# Servlets:

It is an API , which generally used to build an web application

Web-application:

An application which can be accessed over an internet those applications are called as web applications

*all the classes and interfaces regarding to servlets are present inside javax.servlet package

*Servlet is also an interface present in javax.servlet package

*servlet is a program which runs on server,

*servlet classes are going to inherit the properties of server and it produces the response based on client request

*if we want to create any servlet class then we need to implement servlet interface ,these servlet interface will be having some method ,then we need to provide implementation to all the method of servlet

Client- server architecture

the client-server architecture refers to a concept where client and server components interact to fulfill requests and provide responses over a network. Servlets are Java-based components that run on a web server, handling incoming requests from clients (such as web browsers) and generating dynamic responses.

1. **Client**: The client is typically a web browser or a client-side application that initiates communication by sending HTTP requests to the server. These requests are usually triggered by user actions such as clicking a link or submitting a form.
2. **Server**:

   Server is a computer,will have os,and also supports for the applications,which are supported in the system

The server hosts the servlets, which are Java classes that handle incoming requests and generate responses.

Any web application that has built gonna deploy inside the server and each application will have unique identification ,that uniqueness is predicted by the url,each application will have there own url

*inorder to connect with client to server will make use of internet

*whenever the client sending the request it should contain the url ,because there are many applications those are deployed in the server

*mainly request should contain

a.url(uniform source locator)

b.method type

c.data (if it is necessary)

whenever we are sending the request url and method type is compulsory ,data will be sending only when it is necessary

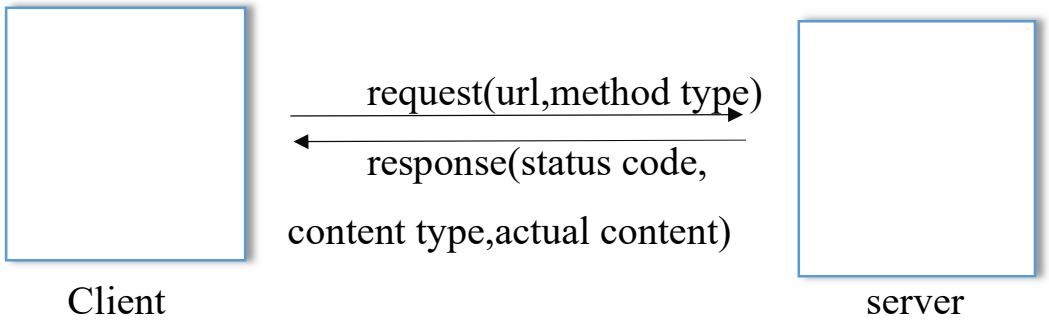
*when request sent by the client the server will procees it and send back the response ,whatever the response generated by server contains

a.status code(server will generate few codes for ex: 400,404,500……inorder to inform client about the request,each status code will represent whether request is success,rejected etc…..)

b.content type(while sending the response ,server may send some media file ,video file,images etc those are called as content type.

Server will check that the content is viewable in a particular client machine then only content will be display)

c.actual content




# Generic Servlet

It is an abstract class ,present in javax.servlet package

*this GenericServlet implements Servlet interface and given the implementation to all the abstract methods except servive(),

*since one of the method remains as abstract ,this class is called as abtract class

*if we want to create our own servlet class means we needto implements GenericServlet and we need to provide an implementation for service()

And your logic has to be written inside the service method ,then it can be triggered

**HttpServlet:**

The HttpServlet is an abstract class present inside javax.servlet.http

*in httpservlet there no abstract method but still they have made httpservlet as an abstract class because ,the one who developed this they dnt want any one create an object of it directly

*for servlet class we were generally extending GenericServlet but it is not recommended ,instead will extends HttpServlet class and will give implementation for those method

*HttpServlet is asubclass of GenericServlet

We have few methods present in httpServlet

1.service(ServletRequest req,ServletResponse res)

2.service(HttpServletRequest req, HttpServletResponse res)

3.doget(HttpServletRequest req, HttpServletResponse res)

