context.xml file
(Eclipse Project explorer ->server -->Tomcat ->context.xml)

Jndi name

```
<Resource name="DsJndi" auth="Container"
    type="javax.sql.DataSource" driverClassName="oracle.jdbc.driver.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1522:xe"
    username="system" password="manager" maxTotal="20" maxIdle="10"
    maxWaitMillis="-1"/>
```

waits infinetly for con obj

collect from file:///E:/Tomcat10.x/webapps/docs/jndi-datasource-examples-howto.html

In Tomcat server the given jndi name should be used having fixed prefix that "java:/comp/env"
For example . if the aiven indi name is "DsJndi" then the complete Jndi name would be "java:/comp/env/DsJndi"

step2) Restart the server , if the server is running already

step3) Develop and execute the Application by specifying the jndi name in the above notation

```
// DataSource ds=(DataSource) ic.lookup("DsJndi"); //other than  tomcat server
DataSource ds=(DataSource) ic.lookup("java:/comp/env/DsJndi"); //only for  tomcat server
```

a) change java version to 10 using Project facets
b) Run the app by deploying in Tomcat server

c) Test the application..
   http://localhost:3232/PoolDBApp/

Procedure to create JDBC con pool for oracle and DAtaSource in WildFly server

==========

step1) start wildfly server

=>In eclipse IDE --> go to servers tab -->
right click on wildfly --> start

note: after configuring wildfly server with eclipse IDE, the IDE still uses the orginal wildfly server itself i.e does not create copy of wildfly server.. unlike eclipse cfg Tomcat server

step3) open admin console wildfly server by submitting the username and password

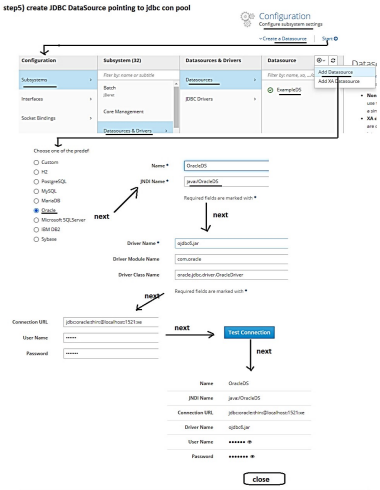
Dec 01 Server Managed JDBC Con pool in wildfly server

http://localhost:9990
username:: nitraja123
password :: 123rajanit*

step4) deploy ojdbc6.jar file to the server

admin console home page--> deployments -->start -->
(+) -->upload deployment --> select ojdbc6.jar from file system --> next -->next --> finish -->close

step5) create JDBC DataSource pointing to jdbc con pool



To test the App

==========

step1) chnage java version of the project to java10 through Project facects..

Right click on Proj (PoolDBApp) -->properties --> Project facets --> java --> 10

step2) change the jndi name in the source code of the Project

DataSource ds=(DataSource) ic.lookup("java:/OracleDS"); //only for wildfly server

step3) Change the java web module version to 4.0 from 5.0

Right click on Proj (PoolDBApp) -->properties --> Project facets --> web module --> 4.0

step4) Run the application.

Right click on Project -->run as --> run on server --> select wild fly server --> ... ...

Annotation driven servlet programming

==========

=> Annotations in servlet programming is introduced from servlet 3.0 . They are given to minimize or avoid xml cfgs in web.xml file
=> if we provide same cfg in both web.xml and annotations the cfgs done in web.xml file will override annotation driven cfgs i.e we can use web.xml cfgs to override the cfgs done using annotations.

The annotations are
a) @WebServlet :: To cfg servlet comp (alternate to <servlet>,<sevlet-mapping> tags)
b) @WebFilter :: To cfg servlet filter comp ((alternate to <filter>,<filter-mapping> tags)
c)@WebInitParam :: To cfg init parm values
d)@WebListener :: To cfg Servlet Listener comp
and etc..

For the following cfgs, we still need to dependend upon web.xml entries
a)welcome files cfg
b) session time out cfgs
c) security cfgs
d) context params
and etc..

Servlet comp cfg using @WebServlet

==========

logical name

```
//@WebServlet("/poolurl")  url pattern
@WebServlet(urlPatterns = "/poolurl",name = "pool",loadOnStartup = 1)
 public class PoolDBServlet extends HttpServlet {                    <l-o-s> priority value
   ....
   ....
 }
```

=>The process of arranging dependent class obj for target class obj is called Depednency Management
=>The class that uses the services of other class is called target class
and the class that acts as helper class is called dependent class.

=>Flipkart (target) needs DTDC courier(dependent) service
=>Vehicle (target) needs Engine(dependent) service
=>Servlet comp (target) needs DataSource (dependent) service
=>Servlet comp (target) needs ServletConfig (dependent) service

Dependency management can be in two ways

a) Using Dependency lookup
b)Using Dependency injection

a) Using Dependency lookup

=>Here target class writes logic and spends some time to search and get dependent class object

eg1:: Student(target) getting coruse material(dependent) by requesting for it
eg2:: servlet comp (target) getting DataSource object(dependent) from Jndi registry using Jndi lookup

code::
InitialContext ic=new InitialContext();
DataSource ds=(DataSource)ic.lookup("DsJndi");

b)Using Dependency injection

=>Here the underlyign server/containe r/framework/JVM dynamically assigns dependent class object to target class object

eg1:: Student(target) getting material(dependent) from NareshIT the moment he joins for the course
eg2:: ServletContainer assiging servletconfig object to our Servlet class object dynamically
eg3:: Making servlet container assigning/injecting DataSource object to our servlet class object by collecting from Jndi registry using @Resource annotation

```
@WebServlet("/poolurl")
public class  CustomerRegistrationServlet extends HttpServlet{
 @Resource(name="DsJndi")
  private  DataSource ds;                    DataSource object is injected here.
   ...
   ..
   .
 }
```

complete servlet comp code

==========

```
//@WebServlet("/poolurl")
@WebServlet(urlPatterns = "/poolurl",name = "pool",loadOnStartup = 1)

public class PoolDBServlet extends HttpServlet {
    private static final String  INSERT_CUSTOMER_QUERY="INSERT INTO SERVLET_CUSTOMER1 VALUES(CNO1_SEQ.NEXTVAL,?,?,?)";

    @Resource(name="DsJndi") //performs dependency injection
    private DataSource ds;                    no need of prefix in jndi name  though we are working with tomcat

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //get PrintWriter object
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read form data
        String name=req.getParameter("cname");
        String addrs=req.getParameter("cadd");
        float billAmt=Float.parseFloat(req.getParameter("billAmt"));
        //get JDBC con object
        try(Connection con=ds.getConnection();
                PreparedStatement ps=con.prepareStatement(INSERT_CUSTOMER_QUERY);
                ){
            //set query param values
            ps.setString(1,name);
            ps.setString(2, addrs);
            ps.setFloat(3,billAmt);
            //execute the query
            int result=ps.executeUpdate();
            //process the Result
            if(result==0)
                pw.println("<h1 style='color:red;text-align:center'> Problem in Customer  registration </h1>");
            else
                pw.println("<h1 style='color:red;text-align:center'>  Customer  registration completed </h1>");

            pw.println("<br> <a href='customer_register.html'> regiswtr customer </a>");
            //close stream
            pw.close();

        }//try
        catch(Exception e) {
            e.printStackTrace();
            pw.println("<h1 style='color:red;text-align:center'> Intenal Problem  -Try again </h1>");
        }


    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
doGet(req,res);
    }


}
```

FileUploading & File Downloading
==========

Dec 02 File Uploading and Downloading

=>The process of selecting files from client machine file system and sending those files to server machine file system is called File Uploading and reverse is called File Downloading

usecases for file uploading

=> profile uploading in matrimony Apps, job portal Apps and etc..
=> uploading photos and files in social networking apps
=> uploading photos and docs in Profile management apps (aadhar apps), Online applications for jobs, govt shemes and etc..
=>Uploading songs , videos to youtube channel

usecases for file downloading

=> donwloaing songs, videos and etc.
=> downloading softwares
=> downloading documents and etc..

To perform file uploading activity we need to "file" comp in the form page

In form page:

```
select file1: <input type="file" name="f1"/>
```

In servlet comp

```
InputStream is=req.getInputStream("f1");

//write the recieved content to destination file
  OutputStream os=new FileOutputStream("abc.txt");
  while((int k=is.read())!=-1){
     os.write(k);
  }
  is.close();
  os.close();
```

is.read() gives "-1" when attempt to read data from EOF (End of File)

Like this we need to write very complex for every file we upload..

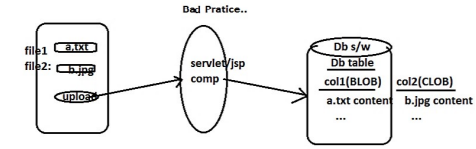
=>Instead of writing the above complex for every file uploading activity , we need to can use one third party api called Java Zoom API for simplification.

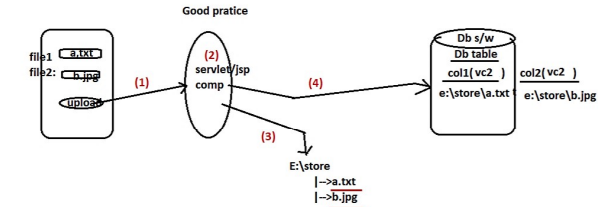
=> while designing form page with file upload comp, we need to take method="POST" and enctype="multipart/form-data"

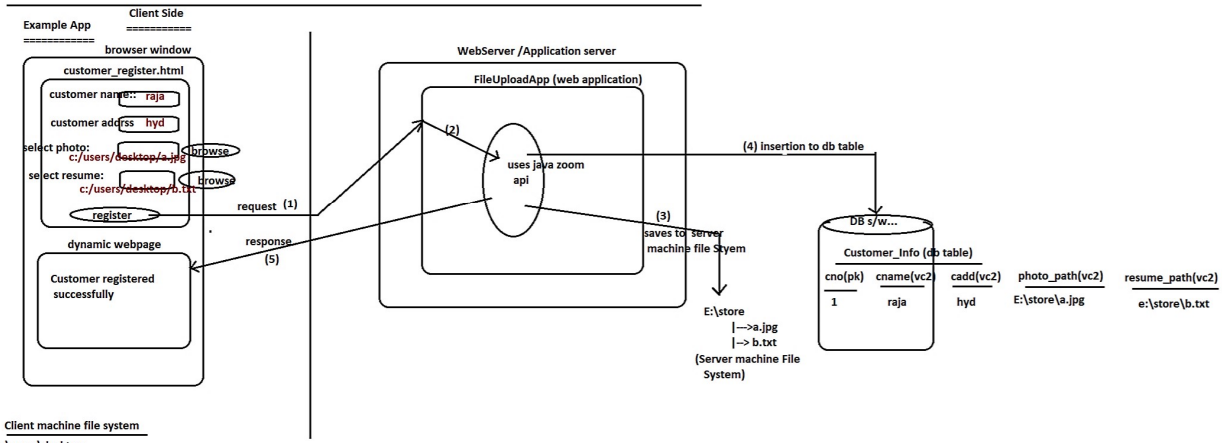
method="POST" (To send any size data along with the request)

enctype="multipart/form-data" (To alert the server that this form page can send different content types/MIME based data becoz along with text data the uploaded file can have any type)

=> Saving Uploaded files directly in Db table cols by taking the col types as the BLOB ,CLOB types is bad pratice becoz BLOB, CLOB type values in db table needs more memory and kills the performence of the Db s/w.. So save the uploaded files in server machine file system and write their address paths to db table cols as String values.







Client machine file system

```
c:\users\desktop
     |-->a.jpg
     |-->b.txt
```

Java zoom API
==========

=>Download this api from javazoom.net as zip file
=> extract zip file and get the jar file

(dependent jar file)

(main jar file)

Here the classes and interfaces of uploadbean.jar file are using the classes and interfaces of cos.jar, fileupload.jar struts.jar files .. So uploadbean.jar is main jar file and the remaining 3 jar files are dependent jar files.

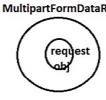
Important classes of javazoom api
==========

a) UploadBean :: To complete file uploading activity by specifying destionation folder to store on the server computer.

b) MultiPartFormDataRequest:: Wrapper around req object to hold both text data and upload files content as streams

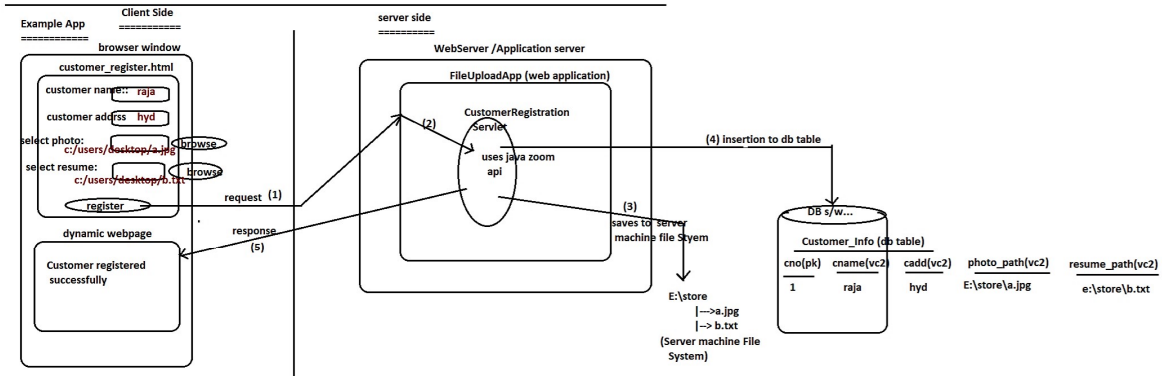
c) UploadFile :: represents each file that is been uploaded
So we can get file info from that like name, size, type and etc..

MultipartFormDataRequest obj



note:: if the web comps of web application are using third party api the we need to add that third party api related main and dependent jar files to the CLASSPATH and also to the WEB-INF\lib folder.

Dec 03 File Uploading and Downloading



Client machine file system
::\users\desktop
|--->a.jpg
|-->b.txt

Java zoom API
=>Download this api from javazoom.net as zip file
=> extract zip file and get the jar file

cos
fileupload
struts
uploadbean
(dependent jar file)
(main jar file)

Here the classes and interfaces of uploadbean.jar file are using the classes and interfaces of cos.jar, fileupload.jar struts.jar files .. So uploadbean.jar is main jar file and the remaining 3 jar files are dependent jar files.

```
// get List of the files that are uploaded
Hashtable<String,UploadFile > ht=nreq.getFiles();
```

cphoto	UploadFile obj
cresume	UploadFile obj

Important classes of javazoom api

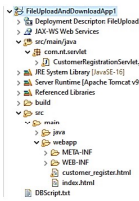
a) UploadBean :: To complete file uploading activity by specifying destionation folder to store on the server computer.

b) MultiPartFormDataRequest:: Wrapper around req object to hold both text data and upload files content as streams

c) UploadFile :: represents each file that is been uploaded So we can get file info from that like name, size, type and etc..

