http://tomcat.apache.org/

http://localhost:9090

D:\Developmentsoftwares\Tomcat9\webapps

in the above folder only we have to develop

we have to develop all web applications

1st - create app1 folder (also called project folder)

- we can develop html,css,javascript files in app1 folder

- even jsp file also in the same location

- develop test1.html in the app1

- http://localhost:9090/app1/test1.html

-how to develop servlet

we require additional directory structure.

- inside app1 create onother folder called WEB-INF every character in the upper case

- in the WEB-INF folder create a classes folder this should be same name and create

src folder this can be with any name its for java files.

-create deployment descriptor its XML file in the WEB-INF level

- write java program

D:\Developmentsoftwares\Tomcat9\webapps\app1\WEB-INF\src>javac -d ../classes com/aptech/\*.java

-we need describe in the deployment descriptor

-after web.xml develoment restart the server

http://localhost:9090/app1/hello

- to identify the mistake in web.xml just press cntr + b it will show full structure if it is proper

http://localhost:9090/app1/s2

http://localhost:9090/app1/Servlet2

http://localhost:9090/app1/xyz

http://localhost:9090/app1/abc

-supplying values through query string.

-after ? any no. of input parameters.

- '&' is the seperator between the inputs

http://localhost:9090/app1/abc?param1="Hello"&firstName="vijay"

- we can make a call to the servlet from html also

- we dont need to restart the server if we just develop and call servlet from html.

http://localhost:9090/app1/test2.html

- all html files should be in the project folder.

http://localhost:9090/app1/test3.html

- use PrintWriter to write back anything to the browser through SOP we cant

- to find out where this messages are printing come to

tomcat9->logs->

we can see the stdouts day wise it is creating

- tomcat is a server that only requires class files not the java files.

-------------------------------------------

http://localhost:9090/app2/test1.html

-without developing servlet, in the html if you give and submit you get 404 error

-after html,css,javascript client side dev is done

---------------------WAR Dev-------------------------------------

1. JAR is nothing but zipped format of java.

2. In java to develop a kind of zip, we can use a JAR.

3. Java archieve is nothing but zipped format in java.

4. For core java appliations we will be using jar

once the core java appliation got over, then we take that core java

application content and making a jar file, and that jar file is sending

to the customer, customer will make use of the jar file, we can run jar

file using java command.

5. in case of web application, in place of jar we will be using

WAR(WebArchive).

That WAR file we can run in any J2EE complient server.

it can be tomcat, jboss, web logic, web spear etc.

6. WAR is also called as portable format of the web application.

7. jar cf app5.war \*

i,e create file name, complete content of the current directory,

includin the sub directories, extension .war can be aslo .jar

8. deploy that in webapps folder.

9. then restart the tom cat server, it will be creating one folder

you get complete content of war file.

10. myapp>jar cf D:\Developmentsoftwares\Tomcat9\webapps\app6.war \*

this also possible, absolute path

------------------development with eclipse---------------

1.we need javaee eclipse for this

https://www.eclipse.org/downloads/packages/

2. we need j2ee edition

3. we can develop dynamic web application

4. before creating dynamic web app check if tom cat running outside or not

5. create a new dynamic web project

- eclipse doesnt have any target runtime environments- no tomcat,jre, database

- choose run time envronemnt- tomcat 9

- if you dont get those lists then its not jee eclipse.

-choose the folder name of tomcat installation

-select dynamic web module version to 2.5

- till now we have developed module version of 2.5

-eclipse is using different directory structure for dynamic web application

its eclipse specific,

- every java program(any java pgm, servlet also)

you should develop inside the java resources->src folder

-under web content only we should develop, html, css, js, image, jsp files

-inside webcontent->web-inf->lib we need to place all third party

jar files, we dont update build-path.

- inside eclipse build path is only for core java projects

- for all web apps we need to copy explicitely jar files into the lib

folder.

-in the web-inf we have web.xml by default design view click source

-diplay-name tag is providing a name to display

-welcome-file-list - we are getting a readymade web.xml file.

-this directory structure is specific to eclipse not universal.

- previous development without eclipse, that is universal directory

structure for web app development.

-every j2ee complient servers expecting the directory structure like

this not eclipse structure.