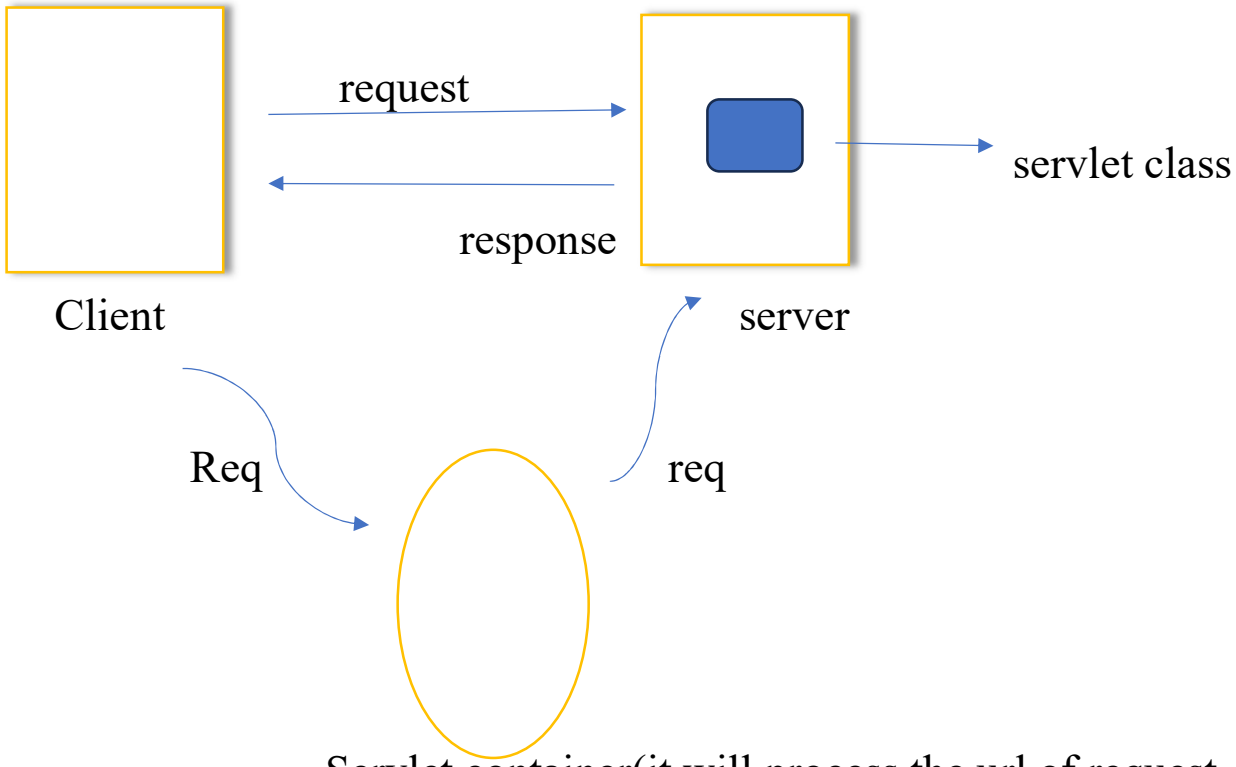
4.dopost(HttpServletRequest req,HttpServletResponse res)

5.dodelete(HttpServletRequest req,HttpServletResponse res)

# How does servlet works

Servlet is a small piece of program that run on the server ,now with any device we send the request to the server

Then server will sends the request to the servlet Container ,where servlet container will process the request and it going to trigger proper servlet class present inside the server,and inside servlet classes we will be having service() where that method is invoked and accept the request and sends back the response




**Servlet life cycle**

The servlet lifecycle is controlled and takencare by our servlet container

1. Loading servlet class

Class loader is responsible to load the servlet classes into memory

class loaders loads the servlet classes into the memory

when it receive first request ,for second time if we requested also it wont be load the object into memory again

2.Instance of servlet classes

servlet container will creates an object of servlet class ,inorder to access the non static method inside it

the object of servlet class is going to create only once in the entire lifecycle ,because web container will instantiate an object only once

3.Intialization

Our servlet container will initialize the values present inside an servlet classes by using init()

All these things will happen only once for entire life cycle

4.invoking service()

This step is much important ,where this method is going to invoked for each and every request that we make

5.Destroy servlet class

Servlet container will destroys the servlet class by using destroy()

And this is gonna happen only once for whole servlet lifecycle

## Welcome-file-list

The welcome file list specifies a list of filenames (such as HTML, JSP, or servlet files) that the servlet container should consider as default resources when a client requests a URL corresponding to a directory (e.g., **`http://example.com/`**) instead of a specific file.

Whenever we are launching some web pages, we need a default page to get open,that default page we will specifies as welcome-file-list

In our servlet ,for default welcome page we have index.jsp , if we want configure customized welcome page ,then we have to configure few code inside web.xml

i.e

```
<welcome-file-list>
    <welcome-file>pageA.html</welcome-file>
    <welcome-file>home.html</welcome-file>
</welcome-file-list>
```

Here our servlet container will search for user define welcome page ,if it is not there then it will execute index.jsp always

## RequestDispacther

It is an interface present in javax.servlet package

The RequestDispatcher interface in java servlet programming provides a way to forward the request from one servlet to another resource (jsp,html)(or) include the content of another resource

The RequestDispacther we have 2 methods

1.forward(req,resp)

2.include(req,resp)

Forwarding Request:-forward(req,resp)

forward(req,resp) this method is used to forward the request,

forwarding the request means passing the data from one servlet to another resources (jsp,html)

this is commonly used when one servlet is responsible for preprocessing a request and wants to pass it on to another resource to generate the response

Including content: include(req,resp):

Include(req,resp) method is used to include the content ,means including the one resource inside another resources,may be one jsp inside another jsp etc…..