MultipartFormDataRequest obj
request obj

note:: if the web comps of web application are using third party api the we need to add that third party api related main and dependent jar files to the CLASSPATH and also to the WEB-INF\lib folder.



jar files in build path and WEB-INF\lib folder

cos.jar - G:\Sol
fileupload.jar -
ojdbc6.jar - C:
struts.jar - G:\S
uploadbean.ja

index.xml

```
<h1 style="color:red;text-align:center"><a href="customer_register.html"> customer Registration </a></h1>
```

customer_register.html

```
<h1 style="color:red;text-align:center"> Customer Registration page</h1>

<form action="register" method="POST" enctype="multipart/form-data">
  <table border="0" align="center" bgcolor="cyan">
    <tr>
      <td> customer name:: </td>
      <td> <input type="text" name="cname"></td>
    </tr>
    <tr>
      <td> customer address:: </td>
      <td> <input type="text" name="cadd"></td>
    </tr>

    <tr>
      <td>Customer Photo </td>
      <td> <input type="file" name="cphoto"></td>
    </tr>

    <tr>
      <td>Customer Profile </td>
      <td> <input type="file" name="cresume"></td>
    </tr>

    <tr>
      <td colspan="2"><input type="submit" value="upload"> </td>
    </tr>
  </table>

</form>
```

web,xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
id="WebApp_ID" version="4.0">
  <display-name>FileUploadAndDownloadApp1</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

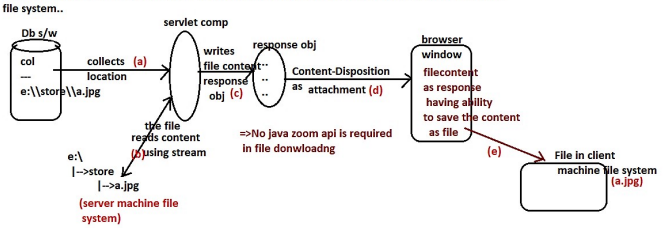
CustomerRegisterationServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.util.Hashtable;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import javazoom.upload.MultipartFormDataRequest;
import javazoom.upload.UploadBean;
import javazoom.upload.UploadFile;


@WebServlet("/register")
public class CustomerRegistrationServlet extends HttpServlet {


    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    //get PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
             res.setContentType("text/html");

                 try {
            //create Special Request object  as wrapper around req object
            MultipartFormDataRequest nreq=new MultipartFormDataRequest(req);

           //read form data (from noramal text boxes)
          String name=nreq.getParameter("cname");
          String addrs=nreq.getParameter("cadd");
            // specify file upload settings
             UploadBean bean=new UploadBean();
             bean.setFolderstore("E:\\store");
             //perform uploading
             bean.store(nreq);

             // get List of the files that are uploaded
              Hashtable<String,UploadFile> ht=nreq.getFiles();
              String filename1=ht.get("cphoto").getFileName();
              String filename2=ht.get("cresume").getFileName();

              pw.println("<br><br> <h1> "+filename1+"  and  "+filename2+" are uploaded succesfully </h1>");

              //write the jdbc code
                Class.forName("oracle.jdbc.driver.OracleDriver");

                try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
                                                               "system", "manager");
                        PreparedStatement ps=con.prepareStatement("INSERT INTO SERVLET_UPLOAD_CUSTOMER VALUES(CNO1
_SEQ.NEXTVAL,?,?,?,?)");
                            ){


                  //set query param value
                    ps.setString(1, name);
                    ps.setString(2,addrs);
                    ps.setString(3, "E:\\store\\"+filename1);
                    ps.setString(4, "E:\\store\\"+filename2);

                    //execute the query
                    int result=ps.executeUpdate();

                  //process the result
                    if(result==0)
                        pw.println("<br><h1 style='color:red;text-align:center'>Customer Registration failed </h1>");

                    else
                        pw.println("<br><h1 style='color:red;text-align:center'>Customer Registration succeded </h1>");

                }//try2
                    }//try1
                catch(Exception e) {
                    e.printStackTrace();
                }

                 //close stream
                  pw.close();

    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
doGet(req,res);
    }

}
```

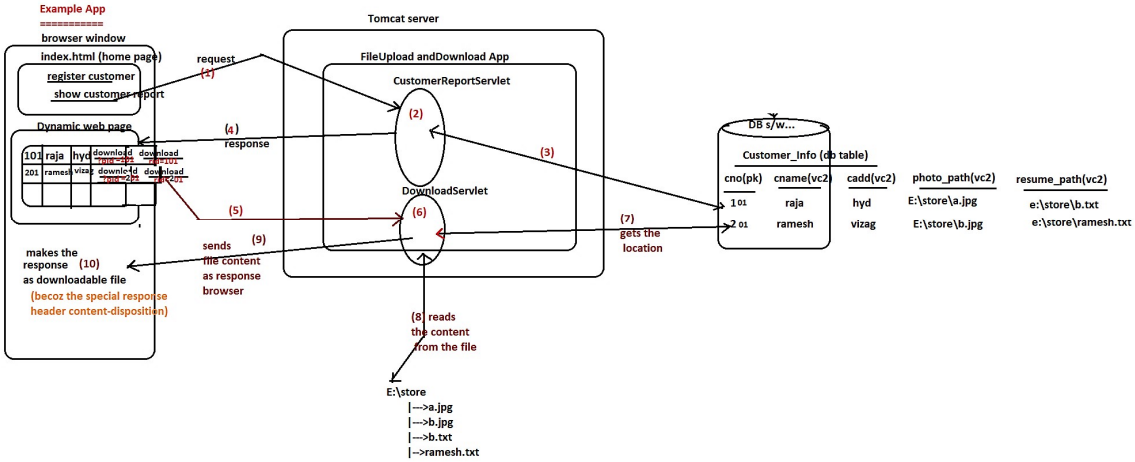
FIeDownloading
==========

Dec 06 File Uploading and Downloading

=>The process of getting the files from server machine file system to client machine file system is called file downloaidng.
=> Get location of the files on server machine file system from db table cols .. then go to that location to read the content of the file and to put in response object (while sending this content to browser make the browser to download the recived content as file in the client machine file system..



Example App
==========



CustomerReportGenerationServlet.java
==========

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/report")
public class CustomerReportGenerationServlet extends HttpServlet {
	private  static final String GET_ALL_CUSTOMERS="SELECT CNO,CNAME,CADD,PHOTO_PATH, RESUME_PATH FROM SERVLET_UPLOAD_CUSTOMER";

	@Override
	public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
	//get PrintWriter
		PrintWriter pw=res.getWriter();
		//set response content type
		res.setContentType("text/html");
		try {
			Class.forName("oracle.jdbc.driver.OracleDriver");
		}//try
		catch(Exception e) {
			e.printStackTrace();
		}
		try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
                                                       "system","manager");
                    PreparedStatement ps=con.prepareStatement(GET_ALL_CUSTOMERS);
                    ResultSet rs=ps.executeQuery();
                    ){
                //process the ResultSet
                pw.println("<table boder='1' align='center' bgcolor='cyan'>");
                pw.println("<tr><th>CNO</th><th>CNAME</th><th>CADD</th> <th>PHOTO</th><th>Resume</th></tr>");
                while(rs.next()) {
                        pw.println("<tr>");
                        pw.println("<td>"+rs.getInt(1)+"</td>");
                        pw.println("<td>"+rs.getString(2)+"</td>");
                        pw.println("<td>"+rs.getString(3)+"</td>");
                        pw.println("<td><a href='download?pid="+rs.getInt(1)+"'>Download Photo </a></td>");
                        pw.println("<td><a href='download?rid="+rs.getInt(1)+"'>Download Resume </a></td>");
                        pw.println("</tr>");
                }
                pw.println("</table>");
		}
		catch(SQLException se) {
			se.printStackTrace();
		}
		catch(Exception e) {
			e.printStackTrace();
		}

		//close stream
		pw.close();


	}

	@Override
	public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
		doGet(req, res);
	}

}
```

==========

Standard procedure to perform file downloading
==========

step1) get the location of the file from db table column

step2) Create inputStream pointing to the file to download by using the location gathered from db table column

step3) create output stream pointing to response obj

step4) get the MIME of the file and make it as the response content type

step5) get the lenght of the file and make it as response content lenght

step6) Give instruction to browser to make the recieved response content as the downloadable file using res.setContentType("Content-Disposition","attachment;fileName=....")

response header

header vlaue possible values are
a)inline(default) :: response should be displayed on the browser
b) attachment :: response should become donwloadble file through browser

step7) Using streams support copy the content of file to response object (any how that content goes to browser as downlodable file)

step8) close all the streams..

Standard procedure to perform file downloading

step1) get the location of the file from db table column

step2) Create InputStream pointing to the file to download by using the location gathered from db table column

step3) create output stream : pointing to response obj

step4) get the MIME of the file and make it as the response content type

step5) get the lenght of the file and make it as response content lenght

step6) Give instruction to browser to make the recieved response content as the downloadable file using res.setContentType("content-Disposition","attachment;fileName=....")

response header

header vlaue
possible values are
a)inline(default) :: response should be displayed on the browser
b) attachment :: response should become donwloadble file through browser

step7) Using streams support copy the content of file to response object (any how that content goes to browser as downloadable file)

step8) close all the streams..

=>Using ByteStream we can write /read both text data and binary data
eg: FileOutputStream ,FileInputStream and etc..
=>Using character we can write /read only text data and binary data
eg: PrintWriter and etc..

CustomerReportGenerationServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/report")
public class CustomerReportGenerationServlet extends HttpServlet {
	private  static final String GET_ALL_CUSTOMERS="SELECT CNO,CNAME,CADD,PHOTO_PATH, RESUME_PATH FROM SERVLET_UPLOAD_CUSTOMER";


	@Override
	public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
		//get PrintWriter
			PrintWriter pw=res.getWriter();
			//set response content type
			res.setContentType("text/html");
			try {
				Class.forName("oracle.jdbc.driver.OracleDriver");
			}//try
			catch(Exception e) {
				e.printStackTrace();
			}
			try(Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
                                                           "system","manager");
				PreparedStatement ps=con.prepareStatement(GET_ALL_CUSTOMERS);
				ResultSet rs=ps.executeQuery();
				){
				//process the ResultSet
				pw.println("<table boder='1' align='center' bgcolor='cyan'>");
				pw.println("<tr><th>CNO</th><th>CNAME</th><th>CADD</th> <th>PHOTO</th><th>Resume</th></tr>");
				while(rs.next()) {
					pw.println("<tr>");
					pw.println("<td>"+rs.getInt(1)+"</td>");
					pw.println("<td>"+rs.getString(2)+"</td>");
					pw.println("<td>"+rs.getString(3)+"</td>");
					pw.println("<td><a href='downloadPid="+rs.getInt(1)+"'>Download Photo </a></td>");
					pw.println("<td><a href='downloadYrId="+rs.getInt(1)+"'>Download Resume </a></td>");
					pw.println("</tr>");
				}
				pw.println("</table>");
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			catch(Exception e) {
				e.printStackTrace();
			}

			//close stream
			pw.close();
		}
		@Override
		public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
```

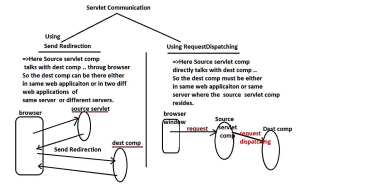
DownloadServlet.java

```
package com.nt.servlet;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.tomcat.util.http.fileupload.IOUtils;


@WebServlet("/download")
public class DownloadServlet extends HttpServlet {
	private static final String GET_PHOTO_PATH_BY_NO = "SELECT PHOTO_PATH FROM SERVLET_UPLOAD_CUSTOMER WHERE CNO=?";
	private static final String GET_RESUME_PATH_BY_NO = "SELECT RESUME_PATH FROM SERVLET_UPLOAD_CUSTOMER WHERE CNO=?";

	@Override
	public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

		// check whether request is coming from download resume or download photo
		// hyperlink or not
		int no = 0;
		String query = null;
		if (req.getParameter("pid") != null) {
			no = Integer.parseInt(req.getParameter("pid"));
			query = GET_PHOTO_PATH_BY_NO;
		} else if (req.getParameter("rid") != null) {
			no = Integer.parseInt(req.getParameter("rid"));
			query = GET_RESUME_PATH_BY_NO;
		}

		// Load jdbc driver class
		try {
			Class.forName("oracle.jdbc.driver.OracleDriver");
		} catch (Exception e) {
			e.printStackTrace();
		}

		try (Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "manager");
				PreparedStatement ps = con.prepareStatement(query)) {
			// set value to query parameter
			ps.setInt(1, no);
			// execute Query
			String FILE_PATH = null;
			try (ResultSet rs = ps.executeQuery()) {
				if (rs.next())
					FILE_PATH = rs.getString(1);

				// create InputStream pointing to File to be downloaded
				try (InputStream is = new FileInputStream(FILE_PATH);
                                                    // create output stream pointing to response object
                                                    OutputStream os = res.getOutputStream()) {
                                                // get the MIME type of file to be downloaded and make it response contentn type
                                                ServletContext sc = getServletContext();
                                                String mimeType = sc.getMimeType(FILE_PATH);
                                                mimeType = (mimeType != null) ? mimeType : "application/octet-stream"; // ternary operation
                                                // get the length of file to be downloaded and make it as response content
                                                // length
                                                File file = new File(FILE_PATH);
                                                res.setContentLength(long)file.length());
                                                // Give instruction to browser to make the recieved response content as the
                                                // downloadable file
                                                res.setHeader("Content-Disposition", "attachment;fileName=" + file.getName());
                                                // copy file content to response object
                                                IOUtils.copy(is, os);

                                            } // try3
                                        } // try2
                                    } // try1
                                    catch (SQLException se) {
                                        se.printStackTrace();
                                    } catch (Exception e) {
                                        e.printStackTrace();
                                    }

