

>Server

Server is a software which manages all the resources along with which process the client request and serve the client request.

>There are different types of Servers present namely

- 1.Database Server
- 2.Application Server
- 3.Web Server

1.Database Server

Database Server is used to deal with the data.

Ex:Oracle,MySQL,MS-SQL,Derby,Mongo DB,Sybase,etc.

2.Application Server:

Application Server is used to execute a dynamic application or a realtime application

Ex JBOSS, JBM Webserver,
OracleWebLogic,etc.

Dynamic Application/Realtime Application: An application which performs all the 3 different types of logics such as Presentation Logic ,Persistence Logic and Business Logic is known as Dynamic Application or a realtime application.
Ex:FaceBook,Whatsapp,etc.

3.Web Server:

Web Server is used to execute only web applications.

Ex Apache-Tomcat Server and OracleGlassFish.

>Deployment

Making all the resources available to the server is known as deployment. There are 2 different types of deployment present

namely

Apache-Tomcat Server comes with 2 different variants

1.zip

2.exp

>Port Number

Port Number is the one which helps us to get connected to a particular Server.

Default Port Number for Apache-Tomcat Server is--8080

Basic Steps to be followed in Installation of Apache-Tomcat Server

Port Number

Port Number is the one which helps us to

get connected to a particular Server.

Default Port Number for Apache-Tomcat Server is----8080

Basic Steps to be followed in Installation of Apache-Tomcat Server

There are 3 different specifications for Apache-Tomcat Server

A)JAVA_HOME

Open C-Drive,Program files and open Java and JDK folder and copy the entire path which includes all the folders related to jdk into the Environment Variables

b) CATALINA HOME:

Open Apache-Tomcat server folder from the respective directory and copy the entire path which includes allthe folders retated to Apache-Tomcat Server

C)Pathi

Open the Apache-Tomcat Server folder from the respective directory and copy the entire path which inciudes bin foider of Apache-Tomcat Server

D) JRE HOME(Optional):

Open C-Drive, Program Files and open Java and IRE Folder and copy the entire

path which includes all the folders related to JRE

To start the Apache-Tomcat Server:

Go to the respective directory where the apache-Tomcat server folder is present "- open bin folder -double click on startup(Windows Batch File)

To see Apache-Tomcat Server Page

Go to any of the browser and type below url

<http://ocalhost:8080>

How to change the Port Number:

Port Number for Apache- Tomcat Server is 8080

Which can be changed

Goto the Apache-Tomcat Server folder --- open conf folder--right click on server.xml-- open with Note pad-Hit Cntl+F(Find)- type (conn)- and hit on FindNext for exactly 8 times ---change the port number --8080 to 8050 ---save

Go to bin folder of Apache-Tomcat server--- start the server--

Goto any of the browser - type --http:// localhost:8050

Folders of Apache-Tomcat Server

There are different folders of Apache-Tomcat are present namely

1.bin

2.conf

3,lib

4.logs

5.webapps

6.work

1 bin

it contains a set of Startup and Shutdown batch files which are used to start and stop the Apache-Tomcat Server,

2. cont

it contains set of Configurations with respect to Apache-Tomcat Server

3. lib

it contains set of libraries in the form of jar which is used to perform some additional

functionalities.

Ex:serviet-api.jar

4 logs

it is used to store all the logs messages displayed on Server console. Since server console has limited memory.

5 webapps

it is used to deploy all the web-applications on to apache-tomcat server.

JEE Container

"Application Server is used to execute both web and enterprise applications(intern we call it as JEE Application).

To run any JEE application on a server ,the server must contain a JEE container in it.

"JEE Container is an engine which is used to manage all the JEE components such as Serviet,JSPEJB,etc.

"Server basically performs 2 important tasks namely.

- A) it manages all the resource.
- B) Provides runtime environment

"Web Server is used to execute only web applications

"There are 3 different types of logics associated with dynamic application

which are as follows

1) Presentation Logic:

it is used to present the contents onto an application

Technologies Involved

Eg) HTML,CSS,JS,JSP, PHP.etc

2) Persistence Logic.

Persist means to store , Persistence Logic is used to persist data into the persistence systems(DataBase),

Persistence to store data in the databases.

Technologies Involved

Eg) JDBC SQL,Hibernate,etc.

3) Business Logic:

It performs the core functionality that is some set of calculation and validation operation on an application

Technologies Involved:

Servlet, Spring, etc

Steps to create a basic Web Project(External Way of Execution)

"Click on Open perspective and select JEE as a Perspective.

*Open Navigator Mode

Right Click within the navigator and create a new Dynamic Web Project

"Generate web.xml. (Right click on Project and select Java EE Tools and click on Generate Deployment Descriptor Stub)- inside WEB-INF folder.

There are 3 different criteria present for any resource in general

1)Create Resource: Create a HTML Resource with any name which will be generated in WebContent Folder. (Select Web Content Folder-->hit control+N-->type

ht)

2) Configure the Resource

All the resources must be configured in web.xml (Deployment Descriptor Stub)

3) Deploy the Resource

All the resources must be deployed onto the webapps folder of Apache-Tomcat Server.

Go To Apache-Tomcat Folder--->webapps- "create new Folder\web-application)--- paste(WEB-INF(web.xml)+Home.html)

Note

Goto Agache- Tomcat -start the server
Goto any browser -and type
<http://localhost:8080/basic>

(folder name which you have created inside webapps)

> Configuration of Apache-Tomcat Server in Eclipse(For Internal way of execution)

Before configuration of Apache-Tomcat Server must not be in use anywhere externally.

Go to Window and Select Preferences.

