

Different types of application built using java

======================================

a. Standalone applications (JSE)

1. CUI applications

2. GUI applications

b. Enterprise applications (JEE and frameworks)

1. Web applications

2. Distributed applications

Web application:

* Web application is a server-side application, it will be designed without distributing application logic over no of jvm.
* To build web application we need to use technologies like CGI, Servlet, JSP and so on ...
* The main purpose of web-application is to generate dynamic response from server machine.
* Web-application provides services to web-clients only (using the browser we need to send the request)

eg: BMS (webapp) ------- > to send the request to this application u need browser + internet

mobile based apps(internet)

* Web-application => client to server model
* In case of web-applications to execute the program we need "webservers".

Distributed application

* Distributed application is a server-side application, it will be designed by distributing application logic over multiple no of jvm.
* To build distributed application we need to use technologies like RMI (remote method invocation), EJB's, Web-Services,
* The main purpose of distributed application is to establish the communication b/w local machine and remote machine to access the remote services.
* Distributed applications will provide service to any type of clients.
* Distributed-application => business to business model.
* In case of distributed-applications to execute the program we need "application servers".

Web-application

The application which is developed only using web-based technologies like html, css, javascript, servlet, jsp etc is called "web-application"

application ===> collection of many programs

webserver:

if we want web-applications to execute, we need one special software that special software only we call it as "webserver".

webserver provides environment to run web-applications

eg: tomcat, resin, jetty, glassfish, jboss, oracleweblogic server, etc..

Deployment and Un-Deployment:

The process of placing the web-application inside webserver is called "Deployment".

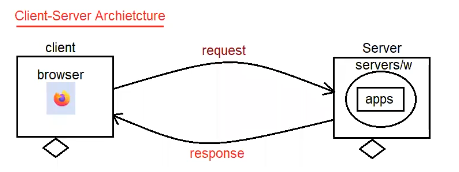
The process of removing web-application from the webserver is called "Un-Deployment".

Web-client:

To send the request from the user we need to have special software installed in the client machine.

The special software is only called as "browser".

eg: mozilla, chrome, safari, etc



web programming for static response

static response:

The response which won't be changed from person to person and time to time, such type of response is called as "static response".

eg: login page of gmail

home page of icici bank

Flow diagram of static response

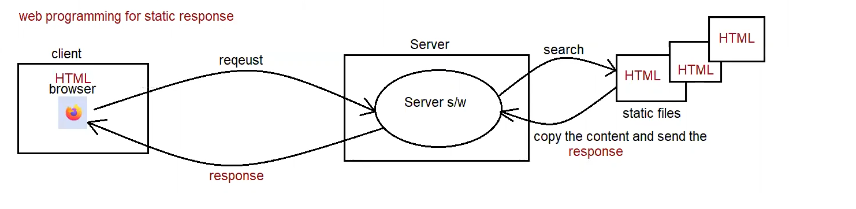
1. client send the request for static files to the server

2. Server searches whether the requested resource (html file) is available or not.

3. If the request resource (html file) is available then server will provide that file as response.

4. If the requested resource (html file) is not available then we will get 404 status code saying requested resource not available.

Note: To server static files, no processing is required at the server side, hence webserver always loves to serve static files.



dynamic response:

The response which is varied from person to person and time to time, such type of response is called as "dynamic response".

eg: inbox page of gmail

balance information of icici bank.

Flow diagram for dynamic response

1. client sends the request for webserver

2. webserver will check whether the request is for static resource or dynamic resource (based on url)

3. if it is a static resource, then webserver only will search for static resource, if it is available server will provide the static file contents (copy and paste) as the response the client.

if it is not available, then 404 status code will be sent as the response to the client saying the requested resource is not available.

4. if the requested resource is for dynamic information, then webserver will forward the request to web container.

5. web container will search for the helper application, which needs to be executed.

6. if it is not available, then 404 status code will be sent as the response to the client saying the requested resource is not available.