-develop html file in web-content folder

- just run with some content of html directly, no need of

configuring tomcat once again.

-html, js, css, image, and jsp should be inside the web-content folder.

-now develop simple servlet inside servlet

Servlet1 with package com.aptech

- click next it shows url mapping as /Servlet1

-eclipse chooses class name itself as URL pattern

-then uncheck dopost and doget. check service

- in the web.xml you will get optional tags

<description></description>

<display-name>Servlet1</display-name>

-just keep these two statements inside the service method

PrintWriter out = response.getWriter();

out.println("Hello to all from Servlet1");

-just run. no need of restarting the server.

-we dont require to mention the port number to call servlet in browser.

just run

- develop Servlet2 and html file and call the servelet from html

just run the html file and click the link

- develop another html file with simple form copy the action value and

develop Servlet3

- not url-mapping should not be the default one select edit and

paste that url-pattern

-override(check) only service method and uncheck doget, dopost

-come to html and run the file, by restarting the tomcat

you get empty form with submit just click submit you get the servlet call o/p

-now we will develop the servlet which interacts with the db

-creating another project

-target runtime you dont have to configure once again

- target runtime will be configured only once in the workspace

- dynamic web module version keep 2.5 only

- copy ojdbc5 jar file from the oracle installation. Keep that in the lib folder.

----------welcome file list---------------

1. When we are making call to our application

Without specifying any resource name, then welcome file will be displayed

1. <http://loacalhost:9090/app2/>

if you just give this in the external address bar you will get 404 Not found error.

1. By default eclipse provides these many file lists

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

<welcome-file>default.html</welcome-file>

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

1. You can develop any of the file ex- index.html, index.htm, or index.sp etc
2. Any file that got configured in the web.xml we can develop.
3. If first welcome file is not available then second then third will be consifered from top to bottom it will consider.
4. Try another file default.jsp
5. URL - <http://localhost:9090/app5/>
6. Now delete index.html and delete now you will ge the default.jsp
7. Welcome file list is not mandatory we can remove this
8. After modifying the web.xml file we need to restart the server.
9. Now come to external browser call the app5/ now you get 404 error. Bcz of no welcome file list.
10. Now come to eclipse and press cntrl + z you will get the removed index.html file. Then come to the extr browser and refresh you get the content from index.html bcz
11. Got to the servers in the eclipse project explorer and expand tomcat folder and open the web.xml file then press cntr + f and type welcome-file

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>index.htm</welcome-file>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

In this list there is no default.jsp but index.html is there

* These welcome-file-list are global
* These file will be available for every applications whichever running under tomcat.
* For all the application there is application specific web.xml is there and for the applications in the tomcat server one global web.xml is there.
* This configuration available to every application.
* If global and local bother welcome file are there then local will be given the more preference.
* We can also keep our own file in our local app web.xml welcome file list. Keep test1.html
* First restart the server and refresh the browser.

1. We can explicitly restart the server in eclipse

Go to window - show view – servers

Right click on server and restart

1. Develop another project with app6

Develop .htm file in the web content

1. select app6 right click run as run on server.

-----------------getParameter()----------------

1. various ways of reading form data (customer data) inside the servlet.
2. Create one new app7 and create reg1.html
3. http://localhost:9090/app6/reg1?firstName=vijay&lastName=kumar&age=22&address=btm&gender=male&skill=c&skill=cpp&skill=java&education=b

afer ? is query string

1. All skills are going to servlet but only first skill is reading
2. getParameter will be reading only first skill means only reading single value not all the values

in the request object there are several method to read the form data

request

-------

Public String getParamer(String name)//returning name single value

Public String[] getParameterValues(String name)// takes name returns all the values

Public String getQueryString()

Public Map getParameterMap()

Public Enumeration getParameterNames()

-------------------Input to servlets----------------

----reading the data from the browser-------------

1. form data

- String[] firstName = request.getParameterValues("firstName"); this method return the string

array

- make changes in the reg2.html

now its multiselection drop down

<select name="education" multiple>

- if in the form method= 'post' then query string will not be generating, returns null

-query string will not be generarating while form is submitting through post will get NPE 500

error internal server error

------constants--------------

1.not only form data we can read constants also

2. any number of init param can be cofigured for the same servlet.