                                }
                            }
```

jar files in BUIDPATH
and in WEB-INF\lib foder
commons-io-2.4 .jar
(https://mvnrepository.com/artifact/commons-io/commons-io/2.4)

Servlet Communication

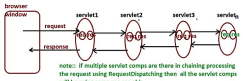
=> It is making the request given to the servlet comp going other web comps from servlet comp.
i.e making servlet comp interacting other web comp to suport request processing



RequestDispatcher based Servlet communincation

if source servlet comp wants to talk with dest sevlet comp.. do not create the object of dest servlet comp in source servlet comp and invoke the service(-,-) method... so use RequestDispather object which make ServletContainer to take of servlet communication..

=>RequestDispatcher object can also be used for servlet chaining..



note:: If multiple servlet comps are there in chaining processing the request using RequestDispatching then all the servlet comps will be using same req,res objs

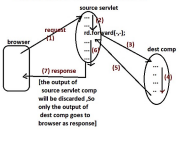
Two types of RequestDispatching

a) Forwarding Request mode of RequestDispatching
b) including Responsse mode of RequestDispatching

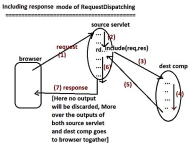
=>create RequestDispticher in source servlet comp having the url of destination comp ..and invoke forward(-,-) on that object to perform forwarding request mode of communication
(rd.forward(req,res))

=>create RequestDispatcher in source servlet comp having the url of destination comp ..and invoke include(-,-) on that object to perform Include response mode of communication
(rd.include(req,res))

forwarding reuest mode of RequestDispatching



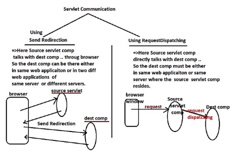
Including response mode of RequestDispatching



Dec 08 Servlet Communication

Servlet Communication
=====================

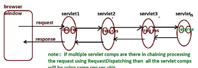
=> it is making the request given to the servlet comp going other web comps from servlet comp.
i.e making servlet comp interacting other web comp to suport request prcoessing



RequestDispatcher based Servlet communincation
==

if source servlet comp wants to talk with dest sevlet comp.. do not create the object of dest servlet comp in source servlet comp and invoke the service(-,-) method... so use ReuestDispather object which make ServletContainer to take care of servlet communuication..

=>RequestDispatcher object can also be used for servlet chaining..



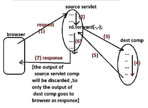
note:: if multiple servlet comps are there in chaining processing the request using RequestDispatching then all the servlet comps will be using same req,res objs

Two types of RequestDispatching
===============================

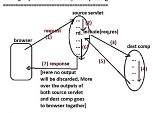
a) Forwarding Request mode of RequestDispatching
b) Including Response mode of RequestDispatching

=>create RequestDisptcher in source servlet comp having the url of destination comp ..and invoke forward(-,-) on that object to perform forwarding request mode of communication (rd.forward(req,res))
=>create RequestDispatcher in source servlet comp having the url of destination comp ..and invoke include(-,-) on that object to perform include response mode of communication (rd.include(req,res))

forwarding reuest mode of RequestDispatching
==



Including response mode of RequestDispatching
===



To perform RequestDispatching based servlet communication, we need RequestDispatcher obj in source servelt comp

Different ways to creating ReqeustDispatcher obj
==

Approach1) using request object

In servlet1 comp (source servlet comp)

```
RequestDispatcher rd=req.getRequestDispatcher("/s2url");
 rd.forward(req,res);
     (or)
 rd.include(req,res);
```

"/" is optional to place
The url pattern of dest comp

Approach2 using ServletContext obj

In Servlet1 comp (source servlet comp)

```
ServletCotnext sc=getServletContext();
RequestDispatcher rd= sc.getRequestDispatcher("/s2url");

 rd.foward(req,res);
   (or)
 rd.include(req,res);
```

"/" is mandatory here
The url pattern of dest comp

=> In approach1 and appraoch2
=>if the dest comp is servlet then we need place its url pattern
=>if the dest comp is jsp page then we can place either its url pattern or file name
=>if the dest comp is html page then we can place either its url pattern or file name

============
Approach3 using ServletContext obj
============

Servlet1 comp (Source Servlet comp)

```
ServletContext sc=getServletContext();
RequestDispatcher rd=sc.getNamedDispatcher("s2");

  rd.forward(req,res);
    (or)
  rd.include(req,res)
```

Logical name of dest web comp which we generally given in <servlet-name> tag of dest web comp cfg.

=>In Approach3 we need to pass
a) logical name of dest servlet comp
b) logical name of dest jsp comp
c) logical name of dest html comp

note:: for jsp ,html comps we get logical name when we cfg them in web.xml file having url pattern.

What is the difference b/w getNamedDispatcher(-) and getRequestDispatcher(-) methods?

getRequestDispatcher(-)	getNamedDispatcher(-)
(a) invokable on both request,ServletContext objs	(a) invokable only on ServletContext obj
(b) we need to pass the url pattern of dest servlet comp as the arg value	b) we need to pass the logical name(obj name) of dest serlet comp as the arg value
(c) if the dest comp is jsp file then we need to pass its file name or url pattern	(c) if the dest comp is jsp file then also we need to pass its logical name as the arg value
(d) if the dest comp is html file then we can pass either url pattern or file name as the argument value	(d) if the dest comp is html file then also we need to pass its logical name as the arg value

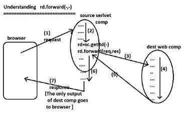
What is difference b/w creating RequestDispatcher obj using request obj and using ServletContext obj?

=>The RequestDispatcher object created using ServletContext obj allows us to keep the Source Servlet comp and the dest web comp either in same web application or in two different web applications of same server.

=>The RequestDispatcher object created using request obj allows us to keep the Source Servlet comp and the dest web comp only in the same web application

=> if Source servlet comp and dest web comp are there in two differnet web applications of two differnent servers beloginging either same machine or different machines then we can not use RequestDispsatcher object.. we must use SendRedirection concept.

Understanding rd.forward(-,-)
=============================



key points
===========

=>rd.forward(-,-) performs forwarding request mode of Servlet communication(RequestDispatching mode)
=>The Source servlet comp and dest web comp uses the same req,res objs
So the request data given to source servlet comp is visible and accessible in dest comp
=>To pass addtional data from source serlvet comp dest comp use request attributes
=> The soruce servlet comp and dest web comp can be there either in same web application or in two different web applications of same server
=> Source servlet comp directly interacts with Dest comp
=> The dest comp can be a servlet comp or jsp comp or html file
=> The statements placed before and after rd.forward(-,-) executes but their html output will be discarded.

=>always place rd.forward(-,-) in source servlet comp as conditional statement to execute
=>usecase:: ErrorServlet configuration

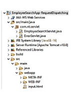
ErrorServlet
==========

=>The Servlet comp that executes only for the exception raised in other main servlet comp is called ErrorServlet comp

=>The ErrorServlet comp is useful to dispaly the exception related messages for enduser as non-technical guiding messages by discarding partial output generated by the MainServlet comp before raising the exception.

if exception raised in main servlet comp , we can display exception related messages as the non-technical guiding message using catch(-) messages.. then what is the need of going for ErrorServlet cfg?

Ans) The messages kept catch(-) block displays excption related message as the non-technical guiding messages but it does not discard partial output messages that are generated before raising the exception
The ErrorSevlet cfg done in catch(-) of main servlet not only displays exception related messages as the non-technical guiding messages and also discards the generated partial outputs (ErrorServlet is good)



//MainServlet comp

```
//EmployeeSearchServlet.java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class EmployeeSearchServlet extends HttpServlet {
	private  static final  String  GET_EMP_INFO="SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE EMPNO=?";
	@Override
	public  void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
		// set response content type
		res.setContentType("text/html");
		//get PrintWriter
		PrintWriter pw=res.getWriter();
		//read form data
		 int eno=Integer.parseInt(req.getParameter("eno"));
		 //write  jdbc code
		 Connection con=null;
		 PreparedStatement ps=null;
		 ResultSet rs=null;
		 try {
			pw.println("<b>hello </b>");
			//Load jdbc driver class
			Class.forName("oracle.jdbc.driver.OracleDriver");
			//establish the connection
			 con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
			 //create Jdbc Statement object
			  ps=con.prepareStatement(GET_EMP_INFO);
			  //set IN param values
			  ps.setInt(1, eno);
			  //execute the SQL query
			  rs=ps.executeQuery();
			  //process the ResultSet
			  if(rs.next()) {
				pw.println("<h1 style='color:red;text-align:center'>"+eno+" Employee Details are </h1>");
				pw.println("<table border='1' align='center'>");
				pw.println("<tr> <td>Empno::  </td> <td>"+rs.getInt(1)+" </td> </tr>");
				pw.println("<tr> <td>Emp Name::  </td> <td>"+rs.getString(2)+" </td> </tr>");
				pw.println("<tr> <td>Emp Desg::  </td> <td>"+rs.getString(3)+" </td> </tr>");
				pw.println("<tr> <td>Emp Salary::  </td> <td>"+rs.getFloat(4)+" </td> </tr>");
				pw.println("<tr> <td>Emp  DeptNo::  </td> <td>"+rs.getInt(5)+" </td> </tr>");
				pw.println("</table>");
			  }//if
			  else {
				pw.println("<h1 style='color:red;text-align:center'> Employee Details are not found </h1>");
			  }//else


		 }//try
		 catch(Exception e) {
			e.printStackTrace();
//forward request to Error sErvlet
			RequestDispatcher rd=req.getRequestDispatcher("/errorurl");
			System.out.println("main servlet :: before rd.forward(-,-) ");
			pw.println("<b>main servlet :: before rd.forward(-,-) </b>");
			rd.forward(req, res);
			System.out.println("main servlet :: after rd.forward(-,-) ");
			pw.println("<b>main servlet ::after rd.forward(-,-) </b>");
		 }
		 finally {
			//close jdbc objs
			try {
				if(rs!=null)
					rs.close();
			}//try
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(ps!=null)
					ps.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
			try {
				if(con!=null)
					con.close();
			}
			catch(SQLException se) {
				se.printStackTrace();
			}
		 }//finally
		 //add  home  hyperlink
		 pw.println("<h1 style='text-align:center'> <a href='input.html'>home </a> </h1>");

		 //close stream
		 pw.close();
	}//doGet(-,-)

	@Override
	public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
	doGet(req,res);
	}//doPost(-,-)

}//class
```

ErrorServlet.java
=================

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/errorurl")
public class ErrorServlet extends HttpServlet {

	@Override
	public  void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
		System.out.println("Error servlet ::doGet(-,-) ");
		//get PrintWriter
		PrintWriter pw=res.getWriter();
		//set response content type
		res.setContentType("text/html");
		//write  non-technical guiding messages to endusers
		pw.println("<h1 style='color:red;text-align:center'> Internal problem  –Try again </h1>");
		pw.println("<br><a href='input.html'>home</a>");

		//close stream
		pw.close();
	}

	@Override
	public  void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
```

Dec 10 Servlet Communication

Different ways of calling forward(-,-) or include(-,-) method on RequestDispacther obj

```
RequestDispatcher rd=req.getRequestDispatcher("/errorurl");
rd.forward(req,res);                                   optional to place

        (or)

ServletContext sc=getServletContext();
RequestDispatcher rd=sc.getRequestDispatcher("/errorurl");
 rd.forward(req,res);                                  mandatory to place

        (or)

    ServletContext sc=getServletContext();
        RequestDispatcher rd=sc.getNamedDispatcher("err");
        rd.forward(req, res);                          :Logical name of the dest web comp
```

if we call

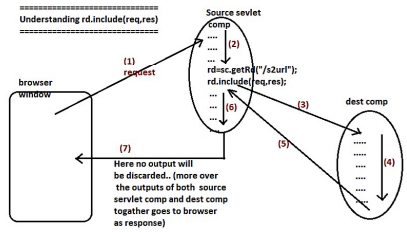
Q)what happends multiple rd.forward(-,-) methods from same source servlet comp pointing to different destinations.

Ans) The dest comp pointed by the first rd.forward(-,-) executes.. and its output goes browser as response through source servlet comp and the call second rd.foward(-,-) throws java.lang.IllegalStateException: Cannot forward after response has been committed

---

The rd.forward(-,-) method called in Source servlet comps performs

a) Discards existing output from the response objects
b) sends the request to dest comp web comp and executes that comp through servletc container
c) keeps the dest comp's output in source servlet comp and commits the response then sends this committed response to browser.



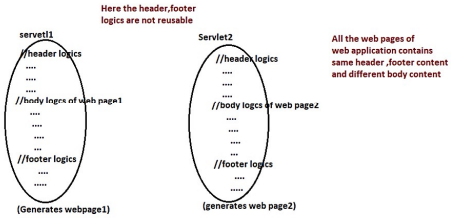
keypoints

=>rd.include(-,-) performs the including response mode of serlvet communication
=>The source servlet comp and the dest comp uses same req,res objs .. so the request data coming to source servlet comp is visible and accessible in dest web comp
=> if source serviet comp wants to send addtional data to dest comp then use request attributes support

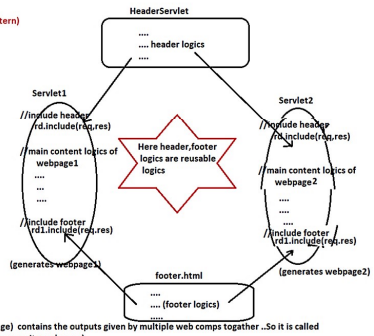
=> The response contains the outputs of both source servlet comp and dest comp
=> The source servlet comp and dest comp can be there either in same web application or in two different web applications of same server
=> The dest comp's output will be included to the source servlet comp in the the place where rd.include(-,-) is called.
[if rd.include(-,-) is called in the middle of source servlet comp then the dest comp output will be included in the the middle of source servlet comp]

=> use-case :: Composite -vew Pattern implentation
(keeping common logics like header,footer logics) in seperate web comp and including their outputs in main web comps)

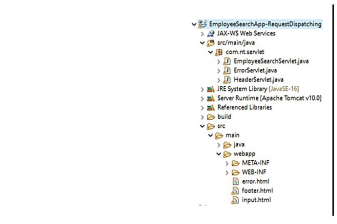
Problem:



Solution::
(Composite view pattern)



Here view(web page) contains the outputs given by multiple web comps togather ..So it is called composite view (composite web page)



footer.html

```
<br><bR><bR><br>

<b><center>&copy; all rigjhts reserved for students 2020-21</center></b>
```

HeaderServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/headurl")
public class HeaderServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //header content
        pw.println("<marquee><h1 style='color:blue'> N A R E S H Technologies  </h1></marquee>");
        //do not close stream
        //pw.close();
    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException {
        doGet(req,res);
    }



}
```

pw.close() :: Not only closes the stream, it commits the output of the response object i.e once the response committed we can add further output to response object. So do not place pw.close() in the Destination comp while working with rd.include(-,-)

What happens if we place both rd.forward(-,-) and rd.include(-,-) in the same source servlet comp?

Ans) Once rd.forward(-,-) executes then it disacrds both generated output and included output of source servlet comp and only the rd.forward(-,-) related dest comp's output goes to browser as response.

//EmployeeSearchServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletContext;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class EmployeeSearchServlet extends HttpServlet {
    private  static final  String  GET_EMP_INFO="SELECT EMPNO,ENAME,JOB,SAL,DEPTNO FROM EMP WHERE EMPNO=?";
    @Override
    public  void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

        // set response content type
        res.setContentType("text/html");
        //get PrintWriter
        PrintWriter pw=res.getWriter();

        Connection con=null;
        PreparedStatement ps=null;
        ResultSet rs=null;

        try {
            //include header
            RequestDispatcher rd=req.getRequestDispatcher("/headurl");
            rd.include(req, res);
        //read form data
        int eno=Integer.parseInt(req.getParameter("eno"));
        //write  jdbc code

            pw.println("<br><br><br>");
            //Load jdbc driver class
            Class.forName("oracle.jdbc.driver.OracleDriver");
            //establish the connection
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system","manager");
            //create Jdbc Statement object
            ps=con.prepareStatement(GET_EMP_INFO);
            //set IN param values
            ps.setInt(1, eno);
            //execute the SQL query
            rs=ps.executeQuery();
            //process the ResultSet
            if(rs.next()) {
                pw.println("<h1 style='color:red;text-align:center'>"+eno+" Employee Details are </h1>");
                pw.println("<table border='1' align='center'>");
                pw.println("<tr> <td>Empno:: </td> <td>"+rs.getInt(1)+" </td> </tr>");
                pw.println("<tr> <td>Emp Name:: </td> <td>"+rs.getString(2)+" </td> </tr>");
                pw.println("<tr> <td>Emp Desg:: </td> <td>"+rs.getString(3)+" </td> </tr>");
                pw.println("<tr> <td>Emp Salary:: </td> <td>"+rs.getFloat(4)+" </td> </tr>");
                pw.println("<tr> <td>Emp  DeptNo:: </td> <td>"+rs.getInt(5)+" </td> </tr>");
                pw.println("</table>");
            }//if
            else {
                pw.println("<h1 style='color:red;text-align:center'> Employee Details are not found </h1>");
            }//else


        }//try
        catch(Exception e) {
                e.printStackTrace();

            RequestDispatcher rd=req.getRequestDispatcher("/errorurl");
            rd.forward(req, res);
        }
        finally {
            //close jdbc objs
            try {
                if(rs!=null)
                    rs.close();
            }//try
            catch(SQLException se) {
                se.printStackTrace();
            }
            try {
                if(ps!=null)
                    ps.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }
            try {
                if(con!=null)
                    con.close();
            }
            catch(SQLException se) {
                se.printStackTrace();
            }
        }//finally
        //add  home  hyperlink
        pw.println("<h1 style='text-align:center'> <a href='input.html'>home </a> </h1>");

            //include header
            RequestDispatcher rd2=req.getRequestDispatcher("/footer.html");
            rd2.include(req, res);

        //close stream
        pw.close();
    }//doGet(-,-)

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    doGet(req,res);
    }//doPost(-,-)

}//class
```

//Main Servlet comp

Dec 11 Servlet Communication

What is happends if we place multiple rd.include(-,-) method calls in single Source servlet comp?