Note:

We can get RequestDispatcher object by using getRequestDispatcher() by using request attribute

# sendRedirect()

In Java servlets, the **`sendRedirect()`** method is a part of the **`HttpServletResponse`** class. It's used to redirect the client's browser to a different URL, either within the same web application or to an external URL.

```
resp.sendRedirect("url");
```

## ServletContext

ServletContext is an interface in java servlet ,where in servlet we need a object of servletcontext inorder to store some information ,and where that information will be accessible by all the servlet classes present in web application

Whenever we need some information has to be shared for all the servlet classes present in web application ,instead of writing code in each n ever servlet classes ,and making code lengthier ,we can use object of servletcontext

*Note:

Internally our servlet container will create an object of servletContext ,when it receives the first request from client

We can get the instance of ServletContext by using getServletContext()

*inside the servletContext the values are going to be stored in the form of key and value pair

We have few methods present in servletContext:

1.getInitParameter(“key”)- this method is used to get the value present inside context object based on key

2.setInitParameter(“key”,”value”);- this method is used to set the value inside the servletcontext object

3.setAttribute(“key”,object reference);

This method is used to save the object inside the context

4.getattribute(“key”)- this method is used to get the object from context based on key associated with it

- getattribute method return object type,need to downcast it for spicofoed object type

# SessionManagement

It refers to the process of maintaining stateful information about a users interaction with a web application across multiple request

Since the http request is stateless ,meaning each request from a client to a server is independent

*when we send the request to server ,it will considered each request as a new request ,because http protocal are stateless,so server cannot remember details about us,

So everytime when we want to open an web application ,it will ask the authentication , so we must manage the status (or) session , so that our server will see the status and provides the response who already logged in

To manage the session we have sessionmanagement

Inorder to manage the session we have 4 ways

1.cookies

2.hidden form field

3.url re-writting

4.httpSession

# Cookies

It is a class in java servlet present in javax.servlet.http package,which is used to manage the session

Cookie is an object which can store a small piece of information

*cookies generally created in server side and stores in the client side

Whenever client sends the first request cookies are created by server ,and when server sends back the reponse it will attach the cookies along with the response and sends to client

For the second request along with request cookie is also attached ,then server will view the cookie and sends the response,without asking the authentications again

Note:

We can create object of cookies in 2 ways

1.new Cookie()

2.new Cookie(“key”,”value”);

- Cookies will stores the values in the form of key and value pair
- For one application we can create multiple cookies

Advantage:

1.simple technique

2.stores in client side

Disadvantage:

1.cookies can store data only in the form of strings

2.if browser disabled the cookies then we cannot able to create the cookies

3.in cookie we can able to store only value ,object cannot be stored

HttpSession:

It is an interface present in javax.servlet.http package

We have cookies also but we already had a lots of drawbacks ,by using this httpsession we can overcome all the drawbacks of cookies

Note:

We can get the object of HttpSession by using getSession() by request object we can call getSession()

Getsession() – this method will return the HttpSession object if session exists ,or else it will create new session object and it is going to return that object

*to the HttpSession we can add any value

Ex: String ,int ,double,student object……..etc

*sessions are one for the application

*we can store the object inside httpsession

Jsp;

Jsp stands for java server page, it is a technology used for building dynamic web-pages and is a part of JEE platform

*jsp allows developers to embedded java code with html page,which are then compiled into servlets by the service's js engine,thus servlet generate dynamic content that can be displayed in a web-server

*we have to create jsp page under the webapp,and extension should be .jsp

*when we want to design UI without having any java code then we will go with html

*if we want to display UI with having java code then will go with jsp

JSP lifecycle :

1.translation

Initialy when servlet container recives the first request ,servlet container will convert jsp into servlet class

2.Compilation

Servlet class generated by jsp should compile

3.class loading

Once after the compilation the servlet class get loaded into memory

4.Instantiating

Our servlet container is gonna create an object of servlet classes ,inorder to access non static members

5.initialization

Servlet container will initialize the members of jsp which converted as servlet class by using jspinit()

6.invoking service()

Service() method is invoked inorder to accept the request and produce the response

7.destroy

Servlet container will unload the servlet , and it will invoke the destroy() to destroy the jsp page