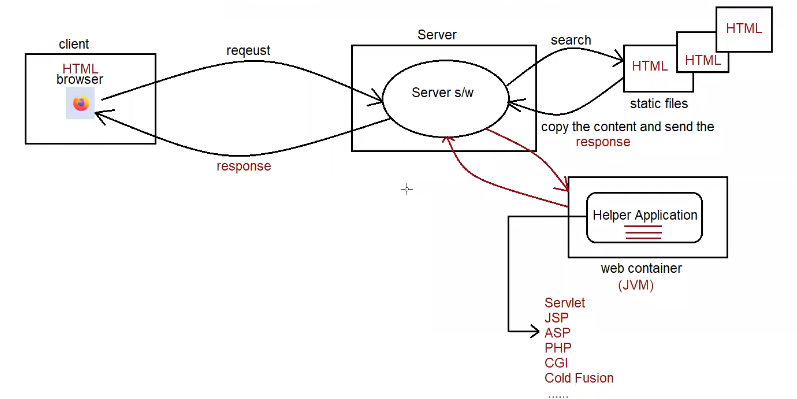
6. if it is available, then the requested helper application will be executed and it will be sent as the response to the webserver and webserver inturn will send as the response to the end user.

7. During the execution if any problem occurs then it would result in exception and status code 500 would be sent as a response to the end user by the server.

Note: To generate the dynamic response at the server side we need some helper applications. To build these applications which are capable of generating dynamic response we need to learn technologies like

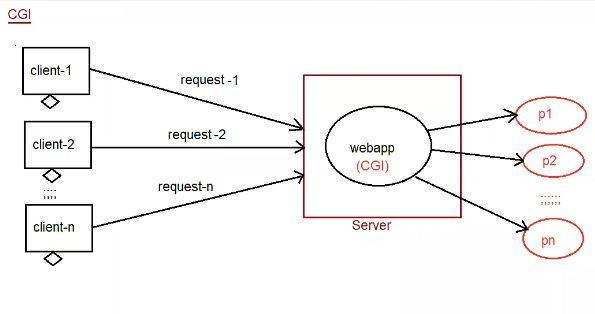
a. Servlet

b. JSP



To design a web-application, we already have CGI then what is the need to go for Servlet?

CGI => it stands for Common Gateway Interface.



CGI is a server side web-technology which is built on top of 'c' language, c language is a process based language, which inturn make CGI as "process based technology".

if we deploy any CGI application, then container will create a separate process for every request.

Process is heavy weight component, to handle single process system has to consume a lot of memory and execution time.

Due to the above reason, more the request comes server would be getting a burden of creating a process which inturn reduce the system performance and increase the response time for the client.

To reduce the burden on server and to increase the performance we need to use server-side technology called "Servlet".

Servlet is a server side technology which is built on top on "Java language".

Java language is Thread based technology.

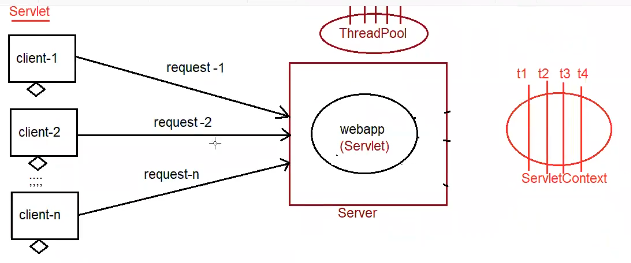
if we deploy a Servlet application at the server side then for every request servlet container will generate a separate thread on the respective Servlet Object.

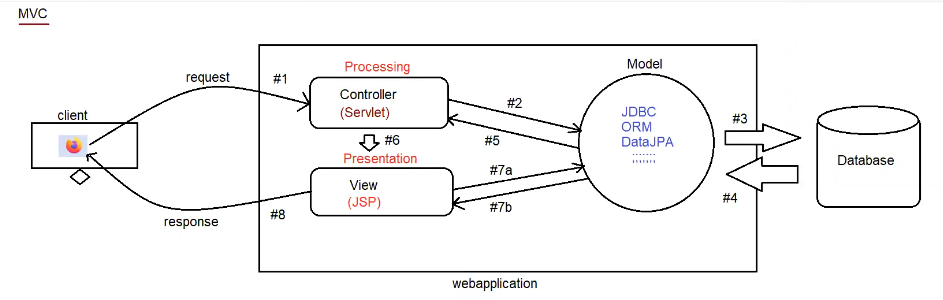
In the above context, if we increase the no of requests container will create a separate thread instead of process.