Ans) The output of multiple dest comps will be included to the output of Source servlet comp

if close stream using pw.close() before calling rd.include(-,-) or rd.forward(-,-) tell me what happens?

Ans) pw.close() method no only closes the stream, it also commits the output to the response object..
So after comming the ouput forwaring request using rd.forward(-,-) and including response using rd.include(-,-) not possible becoz the committed output of the response object can not be changed.
if u still try to perform the activity we get IllegalStateException
add
note:: Commited respose can not be discarded and we can not more output to response object once the response is committed.

what is the different b/w rd.foward(-,-) and rd.include(-,-) ?

rd.forward(-,-)	rd.incldue(-,-)
a) performs the forwarding request mode of servlet communication	a) performs the including response mode of servlet communication
b) Discards the output of soruce serlvet comp that is geranted before calling rd.forward(-,-)	b) Does not discard any output
c) The final generated response contains only the output of dest comp	c) The final generated response contains the outputs of source and dest comp togather.
creates d) does not problem if close the stream in dest comp	d) creates the problem if close the stream in dest comp
e) usecase:: Error Servlet configuration	e) usecase:: Composite View Design pattern implementation.

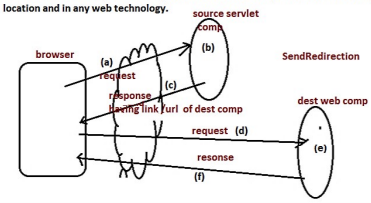
Limitations of RequestDispatching based Servlet Communication
==========

a) The Source servlet comp and dest comp can there either in same web application or in two different web applications of same server .. But can not be there in two web applications two different servers beloging to same or different machines.

b) The dest web comp must be html file or jsp file or servlet comp. it can not be other technologies web comp like php comp, asp.net comp, expressJs comp and etc..

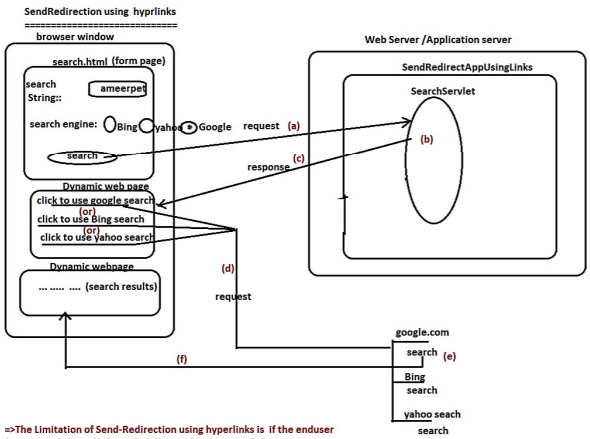
To overcome these problems take the support sendRedirection

SendRedirection makes the source servlet comp talking to dest web comp after having one network round trip with browser . since browser can give request to any technology and location web comp ,so we can place the dest web comp in sendRedirection process in any location and in any web technology.



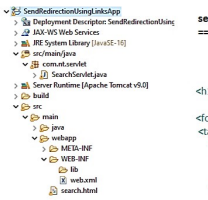
Send Redirection can be done in two ways
a) using hyperlinks
b) using res.sendRedirect(-) method

SendRedirection using hyrplinks
==========



=>The Limitation of Send-Redirection using hyperlinks is if the enduser forgets to click on the hyperlink the sendRedirection fails.

https://www.google.com/search?q=ameerpet
https://www.bing.com/search?q=ameerpet
https://search.yahoo.com/search?p=ameerpet



search.html
==========

```
<h1 style="color:red;text-align:center">SendRedirection using hyperlinks</h1>

<form action="searchurl" method="post">
 <table border="1" align="center" bgcolor="cyan">
    <tr>
      <td> Search String  </td>
      <td> <input type="text" name="ss"> </td>
    </tr>

    <tr>
      <td> Search Engine  </td>
      <td>
        <input type="radio" name="engine"  value="google"  checked>Google Search   
        <input type="radio" name="engine"  value="bing"  >Bing Search   
        <input type="radio" name="engine"  value="yahoo" >yahoo Search   
      </td>
    </tr>

    <tr>
      <td colspan="2"> <input type="submit"  value="send"> </td>
    </tr>
 </table>
</form>
```

SearchSevlet.java
==========

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/searchurl")
public class SearchServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            //get PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read form data
            String ss=req.getParameter("ss");
            String engine=req.getParameter("engine");
            //send hyperlink to browser supporting the sendRedirection
            if(engine.equalsIgnoreCase("google"))
                pw.println("<a href='https://www.google.com/search?q="+ss+"'> click here for google search</a>");
            else if(engine.equalsIgnoreCase("bing"))
             pw.println("<a href='https://www.bing.com/search?q="+ss+"'> click here for Bing search</a>");
            else
                pw.println("<a href='https://search.yahoo.com/search?p="+ss+"'> click here for yahoo search</a>");

            //close stream
                    //close stream
                    pw.close();

        }

        @Override
        public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req,res);
        }

}
```

web.xml
==========

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
version="4.0">
  <display-name>SendRedirectionUsingLinksApp</display-name>
  <welcome-file-list>
    <welcome-file>search.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Dec 14 Send Redirection

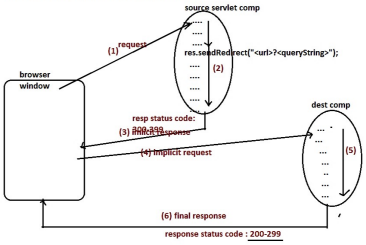
Http response contains
a) HEAD part (response status code, response headers)
b) Body part (output content)

=>The message kept in pw.println(-) becomes the body in the HttpResponse
=>Response headers give special instruction to browser towards recieving and displaying data on the browser
eg: contentType , contentLength , refresh and etc..
=>response status code indicates the state of request processing and response generation

100-199 :: Informational
200-299 :: success *The default response status code is : 200*
300-399 :: Redirection
400-499 :: Incomplete (Client Side errors)
500-599 :: Server side errors

https://developer.mozilla.org/en-US/docs/Web/HTTP/Status *For defalid info on response status codes*

SendRedirection using res.sendRedirect(-) method



w.r.t diagram

(2) All the statements placed in Soruce servlet comp executes including res.sendRedirect(-) method but their html output will be discarded

(3) Source servlet comp sends implicit response to browser having the dest comp url and query String with the response status code: 300-399

(4) Based on the recieved response status code (300-399) The browser does not dispaly any thing browser but uses the url that is coming along with response to generate the next implicit request to dest comp

---

key points

a) res.sendRedirect(-) performs sendRedirection mode Servlet Communication.
b) All the statements placed in res.sendRedirect(-) method executes and their generated output will be discarded
c) The soruce servlet comp and dest comp will not be using same request ,response objects So the requst data coming to source servlet comp is not visible and accessible in the dest comp

d) To pass data from source servlet comp to dest comp place query String to the url of res.sendRedirect(-) method. and use req.getParameter(-) methods to read them in Dest comp.

(e) The source servlet comp and dest comp can be there in the same web application or in the two different web applications of same server or different servers.. belonging to same machine or different machine.

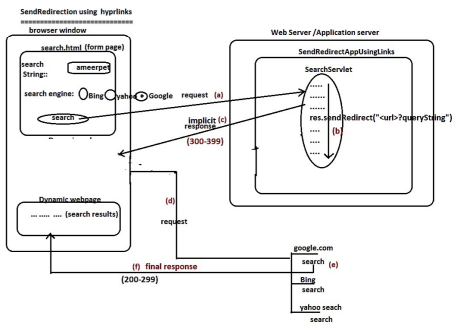
(f) The source servlet comp talks with dest web comp after having network round trip with browser.

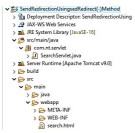
(g) Here , no need of clicking any hyperlinks to complete the send Redirection.,

Usecase:: (a) To use external website services being from Local website with out worrying about the location and technology of external website
eg:: using google search from different webistes
eg:: using google maps from different websistes
(b) If company A aquires company B , the reqeust given to company B websiste will be redirected to company A website.

=>IBM has aquired rational company , so the request given rational.com will be redirected to IBM.com
=>Oracle corp has acquired sun Ms , so the request given to sun.com will be redirected to oracle.com

SendRedirection using hyprlinks





search.html

```
<h1 style="color:red;text-align:center">SendRedirection using hyperlinks</h1>

<form action="searchurl" method="post">
<table border="1" align="center" bgcolor="cyan">
  <tr>
    <td> Search String </td>
    <td> <input type="text" name="ss"> </td>
  </tr>

  <tr>
    <td> Search Engine </td>
    <td>
      <input type="radio" name="engine" value="google" checked>Google Search   
      <input type="radio" name="engine" value="bing" >Bing Search   
      <input type="radio" name="engine" value="yahoo" >yahoo Search   
    </td>
  </tr>

  <tr>
    <td colspan="2"> <input type="submit" value="send"> </td>
  </tr>
</table>
</form>
```

SEarchServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/searchurl")
public class SearchServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read form data
        String ss=req.getParameter("ss");
        String engine=req.getParameter("engine");
        //send hyperlink to browser supporting the sendRedirection
        String url=null;
        if(engine.equalsIgnoreCase("google"))
            url="https://www.google.com/search?q="+ss;
        else if(engine.equalsIgnoreCase("bing"))
            url="https://www.bing.com/search?q="+ss;
        else
            url="https://search.yahoo.com/search?p="+ss;

        // perform sendRedirection
        res.sendRedirect(url);

        //close stream
        pw.close();

    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    doGet(req,res);
    }

}
```

---

what happens if we place multiple res.sendRedirect(-) methods in the same source servlet comp?

Ans) Since first res.sendRedirect(-) method commits the response.. So the second response.sendRedirect(-) throws Exception
(java.lang.IllegalStateException: Cannot call sendRedirect() after the response has been committed)

What happends of we place rd.forward(-,-) and res.sendRedirect(-) method in a source servlet comp?

Ans) Throws IllegalStatement exception

What happends of we place rd.include(-,-) and res.sendRedirect(-) method in a source servlet comp?

Ans) There will not be any effect of rd.include(-,-) becoze the while discarding the response becoz of res.sendRedirect(-) method execution .. It discards both direct and included outputs of Surce servlet comp.

What is the difference b/w rd.forward(-,-) and response.sendRedirect(-) methods?

rd.forward(-,-)	res.sendRedirect(-) method
a) Performs formwarding request mode of servlet communication	a) Performs sendRedirection mode of servlet communication
b)The source servlet comp and dest comp use the same req ,res objs	b)The source servlet comp and dest comp will not use the same req ,res objs
c)Data coming to soruce servlet comp is visible and accessible in dest comp	c)Data coming to soruce servlet comp is not visible and accessible in dest comp
d) To pass addtional data to dest comp being from source sevlet comp use request attributes	d) To pass addtional data to dest comp being from source sevlet comp , append query query String to the url of res.sendRedirect(-) method
(e)The source servlet comp and dest comp can be there either in same web applicalton or in two different web applications of same server	(e)The source servlet comp and dest comp can be there either in same web applicalton or in two different web applications of same server or different dservers. These servers can be there either in same machine or in different machines
(f) While forwaring requst the url in the browser address bar will not changed	(f) While performing sendirection the url in the browser address bar changed to new.
(g) source servlet comp directly interacts with Dest comp	(g) source servlet comp Interacts with Dest comp through browser
h) usecase :: Error Servlet cfg To forward to jsp/html page in layered App	h)usecases :: To use extenal web site conent befing from Localwebsite

How to Pass Data from Source servlet comp to Dest comp? **Dec 16 Send Redirection**

if Source servlet comp and dest web comp are there in the same web application
==

a)request attributes (if source servlet comp and dest web comp are using same req,res objs)
b)session attributes (if source servlet comp and dest web comp are getting request from same browser)
c)application attributes (If source servlet comp and dest web comp are not using same req,res
ServletContext attributes objs and not getting request from same browser)

if the Source servlet comp and dest web comp are there in two different web applications of same server or different servers
==

Append query String to the url of res.sendRedirect(-) method

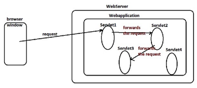
res.sendRedirect(url?<query String>)

=>Attribute is a logical name (string name) that contains object as value in a scope (request scope/session scope/application).

request scope --> specific to each request
session scope --> specific to each browser s/w of client machine
application scope --> specific to each web application.

=>Atribute in servlet programming is no way related to html ,xml tag attributes

request attributes
===============
=>allocates memory in request object
=> contains request scope i.e visible through out request i.e
request scope data can be accessed in all web comps that are using
same request object while processing a request.



=>The request attribute created in Servlet1 is visible and accessible in Servlet2, servlet3 while processing the request given servlet1 but not in Servlet4 becoz servlet1,servlet2,Servlet3 comps are going to use same req,res objects while processing the request given to servlet1 comp.

note: Auto Boxing :: The Process of converting simple value to object .
Auto Unboxing :: The Process of converting Object to simple value .

In core java App/stand alone Apps we can keep data in following scopes
a) Local scope
b) instance scope
c) class scope
d) Thread Scope

To create request attribute
=======================

```
req.setAttribute("attr1","val1");
                            attr value
                            attr name

req.setAttribute("age",30);
```
Will be converted into Integer Wrapper class obj (30)
(Auto Boxing)

To modify the request attribute values
==================================

```
req.setAttribute("attr1","val11");
req.setAttribute(" age ",32);
```

=>setAttribute(-,-) method is multi-purpose method i.e it creates the attribute in the object/scope if the attribute is not already there otherwise it modifies the existing attribute value.