3. <init-param>

<param-name>p1</param-name>

<param-value>200</param-value>

</init-param>

<init-param>

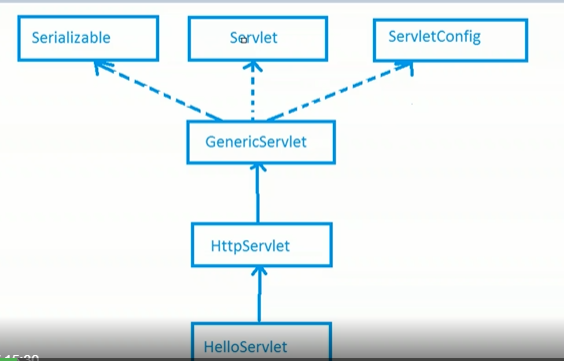
<param-name>country</param-name>

<param-value>India</param-value>

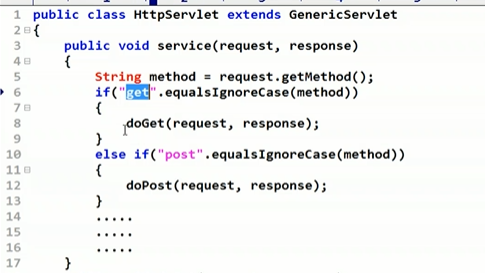
</init-param>

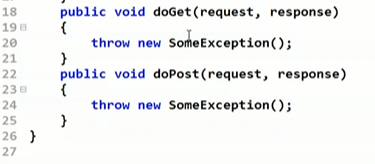
These are specific to one servlet not for other.

Architecture

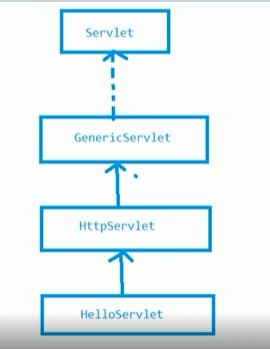


Skeleton of HttpServlet





-------------Init Methods-----------------



-Servlet is an interface

-Generic servlet is an abstract class

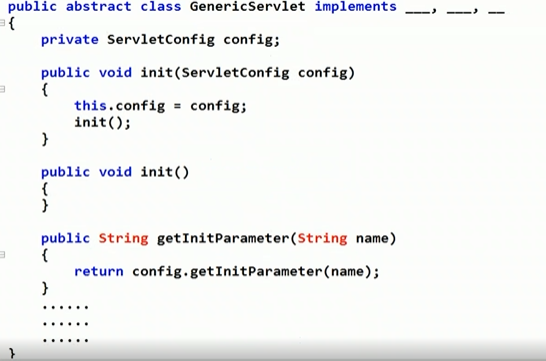
-Http servlet is also an abstract class

-Then we are developing our HelloServlet

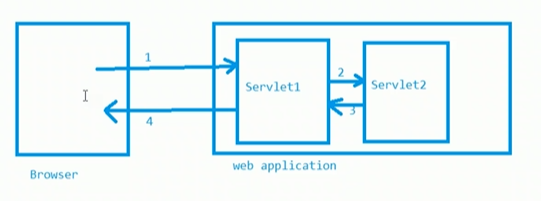
-Servlet contains life-cycle methods

this is how init method got implemented.

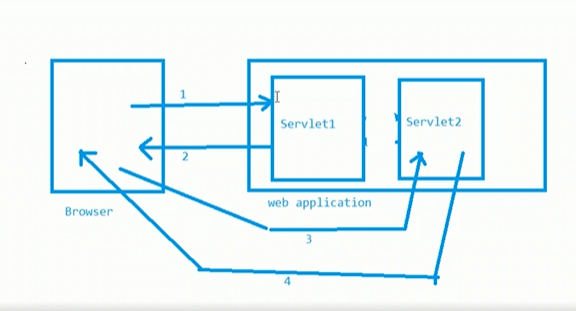
GenericServlet implemts Servlet,ServletConfig,Serializable



--------------RequestDispatcher-------------



-----sendRedirect----------



Because of response.sendRedirect("Servlet2");