Backend and Database Development

13 classes

Day 8 : May 3 2024

Java Technologies

Core Java or JSE (Java Standard Edition)

Web Application or JEE (Java Enterprise Edition)

https://[www.google.com](http://www.google.com) URL

req(http/https)-----------🡪

Client Server

🡨-------------res(http/https) html/ html5

Css/css3

JS (JavaScript)

Angular / React JS

Backend technologies

Java (JEE) / Spring framework and Spring boot

Asp.net

Php

Python with Django or Flash

Node JS

Etc

JEE : Java Enterprise Edition

Servlet , JSP and EJB

Servlet

Java Server Pages

Java Enterprise Bean

npm start or ng server

react js by default run on default port number 3000

angular by default run on default port number 4200

Server : Server is use to run the application with unique port number and they give service for us.

Server mainly divided into two types.

1. Web server : tomcat part of Apache Company
2. Application server : web logic or jboss or WAS etc.

To run servlet or jsp or ejb we need server. Servlet, jsp or ejb doesn’t contains main method we compile the program and run the application on server. So server only responsible to take the execution of servlet, jsp and ejb.

All type of server provide container. Container is a part of server also known as engine which is responsible to execute servlet, jsp and ejb application.

There are two types of container

1. Web container : if server is type of web server which contains only one type of container ie web container responsible to execute servlet and jsp program.
2. Ejb container : if server is type of application server which contains more than one type of container ie web container, ejb container , jms container etc. web container is responsible to execute servlet and jsp and ejb container responsible to execute ejb program.

Servlet : Servlet is normal java program which help to create dynamic web page on server side.

import javax.servlet.\*;

servlet : servlet is a package which contains set of classes and interfaces.

import javax.servlet.Servlet;

Servlet : Servlet is an interface which contains set of methods. This interface contains totally 5 methods.

init, service, destroy, getServletInfo, getServletConfig etc. among these 5 methods starting 3 known as life cycle methods init, service and destroy.

Init :this method call only once at the starting time when client send the request. Which help to do some initialization.

Service : this method called again and again whenever client send the request. Which is use to take the request from a client and give response back to the client.

Destroy : this method get called at last whenever no client accessing that application. This method is use to close the resources.

1st Option

public class MyDemo implements Servlet {

then we need to override all 5 methods.

}

2nd option :

GenericServlet : it is a type of abstract class which internally implements Servlet interface and provided body for 4 methods except service methods.

public class MyDemo extends GenericServlet {

we need to provide the body for service method.

}

3rd option

HttpServlet : it is a type of abstract class which internally extends GenericServlet and provided body for service method as well as it provide some extra method in the form of doXXX like doGet, doPost, doDelete, doUpdate etc

public class MyDemo extends HttpServlet{

we can override service or doGet or doPost

}

<http://localhost:8080/ProjectName/UrlMapping>

through hyper link always servlet doGet method get called.

Through form by default it call doGet method and form internally use get method as default method and it call doGet method. In doGet method information send through URL rewriting technique.

URL?key=value if we pass one information

URL?key=value&key=value&key=value if we pass more than one information.

Get method is not secure.

In form we can use the attribute as method=”post”. This method call servlet doPost method. In post method information send through request body part.

Performance wise post method is slower than get method.

RequestDispatcher : it is an interface which provide set of method which help to navigate from one page to another page.

Syntax to create RequestDispatcher interface reference.

RequestDispatcher rd1 = request.getRequestDispatcher(“path”);

Path 🡪

1. Target page is servlet then page must be that servlet page url pattern or mapping.
2. Target page is html or jsp. Then path must be pageName.jsp or pageName.html.

RequestDispatcher rd1 = request.getRequestDispatcher(“Home”); target is servlet

RequestDispatcher rd2 = request.getRequestDispatcher(“login.html”); target is jsp

rd1.include(request,response); output as source page + target page as one output.

rd1.forward(request,response); output as target page only.