To read request attribute values
===========================

```
String value1=(String)req.getAttribute("attr1");
int age=(Integer)req.getAttribute("age");
```
Auto unboxing

To remove request attributes
=========================

```
req.removeAttribute("attr1");
req.removeAttribute("age");
```

//Servlet1.java

```
@WebServlet("/s1url")
public class Servlet1 extends  HttpServlet{

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       //create request attributes
            req.setAttribute("attr1", "val1");

            //forward request to SecondSeervlet
            RequestDispatcher rd=req.getRequestDispatcher("/s2url");
            rd.forward(req, res);

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
     doGet(req,res);
      }

}
```

//Servlet2.java

```
@WebServlet("/s2url")
public class Servlet2  extends HttpServlet {

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
     //GEt PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read and display the request attribute values
            pw.println("<h1 style='color:red;text-align:center'>Servlet2::: attr1(req) value ::"+req.getAttribute("attr1")+"</h1>");


             //forward the request to Servlet3
            RequestDispatcher rd=req.getRequestDispatcher("/s3url");
            rd.forward(req,res);

            //      close stream
            pw.close();

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       doGet(req,res);
      }

}
```

Sevlet3.java

```
@WebServlet("/s3url")
public class Servlet3  extends HttpServlet {

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
     //GEt PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read and display the request attribute values
            pw.println("<h1 style='color:red;text-align:center'>Servlet3::: attr1(req) value ::"+req.getAttribute("attr1")+"</h1>");

            //close stream
            pw.close();

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       doGet(req,res);
      }

}
```

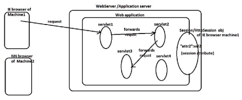
SEvlet4.java

```
@WebServlet("/s4url")
public class Servlet4  extends HttpServlet {

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
     //GEt PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read and display the request attribute values
            pw.println("<h1 style='color:red;text-align:center'>Servlet4::: attr1(req) value ::"+req.getAttribute("attr1")+"</h1>");

            //close stream
            pw.close();

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
       doGet(req,res);
      }

}
```

Give first request to Servlet1
and given remaing qeuests other sevlets and check the visbility of reques attributes

Session attributes
=================

=> Allocates memory in Session /HttpSession object
=> The Session/HttpSession obj and its session attributes are visible and accessible in all the web comps of web application irrespective the req,res objects they are using but they must get requst from same browser for whom Session object and its attributes are created.
=>session attributes scope session scope i.e they are very much specifc to browser s/w of a client machine



The Session atribute created in Servlet1 comp by getting request from IE browser of machine1 is visible and acccessble in all otherweb comps of the web application only when they get request from IE broser of machine1.

To create Session attribute
=======================
```
ses.setAttribute("user","raja");
```

To modify Session attribute
=======================
```
ses.setAttribute("user","raja11");
```

tabs and windows of browser s/w in the client machine will not be treated as seperate clients ..they are part of browser s/w as client.

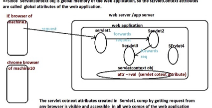
To read Session attribute
=======================
```
String value=(String) ses.getAttribute("user");
```

To remove Session attribute
=======================
```
ses.removeAttribute("user");
```

ServletContext attributes/application attributes
===

=> These attributes allocate memory in ServletContext object and known as global attributes becoz attributes are visible and accessible in all web comps ofweb application irrespective of req,res objs they are using and the browser s/w from which they are getting request.

=>Since Servletcontext obj is global memory of the web application, so the sErvletContext attributes are called global attributes of the web application.



The servlet cotnext attributes created in Servlet1 comp by getting request from any browser is visible and accessible in all web comps of the web application irrespective of any condition i.e irrespective of req,res objs they are using and irrespective of the browser window from which they are request (application scope)

To create sErvletCotnext/application attribute
==============================
```
ServletCotntext sc=getServletContext();
sc.setAttribute("attr1","val1");
sc.setAttribute("admin", "raja");
```

To modify the ServletContext attribute value
==
```
sc.setAttribute("attr1","val1111");
sc.setAttribute("admin","ramesh");
```

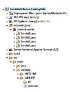
To read ServletContext attribute vlaues
==================================
```
String data1=(String)sc.getAttribute("attr1");
String  data2=(String)sc.getAtribute("admin");
```

To remove ServletContext attributes
=================================
```
sc.removeAttribute("attr1");
sc.removeAttribute("admin");
```

usecases of servlet attributes
=========================
=>In layered App ,if servlet comp generates results /gather results and passes the results to view comp by keeping the results in request scope.

=>The login credentials like username,password will be placed in session scope becoz login activity is always specific to server.

=>The global arithmetic and logical opeartions with respect to all web comps like request count should be taken as the servletcontext attributes,.



servlet1.java

```
package com.nt.servlet;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/s1url")
public class Servlet1 extends  HttpServlet{

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //create request attributes
            req.setAttribute("attr1", "val1");

            //create Session attributes
            HttpSession ses=req.getSession();
            ses.setAttribute("attr2","val2");

           //create ServletContext attributes
            ServletContext sc=getServletContext();
            sc.setAttribute("attr3","val3");



            //forward request to SecondSeervlet
            RequestDispatcher rd=req.getRequestDispatcher("/s2url");
            rd.forward(req, res);


      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
     doGet(req,res);
      }

}
```

SErvlet2.java
==============

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/s2url")
public class Servlet2  extends HttpServlet {

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {
      //GEt PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read and display the request attribute values
            pw.println("<h1 style='color:red;text-align:center'>Servlet2:::
attr1(req) value ::"+req.getAttribute("attr1")+"</h1>");
           //read and display the session attribute values
            HttpSession ses=req.getSession();
            pw.println("<h1 style='color:red;text-align:center'>Servlet2:::
attr2(session) value :"+ses.getAttribute("attr2")+"</h1>");

                //read and display the session attribute values
            ServletContext sc=getServletContext();
            pw.println("<h1 style='color:red;text-align:center'>Servlet2:::
attr3(ServletContext) value :"+sc.getAttribute("attr3")+"</h1>");


             //forward the request to Servlet3
            RequestDispatcher rd=req.getRequestDispatcher("/s3url");
            rd.forward(req,res);

            //      close stream
            pw.close();

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException {
         doGet(req,res);
      }

}
```

SErvlet3.java
============

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/s3url")
public class Servlet3  extends HttpServlet {

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
      //GEt PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read and display the request attribute values
            pw.println("<h1 style='color:red;text-align:center'>Servlet3::: attr1(req) value ::"+req.getAttribute("attr1")+"</h1>");
                //read and display the session attribute values
            HttpSession ses=req.getSession();
            pw.println("<h1 style='color:red;text-align:center'>Servlet3::: attr2(session) value :"+ses.getAttribute("attr2")+"</h1>");

                //read and display the session attribute values
            ServletContext sc=getServletContext();
            pw.println("<h1 style='color:red;text-align:center'>Servlet2::: attr3(ServletContext) value :"+sc.getAttribute("attr3")+"</h1>");

            //close stream
            pw.close();

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req,res);
      }
```

Servlet4.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/s3url")
public class Servlet3  extends HttpServlet {

      @Override
      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
      //GEt PrintWriter
            PrintWriter pw=res.getWriter();
            //set response content type
            res.setContentType("text/html");
            //read and display the request attribute values
            pw.println("<h1 style='color:red;text-align:center'>Servlet3::: attr1(req) value
::"+req.getAttribute("attr1")+"</h1>");
                //read and display the session attribute values
            HttpSession ses=req.getSession();
            pw.println("<h1 style='color:red;text-align:center'>Servlet3::: attr2(session) value
:"+ses.getAttribute("attr2")+"</h1>");

                //read and display the session attribute values
            ServletContext sc=getServletContext();
            pw.println("<h1 style='color:red;text-align:center'>Servlet2::: attr3(ServletContext) value
:"+sc.getAttribute("attr3")+"</h1>");

            //close stream
            pw.close();

      }

      @Override
      public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req,res);
      }

}
```

Different ways of registrering servlet comp and mapping with url pattern
===

a) Declarative Approach
[Using web.xml tags like <servlet> and <servlet-mapping> tags
b) Annotation Approach
[Using @WebSErvlet annotation]
c) Promgramatic Approach /Dynamic Servlet Registration
[Using java code like sc.addServlet(-,-) method)
watch this video ::
Dynamic or Programatic registration of servlet comp
----
https://www.youtube.com/watch?v=aRscgj17Hul&t=2102s

=> if the servlet comp is user-defined sevlet comp then go for annotation approach.

=> if the servlet comp is pre-defined servlet and xml cfgs are allowed in in the application then go for declarative aprpoach (web.xml cfgs)

=> if the servlet comp is pre-defined servlet and xml cfgs are not allowed in in the application then go for programatic aprpoach (sc.addServlet(-,-))

examples for pre-definedSevlet are DispatcherServlet (spring MVC),
ActionServlet (struts)
FacesServlet (jsf)
and etc..

Session Tracking /SEssion Management

Dec 18 Need of Session tracking

Session
=======

=>Set of related and continuos operations performed by the Client /browser in a web application is called session.
eg: Login to logout
eg: start of game to end of game
eg: start of class to end of class

=>If the Web application is remebering client data across the multiple requests then it is called statefull web application.
=>In stateful web application , every request can use previous request data i.e while processing 2nd request we can 1 req data ,similarly while processing 3rd request we can use 1st and 2nd requests data.

=>if web application is not remembering client data across the multiple requests then it is called statless web application ..

=> In stateless web application every request can use current request data .. but can not use previous requests data.

note:: By default every web application is stateless becoz the protocol http is stateless.. To make stateless web application as stateful web application we need the support of Session Tracking or Session Management.

Two types of forms
a)static form
b) dynamic form

=>The form page whose content is fixed is called static form page
-> .html files generate static form pages

=>The form page that generated as the ouput given by servlet or jsp comp having dynamic content/vaying content is called dynamic form page.

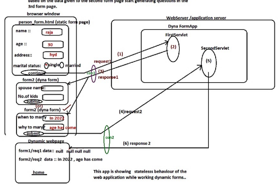
in servlet comp the <form> , <input>tags pplaced in pw.println(-) method calls generate dynamic form i.e form can contain our choice comps as needed.

problem in static fomr page
----

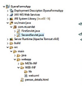
person_details.html (static form page)
==================

=>Asking all category questions in single static form page is not going to work out becoz enduser needs to answer both related and unrelated questions.

Solution:: Ask common /regular question using static form page and based on the data given in the static form page start generatin questions in 2nd dynamic form page , similarly based on the data given to the second form page start generating questions in the 3rd form page.



This app is showing statelesss behaviour of the web application while working dynamic forms..



person_details.html

```html
<h1 style="color:red;text-align: center"> Person details gathering- Form1</h1>

<form action="firsturl" method="POST">
 <table border="1" align="center" bgcolor="cyan">
   <tr>
     <td> name :: </td>
     <td> <input type="text" name="name"> </td>
   </tr>
   <tr>
     <td> age :: </td>
     <td> <input type="text" name="age"> </td>
   </tr>

   <tr>
     <td> address :: </td>
     <td> <input type="text" name="addrs"> </td>
   </tr>

   <tr>
     <td> marital status :: </td>
     <td> <input type="radio" name="ms" value="married">Married
          <input type="radio" name="ms" value="single" checked>Single
     </td>
   </tr>

   <tr>
     <td colspan="2"> <input type="submit" value="continue"></td>
   </tr>
 </table>

</form>
```

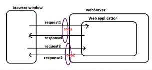
SecondServlet.java

```java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/secondurl")
public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respose content type
        res.setContentType("text/html");
        //read form1/req1 data
        String name=req.getParameter("name");
        String addrs=req.getParameter("addrs");
        String age= req.getParameter("age");
        String ms= req.getParameter("ms");
        //read form2/req2 data
        String f2val1=req.getParameter("f2t1");
        String f2val2=req.getParameter("f2t2");

        pw.println("<h1 style='color:red'> Form1 /request1 data ::"+name+"...."+addrs+"....."+age+"....."+ms+"</h1>");
        pw.println("<h2 style='color:blue'> Form2 /request2 data ::"+f2val1+"...."+f2val2+"</h1>");


        pw.println("<h2 ><a href='person_details.html'> home ::</a></h2");

         //close stream
        pw.close();


    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req, res);
    }//doPost(-,-)

}//class
```

FirstServlet.java

```java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/firsturl")
public class FirstServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respose content type
        res.setContentType("text/html");
        //read form1/req1 data
        String name=req.getParameter("name");
        String addrs=req.getParameter("addrs");
        int age= Integer.parseInt(req.getParameter("age"));
        String ms= req.getParameter("ms");

        //generate form2 dymically based on the marital status
        if(ms.equalsIgnoreCase("married")) {
            pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
            pw.println("<form action='secondurl' method='POST'>");
            pw.println("<table bgcolor='cyan' align='center'>");
            pw.println("<tr><td> spouse name:: </td> <td><input type='text' name='f2t1'></td></tr>");
            pw.println("<tr><td> No.of kids:: </td> <td><input type='text' name='f2t2'></td></tr>");
            pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
            pw.println("</table>");
        }
        else {
            pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
            pw.println("<form action='secondurl' method='POST'>");
            pw.println("<table bgcolor='cyan' align='center'>");
            pw.println("<tr><td> When do u want to marry:: </td> <td><input type='text' name='f2t1'></td></tr>");
            pw.println("<tr><td> why do u want to marry:: </td> <td><input type='text' name='f2t2'></td></tr>");
            pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
            pw.println("</table>");
        }

         //close stream
        pw.close();

    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req, res);
    }//doPost(-,-)

}//class
```

=>Java web applications are stateless by default becoz the protocol http is stateless by defualt i.e we can not use one request data in another request becoz for every request one seperate con will be created b/w Browser and server and that connection will be closed automatically once the response of that request comes back to browser, Due to this the state/data of one request/connection can not be used in another request/connection.



=> Though protocol "http" is stateless(uses seperate connection for every request) we want to see our web application as stateful web application(Client remebering data across the multiple requests) .For this we need to one or another SessionTracking/Session Management Technique.

The following are the session tracking and SessionManagement techniques
a) Hidden Form fields (unofficial)
b) Htp Cookies
c) HttpSession with Cookies
d) Http Session with URL Rewriting

official.. (b, c, d)

=>By Applying these techniques we can make the web application as the stateful web application though the protocol "http" is stateless by default.

=> Stateful web application means the web application remembers the client data across the multiple requests.

Hidden Formfields
=================

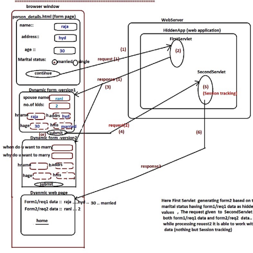
In form page

<input type="hidden" name="h1" value="hello"/>

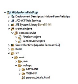
In servlet comp

String s1=req.getParameter("h1"); //gives hello

Example App on Hidden Form fileds based Session Tracking Technique
==



Here First Servlet generating form2 based on the marital status having form1/req1 data as hidden box values , The request given to SecondServlet will read both form1/req1 data and form2/req2 data.. i.e while processing reuest2 it is able to work with request1 data (nothing but Session tracking)



Person_detail.html
==================