Open Server preference and click on Runtime Environments

"Click on add and select the respective version of Apache-Tomcat Server and click on next.

"Browse for: the appropriate version of Apache- Tomcat Server and open Apache-

Tomcat Server folder and click an finish
Apply and close

After this Just run the application
Internally

A Servers folder will be generated within
the navigator

Right Click on Project ---> Run as--> Run On
Server

>Welcome File or Landing Page

A file or page which is automatically
displayed whenever client uses an
application is known as Welcome File or
Landing Page.

"index is considered to the default
welcome file or landing page which is
automatically loaded by the JEE container
whenever client uses an application

"We can explicitly make any file as Welcome File or Landing Page by renaming it as index.

"If the configuration file is not available ,then the JEE container fails to load an application where it throws http 404 error(Client Error).

>web.xml or Deployment Descriptor Stub

"it is a configuration file in xml format which is used to store the information with respect to the configurable resources of an application

*Each and every application must mandatorily have only one web xml

present without which the JEE container fails to load an application where it throws http 404 error(Resources not available).

"The current version of xml used is 1.0.

"The root tag for a xmi file is cweb-app>

>Servlet

Servlet is a server side java program which performs all the 3 different types of Logics such as Presentation Logic ,Persistence Logic and Business Logic along with which process Http Client request and get back some Http response

>Types of Servlet:

There are 2 different types: of Servlet present namely

1.Genericservlet

2.HttpServlet.

GenericServlet - Since it is not specific to any protocol or independent of protocol hence the name GenericServlet

*GenericServlet Does not supports Session.

>Session - Any activity which takes place between start time and stop time is known as Session.

-GenericServlet is an abstract class present in javax.servlet package.

"GenericServlet contains 3 methods in it out of which one is an abstract method and other 2 are concrete methods.

"The abstract method present in

GenericServlet is named as service() which has to be overridden mandatorily for 2 important reasons namely.

1. since service() is an abstract method.
2. Since service() is the only method which takes ServletRequest and ServletResponse as a parameter which is responsible for processing the client request.

Method Signature:

```
+void service(ServletRequest  
req,ServletResponse resp) throws  
ServletException, IOException.
```

"The other two methods present in Generic servlet are named as init() and destroy() where overriding these 2 methods are optional since these 2 are

concrete methods.

> writing a GenericServlet

```
public class OurServlet extends  
javax.servlet.GenericServlet  
{  
    @Override  
    public void service(ServletRequest  
req,ServletResponse resp) throws  
ServletException,IOException  
{  
    // write Servlet Logic here//  
  
}  
}  
}  
}
```

HttpServlet

"Since it is specific to a particular type of

protocol called Http protocol hence the name HttpServlet

"HttpServlet Supports Session

'HttpServlet is an abstract class present in javax.servlet.http package

"HttpServlet contains only concrete methods without any abstract methods in it.

- In case of HttpServlet we need to override a respective concrete methods called doXXX() method for a particular type of Http Request

Method Signature:

```
protected void doXXX(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException.
```

-There are 8 different types of Http

Requests are present

1.POST

2.GET

3.PUT

4.TRACE

5.DELETE

6.OPTION S CONNECT

>Writing a HttpServlet:

```
public class OurServlet extends  
javax.servlet.http.HttpServlet  
{  
    @Override protected void  
    doX(HttpServletRequest  
    req,HttpServletResponse resp) throws  
    ServletException,IOException  
    {  
        //Servlet Logic Here//  
    }
```

}

"Web Resources can be accessed based on unique url pattern.

Since Servlet is also a web resource ,it has to be accessed based on Unique url pattern.

Any resources can always be configured in two different ways namely

- 1.web.xml
- 2.Annotation

"To configure any resource using annotation ,it is mandatory to use JEE 3.x or above.(x=@,1,2,3..)

> Configuration of Servlet in web.xml

There are 3 different properties to be

configured for a Servlet in web.xml

1. Servlet-Name(Unique)

2. Url-Pattern(Unique)

3. Fully Qualified Class Name

> UI/Form Data

Diagram The data which is entered by the enduser(Client) on a form page and is submitted to the server in the form of key and value pair is known as UI/Form Data .

All the keys associated with respect to form page are represented as unique Identifier in the form of id name.

"The UI/Form Data is always associated with a request and can be accessed only within the service() since service() is the only method which takes ServletRequest and ServletResponse as parameter which

is responsible for processing each and every client request.

"Once the request is made the data's are carried to the server as a part of request object in the form of key and value pair which is displayed in the url

"The UI/Form Data can be fetched by using getParameter().

Method Signature:

String getParameter(String key)

- a)In this method the key is taken as an argument.
- b)If the key is present then the method returns associated value.
- c)If the key is not present then the method returns null but not any error or exception.

> Steps to create a Dynamic Web-Application.

- 1) Open Eclipse in JEE Perspective
- 2) Open Navigator mode
- 3) Right click within the navigator and create a new dynamic web project and name it.
- 4) Generate web.xml(Deployment Descriptor Stub):(It will be generated inside WEB-INF folder)
- 5) Add servlet-api.jar file into the lib folder,
- 6) Create Resource-->
 - 1) Form.html|(Select webcontent folder---hit control n---type- ht---and name the file)
 - 2) FirstServlet.java --->select the src folder-->to create standard package structure[select src folder-->hit control +n---> type pac(package)--select the application name and hit control+n to

create a class.

7) Configure Resource-->

1.using web.xml

2.Through annotation

8) Deploy Resource"-->webapps-->create a new folder->

1.WEB-INF (lib folder+webxml+ classes folder).

2.Form.html