When compared to process, threads are light weight, since it is light weight, server would not be burdened.

server would provide quick response for client requests which increase the performance of the application.

To design a web-application, we already have Servlet then what is the need to go for JSP?





Servlet

1. To build web applications using Servlet, we need to have knowledge of Java properly.

2. Servlet is mainly meant for Processing logic (pick the request and process the request).

3. Any modification done in the Servlet, then we need to perform compilation and reloading on the server explicitly.

4. If we build webapps using MVC design pattern, then Servlet will placed inside Controller logic.

5. In case of Servlet, we are unable to separate both presentation logic and business logic

JSP

1. To build web applications using JSP, it is not required to have any java knowledge only presentation skills are enough.

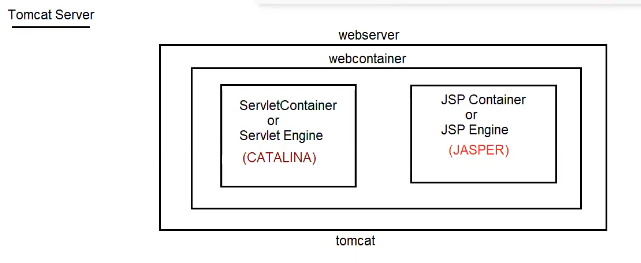
2. JSP is mainly meant for providing dynamic response to the client with good look and feel (only presentation).

3. Any modification done in the JSP, then it is not required to do compilation and reloading because jsp pages are "Auto-compiled".

4. If we build webapps using MVC design pattern, then JSP will placed inside View logic.

5. In case of JSP, there will be a clear cut separation b/w presentation logic and business logic because presentation logic deals with html tags and business logic deals with "JSP tags".

Architecture of webserver (Tomcat)



Tomcat

1. It is webserver provided by apache foundation.

2. Every webserver will have web-container

* web-container is responsible to manage and execute servlet and jsp’s.

3. Internally web-container consists of 2 components

* catalina container (servlet container)
* b. jasper container (jsp container)

4. ServletContainer

* It is also known as ServletEngine.
* It is responsible for managing and executing servlet components.
* Tomcat servlet container name is "CATALINA".

5. JSP container

* It is also known as JSP Engine
* It is responsible for managing and exeucting jsp componenets
* Tomcat jsp container name is "JASPER".

Download the tomcat

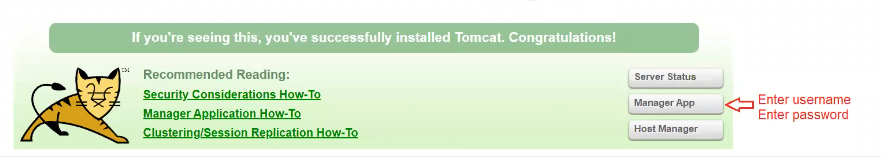
Set the shutdown port as 9090, And connector port as 9999

Note:

start the tomcat server by opening bin folder and select tomcat9.exe and double click on it.

now send the request by opening browser of your choice and hit the request as

http://localhost:9999





Servlet

It is an API which helps the programmer to build web-applications.

ServletAPI provides 2 packages

a. javax.servlet .\*

b. javax.servlet.http .\*

javax.servlet .\*

==============

1. Servlet(I)

2. GenericServlet(AC)

3. ServletConfig(I)

4. ServletContext(I)

5. RequestDispatcher(I)

6. ServletRequest(I)

7. ServletResponse(I)

javax.servlet.http.\*

===================

1. HttpServletReqeust(I)

2. HttpServletResponse(I)

3. HttpSession

4. HttpServlet(AC)