```html
<h1 style="color:red;text-align: center"> Person details gathering- Form1</h1>

<form action="firsturl" method="POST">
 <table border="1" align="center" bgcolor="cyan">
   <tr>
     <td> name :: </td>
     <td> <input type="text" name="name"> </td>
   </tr>
   <tr>
     <td> age :: </td>
     <td> <input type="text" name="age"> </td>
   </tr>

   <tr>
     <td> address :: </td>
     <td> <input type="text" name="addrs"> </td>
   </tr>

   <tr>
     <td> marital status :: </td>
     <td> <input type="radio" name="ms" value="married">Married
          <input type="radio" name="ms" value="single" checked>Single
     </td>
   </tr>

   <tr>
     <td colspan="2"> <input type="submit" value="continue"></td>
   </tr>
 </table>

 ..
```

FirstServlet.java
=================

```java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/firsturl")
public class FirstServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respose content type
        res.setContentType("text/html");
        //read form1/req1 data
        String name=req.getParameter("name");
        String addrs=req.getParameter("addrs");
        int age= Integer.parseInt(req.getParameter("age"));
        String ms= req.getParameter("ms");

        //generate form2 dymically based on the marital status
        if(ms.equalsIgnoreCase("married")) {
            pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
            pw.println("<form action='secondurl' method='POST'>");
            pw.println("<table bgcolor='cyan' align='center'>");
            pw.println("<tr><td> spouse name:: </td> <td><input type='text' name='f2t1'></td></tr>");
            pw.println("<tr><td> No.of kids:: </td> <td><input type='text' name='f2t2'></td></tr>");
            pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
            pw.println("<tr><td><input type='hidden' name='hname' value="+name+"/></td></tr>");
            pw.println("<tr><td><input type='hidden' name='haddrs' value="+addrs+"/></td></tr>");
            pw.println("<tr><td><input type='hidden' name='hage' value="+age+"/></td></tr>");
            pw.println("<tr><td><input type='hidden' name='hms' value="+ms+"/></td></tr>");
            pw.println("</table>");
            pw.println("</form>");
        }
        else {
            pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
            pw.println("<form action='secondurl' method='POST'>");
            pw.println("<table bgcolor='cyan' align='center'>");
            pw.println("<tr><td> When do u want to marry:: </td> <td><input type='text' name='f2t1'></td></tr>");
            pw.println("<tr><td> why do u want to marry:: </td> <td><input type='text' name='f2t2'></td></tr>");
            pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
            pw.println("<tr><td><input type='hidden' name='hname' value="+name+"/></td></tr>");
            pw.println("<tr><td><input type='hidden' name='haddrs' value="+addrs+"/></td></tr>");
            pw.println("<tr><td><input type='hidden' name='hage' value="+age+"/></td></tr>");
            pw.println("<tr><td><input type='hidden' name='hms' value="+ms+"/></td></tr>");
            pw.println("</table>");
            pw.println("</form>");
        }

         //close stream
        pw.close();

    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req, res);
    }//doPost(-,-)

}//class
```

SecondServlet.java
==================

```java
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/secondurl")
public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respose content type
        res.setContentType("text/html");
        //read form1/req1 data  form hidden boxes  of form2 (session tracking)
        String name=req.getParameter("hname");
        String addrs=req.getParameter("haddrs");
        String  age=  req.getParameter("hage");
        String ms= req.getParameter("hms");
        //read form2/req2  data
        String f2val1=req.getParameter("f2t1");
        String f2val2=req.getParameter("f2t2");

        pw.println("<h1 style='color:red'> Form1 /request1 data ::"+name+"...."+addrs+"....."+age+"....."+ms+"</h1>");
        pw.println("<h2 style='color:blue'> Form2 /request2 data ::"+f2val1+"...."+f2val2+"</h1>");


        pw.println("<br><bR><h2 ><a href='person_details.html'> home ::</a></h2");

         //close stream
        pw.close();


    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        doGet(req, res);
    }//doPost(-,-)

}//class
```

pros (advnatages) and cons(disadnavages) of Hidden Form fileds technique
==

pros
====
=>This technique work in all web servers and application server (both java and non-java)
=>This technique work in all server side web technologies (both java and non-java)
=>This technique needs basic HTML knowledge

cons
====
=>There is no data secracy becoz we can see hidden boxes data using view --> source option of browser.
=> hidden boxes can only text data (i.e we can not place all kinds of java objects init)
=> The data of hidden boxes travel along with request and response across the multiple requests ..so there will be network traffic b/w browser and webserver..
=> Each dynamic form , we need to add previous forms related data as hidden box values.. which will be burden to the programmers.
=> It is not industry standard technique => It is not official technique to use.

Q) findout the places of real websiste where session tracking is already implemented..

Q) findout the places of real websiste where session tracking is already implemented..

a) Login to Logout in any website
b) NetBanking application
c) Shopping cart app
d) TicketBookign Apps
e) Online examination Apps
f) Online Gaming Apps
and etc..

Dec 21 Session tracking using cookies

Http Cookies

=>Http Cookies are small textual Informations that allocates memory at client side (brower side) remebering Client data across the multiple requests

Two types of Cookies

a) InMemory Cookies /Per SEssion Cookies

[These cookies allocate memory of browser's memory . The RAM memory that is used for executing browser s/w]

=> these cookies do not have expiry time

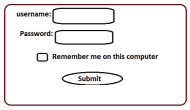
=>These cookies will be destroyed once the browser window is closed

b) Persistent cookies

[These cookies allocate memory in the client machine HardDisk(File system) having expirty time i.e these cookies will not be deleted even browser window is closed]

=> InMemory cookies are suitable for Session tracking where as Persistent cookies are not suitable for the same.

=>Persistence cookies useful for creating remember me option in the web application



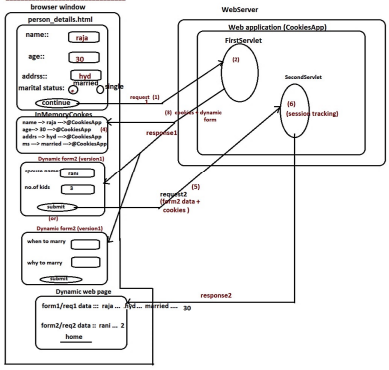
Every Cookie contains following details

=>cookie name (string)
=> cookie value (as text value)
=> expiry time (-1 for InMemory cookies)
=> Domain name (website/ web application name)
and etc..

=>cookies are created at server side in web comps (servlet or jsp comps) and added to the response as the value of response header "set-cookie" , comes to browser and allocates in browser memory (InMemory cookie) or allocates in Client machine file system (Persistence cookies)

=>The cookies available at client side (browser's memory or file system) goes back to web application when the request is given back to same web application as values of req header "cookies"

Http cookies based Session tracking



w.r.t diagram

(2) FirstSErvlet comp reads form1/req1 data and creates InMemory cookies having form1/req1 data and also adds them response.
(3) FirstServlet comp sends response to browser having form2 and cookies holding form1/request1 data
(4) The InMemory cookies that came along with the response allocate memory in browser
(5) form2 submits the reques to SecondServlet comp of same web application.. so the cookies belonging to that web application goes back to web application along with request..
(6) SecondSErvlet comp reads form2 /req2 data directly .. and reads form1/req1 data from cookies i.e SecondSevlet is able to read form1/req1 data while processing form2/req2 {This is nothing but Session tracking}

Cookie API (working with jakarta.servlet.http.Cookie class)

To create cookies

In servlet comp

```
Cookie ck1=new Cookie("ap","amaravathi");
res.addCookie(ck1); // InMamory cookie

Cookie ck2=new Cookie("TN","chennai");
ck2.setMaxAge(1800); //1800 secs as the expirty time
res.addCookie(ck2); // Persistence cookie
```

To modify cookie value

```
ck1.setValue("vizag/kurnool/amaravathi");
ck2.setValue("New Chennai");
```

To delete cookies

=> we can not delete cookies using servlet api ...becoz they allocate memory at client side .. but InMemory cookies will be deleted once the browser window closed where as persistent cookies will be deleted once expirity time is completed

=> Using Browser settings we can delete cookies explicitly..

To read cookies

In Servlet comp

```
Cookies ck[]=req.getCookies();

if(ck!=null){
  for(Cookie c: ck){
    s.o.p(c.getName()+"  "+c.getValue());
  }
}
```

To get domain domain of cookie

```
String d1=ck1.getDomain(); //CookiesApp
String d2=ck2.getDomain(); //CookiesApp
```

To get expiry time of Cookie

```
int time=ck1.getMaxAge(); //gives -1 (InMemory cookie)
int time1=ck2.getMaxAge(); //gives 1800 (Persistent cookie)
```

To set comment to the cookie

```
ck1.setComment("ap's capital");
ck2.setCommnet("TN's capital");
```

To read comments

```
String s1=ck1.getComment();
String s2=ck2.getCommnet();
```

(POC)

FirstServlet.java

```
@WebServlet("/firsturl")
public class FirstServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            //get PrintWriter
            PrintWriter pw=res.getWriter();
            // set response content type
            res.setContentType("text/html");
            //create cookies
            Cookie ck1=new Cookie("ap","amaravathi");
            Cookie ck2=new Cookie("TS","hyd");
            res.addCookie(ck1); res.addCookie(ck2);  //in Memory Cookie
            Cookie ck3=new Cookie("MH","Mumbai");
            Cookie ck4=new Cookie("MP","bhopal");
            ck3.setMaxAge(60);  // 60 secs
            ck4.setMaxAge(120);  //120 secs
            res.addCookie(ck3); res.addCookie(ck4);  //Persistent cookies

            pw.println("<h1 style='color:red;text-align:center'> Cookies are sucessfully created </h1>");

            //close stream
            pw.close();

    }

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
doGet(req,res);
    }

}
```

Secondervlet.java

```
@WebServlet("/secondurl")
public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            //get PrintWriter
            PrintWriter pw=res.getWriter();
            // set response content type
            res.setContentType("text/html");
            //read and display cookies
            Cookie cks[]=req.getCookies();
            pw.println("<table>");
            pw.println("<tr><th> cookie names</th> <th> cookie value </th> </tr>");
            for(Cookie ck:cks) {
                    pw.println("<tr><td>"+ck.getName()+"</td> <td>"+ck.getValue()+"</td></tr>");
            }
            pw.println("</table>");

            //close stream
            pw.close();


    }

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
doGet(req,res);
    }
}//class
```

http://localhost:2020/CookiesApp1/firsturl

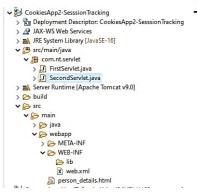
->creates the cookies

http://localhost:2020/CookiesApp1/secondurl
display all the 4 cookies (2+ 2 cookies)

http://localhost:2020/CookiesApp1/secondurl (close browser window and reopen)
gives 2 cookies (persistent cookies)

http://localhost:2020/CookiesApp1/secondurl (after 1 min)
gives 1 cookie

http://localhost:2020/CookiesApp1/secondurl (after 2 min)
gives 0 cookies



Example App on http cookies based Session tracking

form page

Dec 22 Session tracking using cookies

```
<h1 style="color:red;text-align: center"> Person details gathering- Form1</h1>

<form action="firsturl" method="POST">
 <table border="1" align="center"  bgcolor="cyan">
   <tr>
    <td> name :: </td>
    <td> <input type="text" name="name"> </td>
   </tr>
   <tr>
    <td> age :: </td>
    <td> <input type="text" name="age"> </td>
   </tr>

   <tr>
    <td> address :: </td>
    <td> <input type="text" name="addrs"> </td>
   </tr>

   <tr>
    <td> marital status :: </td>
    <td> <input type="radio" name="ms" value="married">Married
         <input type="radio" name="ms" value="single"  checked>Single
    </td>
   </tr>

    <tr>
    <td colspan="2"> <input type="submit"  value="continue"></td>
   </tr>
 </table>

</form>
```

FirstServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/firsturl")
public class FirstServlet extends HttpServlet {

    public  void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respose content type
        res.setContentType("text/html");
        //read form1/req1 data
        String name=req.getParameter("name");
        String addrs=req.getParameter("addrs");
        int  age=  Integer.parseInt(req.getParameter("age"));
        String ms= req.getParameter("ms");
        // create 4 InMemory  Cookies having form/req1 data
        Cookie ck1=new Cookie("name",name);
        Cookie ck2=new Cookie("addrs",addrs);
        Cookie ck3=new Cookie("age",String.valueOf(age));
        Cookie ck4=new Cookie("ms", ms);
        //add the cookies to the response
        res.addCookie(ck1);
        res.addCookie(ck2);
        res.addCookie(ck3);
        res.addCookie(ck4);

        //generate form2 dymically based on the marital status
        if(ms.equalsIgnoreCase("married")) {
            pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
            pw.println("<form action='secondurl' method='POST'>");
            pw.println("<table bgcolor='cyan' align='center'>");
            pw.println("<tr><td> spouse name:: </td> <td><input type='text' name='f2t1'></td></tr>");
            pw.println("<tr><td> No.of kids:: </td> <td><input type='text' name='f2t2'></td></tr>");
            pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
            pw.println("</table>");
            pw.println("</form>");
        }
        else {
            pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
            pw.println("<form action='secondurl' method='POST'>");
            pw.println("<table bgcolor='cyan' align='center'>");
            pw.println("<tr><td> When do u want to marry:: </td> <td><input type='text' name='f2t1'></td></tr>");
            pw.println("<tr><td> why do u want to marry:: </td> <td><input type='text' name='f2t2'></td></tr>");
            pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
            pw.println("</table>");
            pw.println("</form>");
        }

         //close stream
        pw.close();

    }//doGet(-,-)

    public  void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            doGet(req, res);
    }//doPost(-,-)

}//class
```

SecondServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/secondurl")
public class SecondServlet extends HttpServlet {

    public  void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respose content type
        res.setContentType("text/html");
        //read form1/req1 data  form cookies request2 (Session Tracking)
        Cookie cks[]=req.getCookies();
        String name=cks[0].getValue();
        String addrs=cks[1].getValue();
        int age=Integer.parseInt(cks[2].getValue());
        String ms=cks[3].getValue();

        //read form2/req2  data
        String f2val1=req.getParameter("f2t1");
        String f2val2=req.getParameter("f2t2");

        pw.println("<h1 style='color:red'> Form1 /request1 data ::"+name+"...."+addrs+"....."+age+"....."+ms+"</h1>");
        pw.println("<h2 style='color:blue'> Form2 /request2 data ::"+f2val1+"...."+f2val2+"</h1>");


        pw.println("<br><bR><h2 ><a href='person_details.html'> home ::</a></h2");

         //close stream
        pw.close();


    }//doGet(-,-)

    public  void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            doGet(req, res);
    }//doPost(-,-)

