

Unit No.-6 Middleware and Server side Technologies

* Syllabus -

- Introduction to middleware, Types of middleware, Applic servers, Intro to Java EE, Intro to Java EE technologies like JMS, JDBC, RPC, RMIS, Socket, EJB 3.0 Architecture, Entity, Session, msg beans, XML, XSLT, Specifications & characteristics of middleware technologies, Server side Technologies - Need of server side technology in multi-tier architecture, Java web services, Server side technol.- JSP, JSF, SCA, MVC, Java servlets, struts.

* Introduction to Middleware -

- Middleware is glw that runs betn client & server processes. It is the "glue" betn the client & server, which makes it possible for them to communicate to each other.
- Middleware is written in such a way that the user never notices its presence.
- It delivers secure & transparent services to users.
- It is used most often to support complex, distributed apps.
- It includes web servers, applic servers, content mgnt. systems, & similar tools that supports applic development & delivery.
- Middleware is especially integral to modern informat' technology based on XML, SOAP, Web services & service-oriented architecur.

* Types of Middleware -

i) Remote Data Access - (RDA) -

which implements a RDA protocol for sending data manipulat' lang. statements to an appropriate db server for processing & transporting the result back to invoking proc.

- 2) Remote Procedure calls (RPCs) — RPC is used in most new operating system services.
- 3) Message-Oriented middleware (MOM) — MDM can be used as a mechanism for storing & forwarding messages queuing. It can be used when client & server processes communicate asynchronously.
- 4) Object-Request Brokers (ORBs) — A standard implementation of the ORB standard is CORBA. ORB makes it possible to invoke a remote object by allowing a source object to send a msg to that remote object.
- 5) Distributed transaction processing (DTP) — This type of mechanism uses execution semantics to interact betw the client & the server.

* Remote Procedure Call (RPC) —

Defn- When a process on machine A calls a procedure on machine B, the calling process on A is suspended, & execution of the called procedure takes place on B. Information can be transported from the caller to the callee in the parameters & can come back in the procedure result. No message passing at all is visible to the programmer. This method is known as Remote procedure call (RPC).

- RPC is an inter-process comm' technology that allows a computer program to cause a subroutine or procedure to execute in another address space without the programmer explicitly coding the details of for this remote interaction.

* RPG occurs in the following steps -

1. The client procedure calls the client stub in the normal way.
 2. The client stub builds a msg & calls the local operating system.
 3. The client's OS sends the msg to the remote OS.
 4. The remote OS gives the msg  server stub.
 5. The server stub unpacks the parameters & calls the server.
 6. The server does the work & returns the result to the stub.
 7. The server stub packs it in a msg & calls its local OS.
 8. The server's OS sends the message to the client's OS.
 9. The client's OS gives the msg. to the client stub.
 10. The stub ~~un~~ unpacks the result & returns to the client.

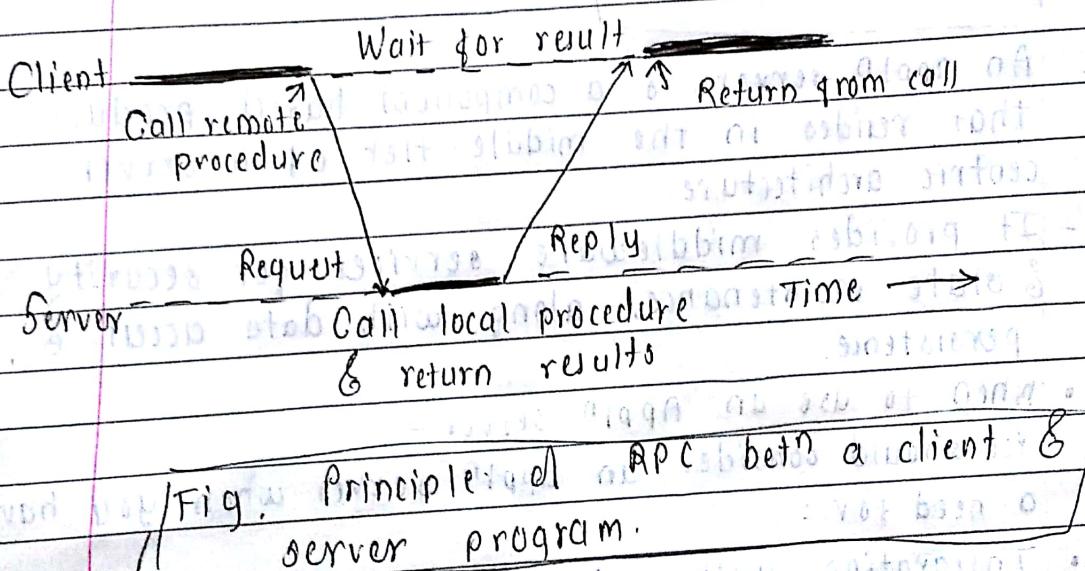


Fig. A. Principle of RPC between a client & server program.

