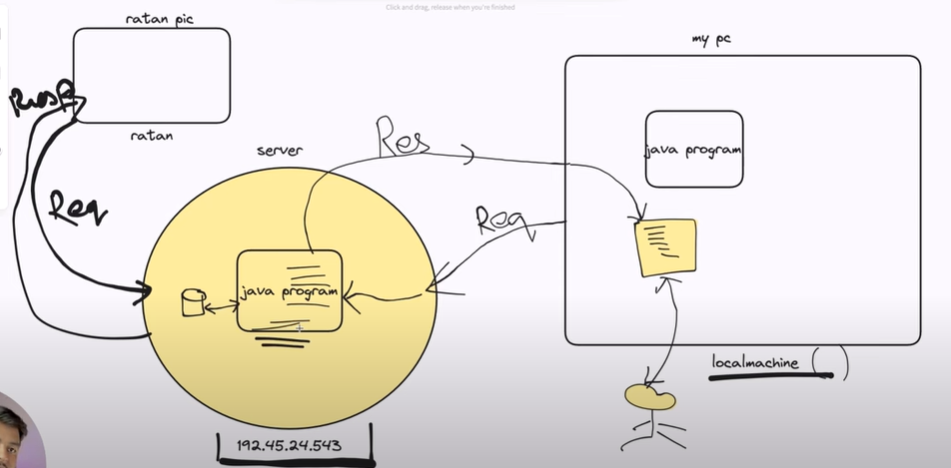
**Servlet**

Servlets are java programs that runs on web servers and performs all the three logics such as Presentation logic, persistence logic and business logic and along with that it also handle the client request, it receives the client request, process it and give response to the client.



**CLIENT-SERVER ARCHITECTURE**

Client will make a request to server, server will process that request and give back response to the client.

Client can be your browser or mobile application etc.

Server – Server manages all the resources and process the client request and give back response to client.

What is Server?

* Server manages all the resources,
* It provide runtime environment to our application
* it takes client request, process it and give back response to client

**WELCOME FILE/LANDING PAGE**

* Welcome file or landing page is the first page which the user will see automatically whenever he uses the application.
* The file whose name is index will be considered welcome file or landing page by default.

**External way of deployment**

1. Copy WEB-INF folder and paste inside the webapps by making a new folder(choose name of your choice) – lib, web. xml, classes folder (which will be present in build folder) OR if build folder is not opening then directly create WAR file and export it to webapps folder.
2. Copy the resource and paste inside folder created by you inside webapps folder like html file.

**web.xml**

1. It is a configuration file where we configure resources of an application and it is present in .xml format.
2. During external deployment every application must have one web.xml file otherwise JEE container will not be able load the application and throws http404 Error(resource not found).
3. Root tag of web.xml file is <web-app>.
4. Current version of xml is 1.0

**Session**

* The time-period of interaction between the user and application is called session.

(from login to logout) .

Servlet

Definition of servlet

There are two type of Servlets

1. GenericServlet
2. HttpServlet

**GenericServlet**

* It is independent of protocols(it does not follow any protocol) that why it is called GenericServlet.
* It does not support session.

GenericServlet is present in javax.servlet package.

GenericServlet is a abstract class(means there will be atleast one abstract method)

GenericServlet has 3 methods out of which 2 are concrete and 1 is abstract method.

The Abstract method name is service() method and it is compulsory to override this method for 2 reason

1. It is abstract so we have to provide implementation to it by overriding it.
2. service() method is the only method which takes ServletRequest() and ServletResponse() as parameter which are responsible for processing the client request.

The concrete method names are init() method and destroy() method overriding these methods is optional as they are concrete method.

**Method Signature of service method**

public void service(ServletRequest req, ServletResponse resp)throws ServletException,

IOException {

}

**Creation of GenericServlet**

public class FirstServlet extends GenericServlet{

public void service(ServletRequest req, Servlet resp) throws ServletException, IOException{

//Servlet logic

}

}

**HttpServlet**

* HttpServlet is specific to Http protocol that why its name is HttpServlet.
* Httpservlet support session.

HttpServlet is present in javax.servlet.http package

HttpServlet is an abstract class but it contains only concrete methods, No abstract method present.

In HttpServlet we need to only override one concrete method called doXXX() method.

**Method Signature of doXXX Method**

Get, Post

protected void doXXX(HttpServletRequest req, HttpServletResponse resp) throws

ServletException , IOException {

//Write your Servlet logic here

}

**Writing a HTTPServlet**

public class FirstServlet extends HttpServlet{

protected void doGet/Post(HttpServletRequest req, HttpServletResponse resp) throws

ServletException , IOException{

}

}

**Configuration of Servlet**

There are two ways to configure Servlet

1. web.xml
2. annotation

So if we are using web.xml to configure our servlet, the we have to use 3 properties inside web.xml

1. servlet name – it should be unique
2. url-pattern- it should be unique
3. fully qualified class name

And using annotation

* No need of web.xml(you can delete your web.xml)
* Rename html file to index.html
* Add annotation in your .java file (where you have written servlet code)

**UI/FORM DATA**

* The data which is entered by the user in the form page and submitted to the server in the form of key value pair is known as UI/FORM Data.
* All the keys must be unique and that key is represented as id/name in the form page.
* And if you are using genericServlet inside genericServlet only service method can access this data because service is the only method which has ServletRequest and ServletResponse as a parameter which are responsible for processing the client request. Like – req.getParameter(“key”).
* If you are using HttpServlet then this data will be accessed inside doXXX() method, as it has HttpServlet and HttpResponse as a parameter which are responsible for processing the client request. Like – req.getParameter(“key”)
* And data will always come in String datatype.
* If the key is not found then it will return null value, it will not throw any exception.

|  |  |
| --- | --- |
| GET | POST |
| Get is used to get some content from server | Post is used to post some content in server |
| Get does not make changes in server side | Post make changes in server side |
| Get deals with limited data | Post deals with unlimited data |
| Get request can be bookmarked or saved | Post request cannot be bookmarked |
| In case of Get, data is not safe as data is shown to the end user when data is taken from client to server. | In case of post, data is safe as it is not shown to the enduser when data is taken from client to server. |
| In case of link , we use get request. | In case of n number of data, we use post request. |