}//class
```

Pros and cons of http cookies based SEssion tracking

==========

pros

====

=> they will be work with all web technologies (both java and non-java web technologies)

=> Works in all web servers (both java and non java)

=>allocate memory at client side i.e does not give burden to Server

=> Persistence cookies mataintains the data having expirty time

cons

====

=> we must work with protocol http bbased servlet comp (Generic servlet does not support cookies)

=> we can add only text content to the cookies i.e we can not different types of java objects to it

=> Cookies data can be viewd using browser settings

=> cookies can be deleted in the middle of session , so the session tracking fails

-> Cookies can be restricted coming to browser , Again session tracking fails

=>There is restriction on no.of cookies that can be there at client side

(The browser is expected to support 20 cookies for each Web server,
300 cookies total, and may limit cookie size to 4 KB each)

In Chrome

=========

To view cookies content/remove :: menu button ---> settings ---> search for cookie ---> cookies and site data
-->view all cookies --> remove

To block the cookies :: menu button ---> settings ---> search for cookie ---> cookies and site data
-->Block all the cookies

What is differnce b/w state management and Session Manageneni?

=>In SessionManagent the web application remembers Client Data only during the session across the multiple requests, not after the Session. In session Mgmt the data is always specific to a client (browser)
note:: All Session tracking techniques are there performing SessionManagement becoz the they remember client data only during session .. For example the login credentials will be remembered from login to logout (session)

=>In state managent , the web application remebers the client Data across the multiple requets and the data can be used before the session, after the session and during session.
In state Mgmt the data is not specific to a client (browser)

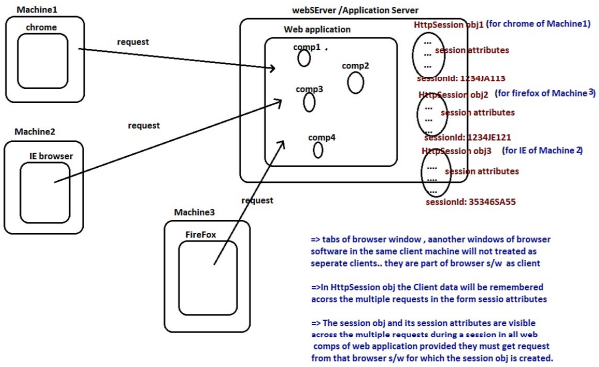
eg:: keeping data as ServletContext attributes , persistent cookies , wrintig each request data to db table as record and etc..

Http Session with Cookies

==========

=>HttpSession object allocates memory in the server and rembers client data across the multiple requests during a session in the form session attributes..

=> HttpSession objects will be created in the server with respect to web application on 1 per browser s/w of a Client machine



=> tabs of browser window , aanother windows of browser software in the same client machine will not treated as seperate clients.. they are part of browser s/w as client

=>In HttpSession obj the Client data will be remembered acorss the multiple requests in the form sessio attributes

=> The session obj and its session attributes are visible across the multiple requests during a session in all web comps of web application provided they must get request from that browser s/w for which the session obj is created.

=>When the session started b/w browser and web application the Session object will be created and when the session is completed the Session object will be destroyed

eg: For Login ---> Session obj is created
For Logout ---> session obj is destroyed

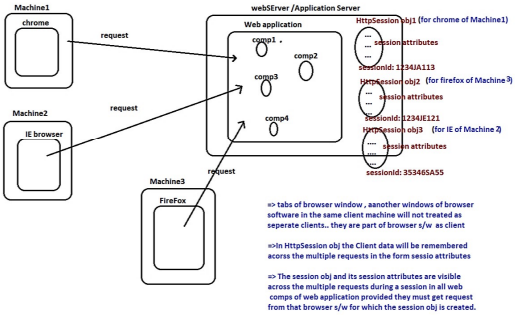
=> The Session id of session object will be send back browser and will be brough back to web application along with the request and response in the form InMemory Cookie , So this technique is called HttpSession with Cookies technique.

Http Session with Cookies

Dec 23 Session tracking using cookies

=>HttpSession object allocates memory in the server and rembers client data across the multiple requests during a session in the form session attributes..

=> HttpSession objects will be created in the server with respect to web application on 1 per browser s/w of a Client machine



=> tabs of browser window , aanother windows of browser software in the same client machine will not treated as seperate clients.. they are part of browser s/w as client

=>In HttpSession obj the Client data will be remembered acorss the multiple requests in the form sessio attributes

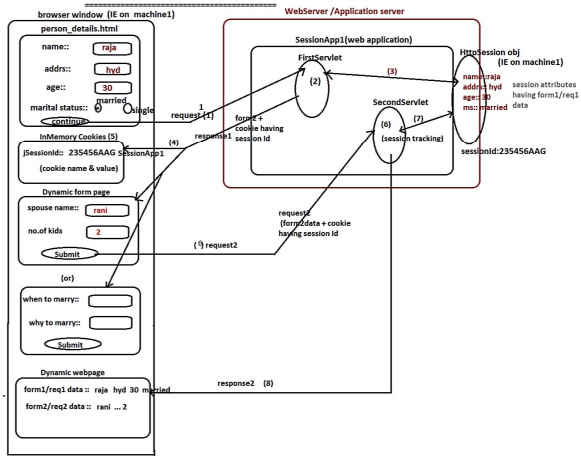
=> The session obj and its session attributes are visible across the multiple requests during a session in all web comps of web application provided they must get request from that browser s/w for which the session obj is created.

=>When the session started b/w browser and web application the Session object will be created and when the session is completed the Session object will be destroyed

eg: For Login —> Session obj is created
For Logout —> session obj is destroyed

=> The Session id of session object will be send back browser and will be brough back to web application along with the request and response in the form InMemory Cookie , So this technique is called HttpSession with Cookies technique.

Session tracking using Http Session with Cookies



w.r.t diagram

1) form1 generates requst1 to FirstServlet
2) FirrstServlet reads form1/req1 data
3) FirstServlet creates HttpSession object on server form current browser s/w of client machine to write form1/req1 data as session attributes and also collects the session id of the session obj and makes it as InMemory Cookie having cookie name JSESSIONID.
4) First Servlet generate dynamic form2 as response having SessionId cookie in the response
5) Since the cookie is InMemory cookie the cookie allocates in Browser's memory
6) Form2 generates the request2 to SecondServlet of same web application.. So the cookie having sessionid belong to the same web application goes along with the request
7) Read form2/req2 data
8) Reads the Session Id from cookie and gets to access to Session object to read session attribute values nothing but form2/request2 data (i.e we are able to use request1 data while processing request2 data --session tracking)

session tracking

9) SecondServlet Generates generate the dynamic web page having form1/req1 data and form2/req2 data..

Session API (Working with jakarta.servlet.HttpSession Interface)

Sesssion object/HttpSession obj is nothing but object of underlying container supplied java class that implements jakarta.servlet.HttpSession(I).

To create or Locate HttpSession obj

```
HttpSession ses=req.getSession(true)
        (or)                                  | both are same
HttpSession ses=req.getSession();
```

=>for the given request from browser s/w of client machine if Session object not avaliable in server then it creates new Session obj on the server for browser s/w otherwise it will acess the existing session obj ..

=> This method makes the current request joing in the already started session if session is already
created for browser (client)
=> This method makes the current request creating new session between browser and web application if session is not already created for browser (client)

=> when this method is called in servlet comp the web container performs
a) checks current request carrying cookie or not if not carrying then it creates new session obj for browser
b) if curreent request is carrying cookie then it checks is there any cookie with the name JSESSIONID if not there then it creates new Session obj.
c) If Cookie with the name JSESSIONID is avaiable then collects the coookie value that session id and checks is threre any session obj having that session id ..if not avaiable then creates session obj
d) if session obj avaiable with the recived session id then it will access to the session obj

Only to Locate Session obj

```
HttpSession ses=req.getSession(false);
```

=>This mehtod get access to session object if it is already avaiable for browser s/w of client machine otherwise returns null
=>This method makes current request joining in the existing sesssion object by accessing session but does not create the new Session obj.

Login form submit Button :: req.getSession(true) or req.getSesssion(false);
InBox request :: req.getSession(false)

To get the SessionId

```
Sting id=req.getId(); //gives the Session id
```

To check whether Session obj is newly created object or not

```
boolean b=ses.isNew(); ->returns true if the current request creates the new session obj
                         otherwise returns false i.e if the current request is accessing the existing session obj
```

To know when the Session obj created

```
long ms=ses.getCreationTime(); //gives session object creation time in ms(milli seconds)
Date d=new Date(ms);
```

The difference in ms b/w jan 1st 1970 00:00 hours (Epoach standard) to the session obj creation date and time

To know when the Session obj is lastly accessed

```
long ms=ses.getLastAccessedTime();
Date d=new Date(ms);
```

To get Access to ServletContext obj

```
ServletContext sc=ses.getServeltContext();
```

To invalidate Session

=> Invalidating the session means making the session object as the inactive object i.e we can not do any operations on the Session object.

a) By calling ses.invalidate() method

b) By closing browser window
(becoz the cookie having session id is InMemory cookie allocating in browser's memory..So once the browser is closed all the InMemory cookies including JSESSION Id will be destroyed.. In this process Session obj will invalidated becoz Session id is avaiable with the browser accessss it)

c) when session's idle timeout period / max inactive interval period is completed.

i) using programatic appproach (java code)

```
ses.setMaxInactiveInterval(1800); //1800 secs --30 min
```

ii) using decl appproach (web.xml code)

```
<session-config>
  <session-timeout>10</session-timeout>
</session-config>
```

10 minutes

Q) If we specify Session's idle time out period in both programmatic and Declarative approach ..can u tell me which time will be taken as final time?

Ans) The idle time out period specified in the Programatic approach(java code in servlet) will take place as the final value becoz the Programatic code executes after the declarative code (web.xml code executes during the deployment where as servlet code executes when given request).

Dec 24 Session tracking using cookies

=>In HttpSession object , we keep the data in the form of Session attributes..

=>HttpSession obj and its session attributes are visible in multiple web comps of web application across the multiple requests provided when they get request from that browser s/w client machine for whom HttpSession obj and attributes are created.

To create Session attribute
==========

```
ses.setAttribute("username","raja");
```

To modify Session attribute values
==========

```
ses.setAttribute("username","king");
```

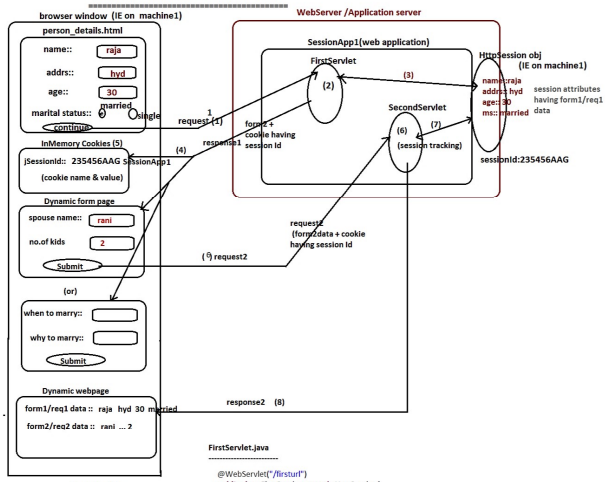
To read Session attribute values
==========

```
String value=(String)ses.getAttribute("username");
```

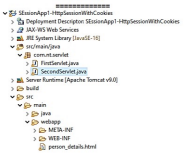
To remove Session attribute
==========

```
ses.removeAttribute("username");
```

Session tracking using Http Session with Cookies
==========



Example App
==========



FirstServlet.java

```
@WebServlet("/firsturl")
public class FirstServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
            PrintWriter pw=res.getWriter();
            //set respose content type
            res.setContentType("text/html");
            //read form1/req1 data
            String name=req.getParameter("name");
            String addrs=req.getParameter("addrs");
            int age= Integer.parseInt(req.getParameter("age"));
            String ms= req.getParameter("ms");

            //create HttpSession object
            HttpSession ses=req.getSession(true);
            ses.setAttribute("name", name);
            ses.setAttribute("addrs", addrs);
            ses.setAttribute("age", age);
            ses.setAttribute("ms", ms);
             //Collecting the Session Id of Session obj , creating InMemory Cookie having
             //the SessionId and sending that InMemory cookie to browser as response will be
             //done automatically by the ServletContaienr the moment HttpSession object is created.


             //generate form2 dymically based on the marital status
             if(ms.equalsIgnoreCase("married")) {
                  pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
                  pw.println("<form action='secondurl' method='POST'>");
                  pw.println("<table bgcolor='cyan' align='center'>");
                  pw.println("<tr><td> spouse name:: </td> <td><input type='text' name='f2t1'></td></tr>");
                  pw.println("<tr><td> No.of kids:: </td> <td><input type='text' name='f2t2'></td></tr>");
                  pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
                  pw.println("</table>");
                  pw.println("</form>");
             }
             else {
                  pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
                  pw.println("<form action='secondurl' method='POST'>");
                  pw.println("<table bgcolor='cyan' align='center'>");
                  pw.println("<tr><td> When do u want to marry:: </td> <td><input type='text' name='f2t1'></td></tr>");
                  pw.println("<tr><td> why do u want to marry:: </td> <td><input type='text' name='f2t2'></td></tr>");
                  pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
                  pw.println("</table>");
                  pw.println("</form>");
             }

             pw.println("<br><br><b> Session Id of the Session obj::"+ses.getId()+"</b>");

              //close stream
             pw.close();

        }//doGet(-,-)

        public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
              doGet(req, res);
        }//doPost(-,-)

   }//class
```

SecondServlet.java
==========

```
@WebServlet("/secondurl")
public class SecondServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
            PrintWriter pw=res.getWriter();
            //set respose content type
            res.setContentType("text/html");
            //read form2/req2 data
  String f2val1=req.getParameter("f2t1");
  String f2val2=req.getParameter("f2t2");
            //get Access to the Session obj (this method internally reads InMemory Cookie vlaue
                        //     to get SessionId and uses that SessionId to access the SEssion obj
            HttpSession ses=req.getSession(false);
            //read form1/req1 data from SEssion obj as Session attribute values
            String name=(String)ses.getAttribute("name");
            String addrs=(String)ses.getAttribute("addrs");
            int age=(Integer)ses.getAttribute("age");
            String ms=(String)ses.getAttribute("ms");

            pw.println("<h1 style='color:red'> Form1 /request1 data ::"+name+"....."+addrs+"....."+age+"....."+ms+"</h1>");
            pw.println("<h2 style='color:blue'> Form2 /request2 data ::"+f2val1+"...."+f2val2+"</h1>");


            pw.println("<br><br><h2 ><a href='person_details.html'> home ::</a></h2>");

            pw.println("<br><br><b> Session Id of the Session obj::"+ses.getId()+"</b>");

            //invalid the Sessin
            ses.invalidate();

             //close stream
            pw.close();
    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
            doGet(req, res);
    }//doPost(-,-)


}//class
```

Person_detail.shtml

```
<h1 style="color:red;text-align: center"> Person details gathering- Form1</h1>

<form action="firsturl" method="POST">
 <table border="1" align="center" bgcolor="cyan">
   <tr>
     <td> name :: </td>
     <td> <input type="text" name="name"> </td>
   </tr>
   <tr>
     <td> age :: </td>
     <td> <input type="text" name="age"> </td>
   </tr>

   <tr>
     <td> address :: </td>
     <td> <input type="text" name="addrs"> </td>
   </tr>

   <tr>
     <td> marital status :: </td>
     <td> <input type="radio" name="ms" value="married">Married
          <input type="radio" name="ms" value="single" checked>Single
      </td>
   </tr>

    <tr>
     <td colspan="2"> <input type="submit"  value="continue"></td>
    </tr>
 </table>

</form>
```

pros of HttpSession with Cookie technique

a) This technique works with all web technologies including java

b) this technique works with all web server (both java and non-java)

c) session attributes can have both text data and java objects as values

d) Session creation and SEssion invalidation/Ending will be there in our control completely

e) we can sepecify session's idletimeout period .. to invalidate the session ..if it is in idle mode for long time.

f) Since client data is stored in the sever as session attributes of HttpSEssion obj, we can say client data is not travelling over the network across the multiple requests .. This indicates less network traffic

g) Creating InMemory Cookie having SessionId of the Session obj and sending it browser by adding response and also reading session id from the cookies request will be taken care by Servletcontainer .. This reduces the burden on the Programmer.

cons
====

=> We must work with the "protocol" http based Servlet comp like HttpServlet in order to work with this technique..

=> HttpSession objects allocate memory in the server.. So they improve burden on the server
=> If cookies are deleted in the middle of the session or cookies are restricted/blocked to come to browser then the Session tracking will fail..

What happens If we close and open browser having same url in the middle of the session ?

What happens If we restart the server in the middle of the session ?

Dec 26 Session tracking using URL Rewriting

What happens if we close and open browser having same url in the middle of the session ?

=> when we close the browser window .. we will loose the session Id becoz the InMemory that contains the Session Id will be destroyed.. So when we open the new browser s/w .. the new SessionId will be created.. for the new Session that is started.. i.e the old session will not be continued.

What happens if we restart the server in the middle of the session ?

=> In Most of the server the session will be not continued becoz when the server is down , all objects will be destroyed including Session objects.. So after restarting the server , the session will not be continued

=> Only In tomcat server , The Session persistence takes place across the multiple restarts by internally using Serialization and DeSerilziation concepts..

=>As part of Serilziation , they write Session objs their data including Session ids to files while server is shutting down.

=>As part of DeSerilziation , they read data from files and constructs the objects like Session objs having data with session ids..

=>To disable this Feature of Session Persistence across the multipe restarts we need to use peform the following operations..

Project explorer --> servers --> Tomcat Server --> context.xml -->

uncomment the <Manager pathname="" />

==========
HttpSession with URL Rewriting
==========

=>In HttpSession with cookies Session tracking tehnique .. the Session will not continued or the session tracking does not take place if cookies are deleted in the middle of the session or cookies are restricted to come to browser along with the response .. So do not use cookies to send the session id to browser from web application and vice-versa.

=>Instread cookies use other alternate process like append Session id to the URL that goes to browser from web application along with the response and comes back to web application from browser along with the request.. this process is called URL Rewriting (Given URL is rewritten by appending the session id)

=>The href ulrs of <a> tags, action urls of <form> tags that are crated ins servlet comp any dynamic hyperlinks or dynamic forms go to browser along the response and comes back to web application along with request .. On these urls we can append session id by using URL Rewriting technique.

```
pw.println("<form action='secondurl' method='POST'>");
```

On this kind of URLS we need to do URL Rewriting

```
pw.println("<form action='secondurl;jSessionId=54556AA66' method='POST'>")
  ...
  ...
pw.println("</form>");
```

url rewriting can be done using res.encodeURL(-) method
res.encodeURL("secondurl"); ->gives secondurl;jSessionId=545662AA54

```
pw.println("<form action='"+res.encodeURL("secondurl")+"  method='POST'>")
```

example App
==========

person_details.html

```
<h1 style="color:red;text-align: center"> Person details gathering- Form1</h1>

<form action="/firsturl" method="POST">
 <table border="1" align="center" bgcolor="cyan">
   <tr>
     <td> name :: </td>
     <td> <input type="text" name="name"> </td>
   </tr>
   <tr>
     <td> age :: </td>
     <td> <input type="text" name="age"> </td>
   </tr>

   <tr>
     <td> address :: </td>
     <td> <input type="text" name="addrs"> </td>
   </tr>

   <tr>
     <td> marital status :: </td>
     <td> <input type="radio" name="ms" value="married">Married
          <input type="radio" name="ms" value="single" checked>Single
     </td>
   </tr>

   <tr>
     <td colspan="2"> <input type="submit"  value="continue"></td>
   </tr>
 </table>
```

FirstServle.tjava

```
@WebServlet("/firsturl")
public class FirstServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
  // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        //read form1/req1 data
        String name=req.getParameter("name");
        String addrs=req.getParameter("addrs");
        int  age=  Integer.parseInt(req.getParameter("age"));
        String ms= req.getParameter("ms");

        //create HttpSession object
        HttpSession ses=req.getSession(true);
        ses.setAttribute("name", name);
        ses.setAttribute("addrs", addrs);
        ses.setAttribute("age", age);
        ses.setAttribute("ms", ms);
        //Collecting the Session Id of Session obj , creating InMemory Cookie having
        //the SessionId and sending that InMemory cookie to browser as response will be
        //done automatically by the ServletContaienr the moment HttpSession object is created.


        //generate form2 dymically based on the marital status
        if(ms.equalsIgnoreCase("married")) {
              pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
              pw.println("<form action="+res.encodeURL("secondurl")+" method='POST'>");
              pw.println("<table bgcolor='cyan' align='center'>");
              pw.println("<tr><td> spouse name:: </td> <td><input type='text' name='f2t1'></td></tr>");
              pw.println("<tr><td> No.of kids:: </td> <td><input type='text' name='f2t2'></td></tr>");
              pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
              pw.println("</table>");
              pw.println("</form>");
        }
        else {
              pw.println("<h1 style='color:red;text-align:center'> person Information gathering -Form2 </h1>");
              pw.println("<form action="+res.encodeURL("secondurl")+" method='POST'>");
              pw.println("<table bgcolor='cyan' align='center'>");
              pw.println("<tr><td> When do u want to marry:: </td> <td><input type='text' name='f2t1'></td></tr>");
              pw.println("<tr><td> why do u want to marry:: </td> <td><input type='text' name='f2t2'></td></tr>");
              pw.println("<tr><td colspan='2'><input type='submit' value='submit'> </td></tr>");
              pw.println("</table>");
              pw.println("</form>");
        }

        pw.println("<br><br><b> Session Id of the Session obj::"+ses.getId()+"</b>");

        //close stream
        pw.close();

    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
          doGet(req, res);
    }//doPost(-,-)

}//class
```

SEcondrSErvlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


@WebServlet("/secondurl")
public class SecondServlet extends HttpServlet {

      public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
   // get PrintWrtier
        PrintWriter pw=res.getWriter();
        //set respone content type
        res.setContentType("text/html");
        //read form2/req2 data
   String f2val1=req.getParameter("f2t1");
   String f2val2=req.getParameter("f2t2");
        //get Access to the Session obj (this method internally reads InMemory Cookie vlaue
              //    to get SessionId and uses that SessionId to access the SEssion obj
        HttpSession ses=req.getSession(false);
        //read form1/req1 data from SEssion obj as Session attribute values
        String name=(String)ses.getAttribute("name");
        String addrs=(String)ses.getAttribute("addrs");
        int  age=(Integer)ses.getAttribute("age");
        String ms=(String)ses.getAttribute("ms");

        pw.println("<h1 style='color:red'> Form1 /request1 data ::"+name+"...."+addrs+"....."+age+"....."+ms+"</h1>");
        pw.println("<h2 style='color:blue'> Form2 /request2 data ::"+f2val1+"...."+f2val2+"</h1>");


        pw.println("<br><bR><h2 ><a href='person_details.html'> home ::</a></h2");

        pw.println("<br><br><b> Session Id of the Session obj::"+ses.getId()+"</b>");

        //invalid the Sessin
        ses.invalidate();

        //close stream
        pw.close();
    }//doGet(-,-)

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
          doGet(req, res);
    }//doPost(-,-)


}//class
```

Advantages and DisAdvantages of this technique

Same as HttpSession with cookies having the following addtional advantages and disadvantages

addtional advantage

The session tracking will not be stoppied even though cookies are restricted to come browser or cookies are deleted in the int middle of the Session

addtional disadvanage
---

we need to perform URL rewriting on the every URL(href,action urls) that goes to browser along with response and comes back to web application along with the request.

What is SessionTracking tehnique that u have used in your Project?

=> if the customer base is more and web application is getting huge no.of requsts the prefer using http cookies becoz they allocate memory at client side .. not distrubing server side memory
eg:: google.com, flipkart .com , amazon.com and etc..

=> if the customer base is less and web application is not getting huge no.of requsts the prefer using HttpSession with URL writing ..
eg:: Banking apps, financial services App,College Apps, University Apps and etc..

=> If needed we can use multiple Session tracking techniques in a Single App
For login activies , use HttpSession with URL rewriting
For shopping cart activities , use Http cookies

Servlet Listeners /Event Handling in Servlets

Dec 28 Servlet Listeners

=> The action perform on the object / comp is called event.
=> The process of executing some logic or task for action(Event) that is raised is called Event Handling.
=> Event Listeners are required for Event handling.. becoz they provide event handling methods..
=> When the event is raised the event handling methods will be called automatically..

For Event handling we need 4 details (in AWT/Swing env...)
a) Source comp (eg: Button)
b) Event (eg: ActionEvent(c) -- clicking on the Button)
c) Event Listener (eg: ActionListener(I) -->By implemting this we can write event handling logic)
d) Event Handling method (eg: public void actionPerformed(ActionEvent ae){.....})

=> we can perform event handling on request, response, ServletContext objs to keep track of when those objects are created and destroyed and accordingly we can complete some tasks..

=>Event handling on request obj helps us to know when request object is created and destroyed..
The difference b/w both timings can be taken as time required to prcoess each request

=>Event handling on ServletContext obj helps us to know when the servletContext object is created and destroyed.. The difference b/w both timings can be taken as the time of executing web application continuosly.. (suitable for web site maintainance)

=>Event handling on Session obj helps us to know when the Session object is created and destroyed..
The difference b/w both timings can be taken as time taken by user to the session (session duration of a user)

note:: Event Hanlding in servlet programming is not given for regular request processing and response generation.. It is puerely there to enable mononitoring on the web applications..

To perform event handling in servlet programming the 4 details are required

source obj	Event clss(c)	Event Listener(I)	Event handling methods
request obj	SevletRequestEvent	ServletRequestListner	void requestDestroyed(ServletRequestEvent sre) Receives notification that a ServletRequest is about to go out of scope of the web application. void requestInitialized(ServletRequestEvent sre) Receives notification that a ServletRequest is about to come into scope of the web application.
session obj	HttpSessionEvent	HttpSessionListener	void sessionCreated(HttpSessionEvent se) Receives notification that a session has been created. void sessionDestroyed(HttpSessionEvent se) Receives notification that a session is about to be invalidated.
ServletContext obj	ServletContextEvent	ServletContextListener	void contextDestroyed(ServletContextEvent sce) Receives notification that the ServletContext is about to be shut down. void contextInitialized(ServletContextEvent sce) Receives notification that the web application initialization process is starting.

Every Servlet Listner is a impl class of XxxListener(I) and should cfg either using xml driven cfgs or using annotation driven cfgs..

=> using xml driven cfgs :: <listener> , <listener-class> tags
=> using annotation cfgs :: @WebListener (class level)

note:: ServletListeners are very useful to enable monitoring the apps..

Upto Servlet 3.x the XxxListener interfaces are given as the normal java interfaces where as the from servlet 4.x XxxListener interfaces are given interfaces with default methods.

=>Each ServletListner class obj will be created during server startup or duing the deployment of the web application.

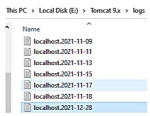
The order of creating objects during the deployment of web application
a)Listener classes objs (if avaiable)
b) SErvletContext obj
c) Servlet Filter class objs (if avaiable)
d) FilterConfig objs(of filters are avaible)
e) <load-on-startup> enabled Servlet class objs
f) ServletCofig objs for the <load-on-startup> enabled Servlet classes

example App

step1) keep any example App ready

step2) Develop and configure the ServletListener..

```
//Request Listener
package com.nt.listener;

import javax.servlet.ServletContext;
import javax.servlet.ServletRequest;
import javax.servlet.ServletRequestEvent;
import javax.servlet.ServletRequestListener;
import javax.servlet.annotation.WebListener;
import javax.servlet.http.HttpServletRequest;

@WebListener
public class RequestProcessingAnalyzer implements ServletRequestListener {
  privete long start,end;

    @Override
    public void requestInitialized(ServletRequestEvent sre) {
    start=System.currentTimeMillis();
    }

    @Override
    public void requestDestroyed(ServletRequestEvent sre) {
      end=System.currentTimeMillis();
      //get request object
      ServletRequest req=sre.getServletRequest();
      // get ServletContext obj
      ServletContext sc=sre.getServletContext();
      System.out.println(((HttpServletRequest)req).getRequestURL()+" has taken "+(end-start)+" ms");  //writes to server console
      //write to log file
      sc.log(((HttpServletRequest)req).getRequestURL()+" has taken "+(end-start)+" ms");
    }                                                    //writes the current day's log file of Tomcat server. which will be ceated on 1 per day basis
}
```




```
//ServletContext Listener
@WebListener
public class WebapplicationMonitoringAnalyzer implements ServletContextListener {
    private long start,end;


    @Override
    public void contextInitialized(ServletContextEvent sce) {
            start=System.currentTimeMillis();
        //get ServletContext obj
        ServletContext sc=sce.getServletContext();
        sc.log("Web application is deployed/started at:"+new Date());
    }


    @Override
    public void contextDestroyed(ServletContextEvent sce) {
            end=System.currentTimeMillis();
        //get ServletContext obj
            ServletContext sc=sce.getServletContext();
            sc.log("Web application is stopped/undeployed/reloaded at:"+new Date()+"  Running  duration::"+(end-start)+" ms");


    }

}
```

```
//SessionListner
================

  import java.util.Date;

  import javax.servlet.ServletContext;
  import javax.servlet.annotation.WebListener;
  import javax.servlet.http.HttpSessionEvent;
  import javax.servlet.http.HttpSessionListener;

  @WebListener
  public class SessionDurationAnalyzer implements HttpSessionListener {
      private long start,end;

      @Override
      public void sessionCreated(HttpSessionEvent se) {
          start=System.currentTimeMillis();
          //get ServletContext obj
          ServletContext sc=se.getSession().getServletContext();
          sc.log("Session started at:"+new Date());
      }

      @Override
      public void sessionDestroyed(HttpSessionEvent se) {
          end=System.currentTimeMillis();
              //get ServletContext obj
              ServletContext sc=se.getSession().getServletContext();
              sc.log("Session ended at::"+new Date()+" Session duration is ::"+(end-start)+" ms");
```