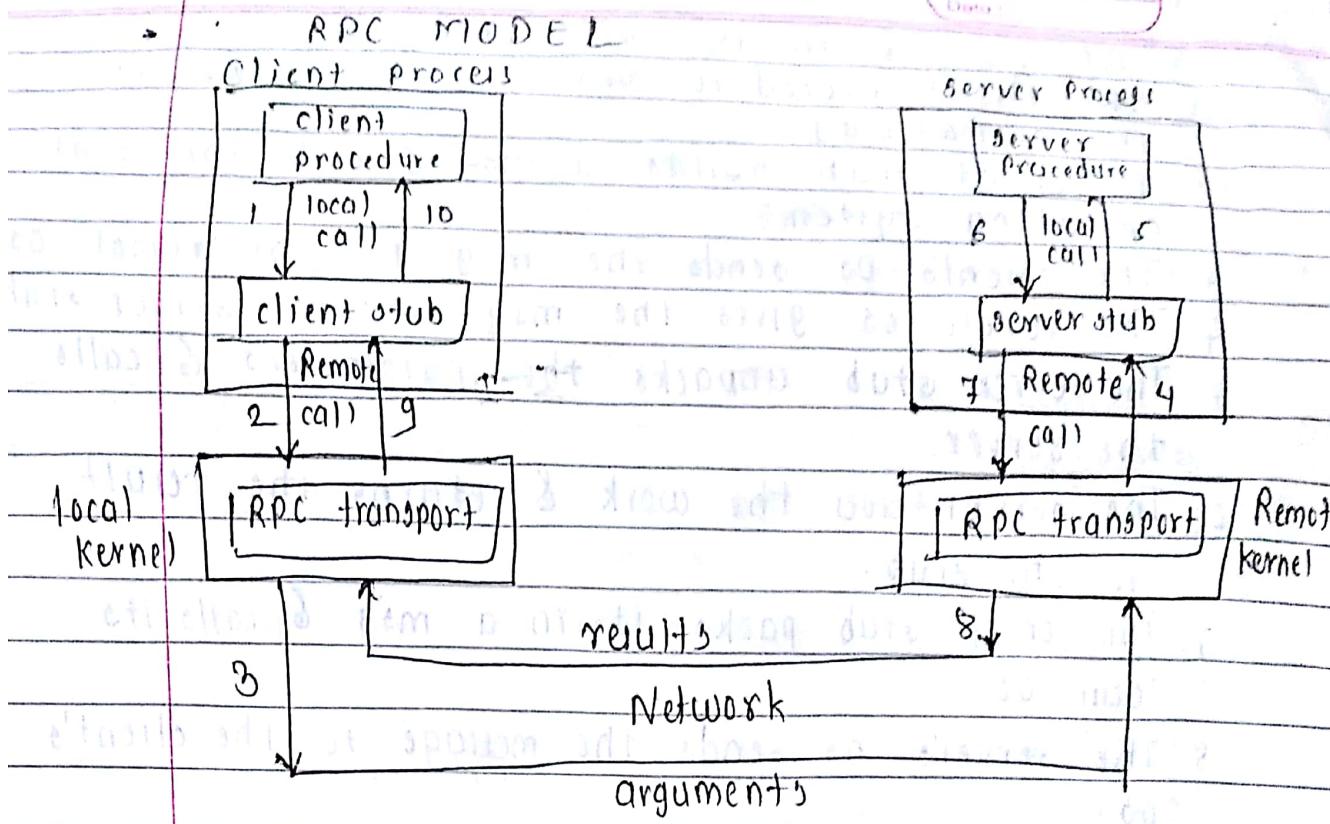
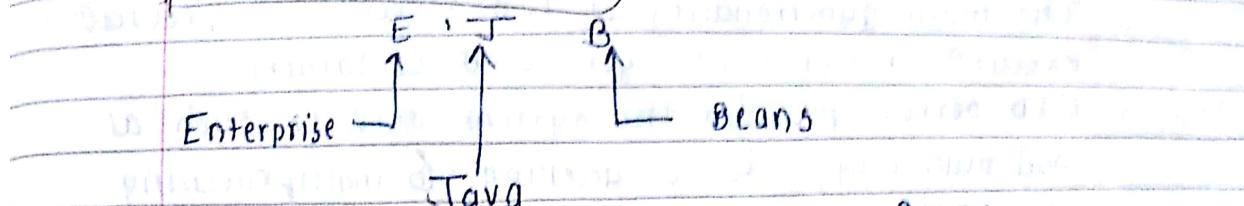


Fig. RPC Model

* Application Server

- An appln server is a component based product that resides in the middle-tier of a server centric architecture.
- It provides middleware services for security & state maintenance, along with data access & persistence.
- When to use an appln server -
- You should consider an appln server when you have a need for :
 - Integration with existing systems & databases.
 - Website support.
- Secondary reasons to use appln servers derive from the primary reasons. A few secondary reasons are :
 - E-commerce
 - Web-integrated collaborat?
 - Component re-use.

* EJB 3.0 Architecture :-



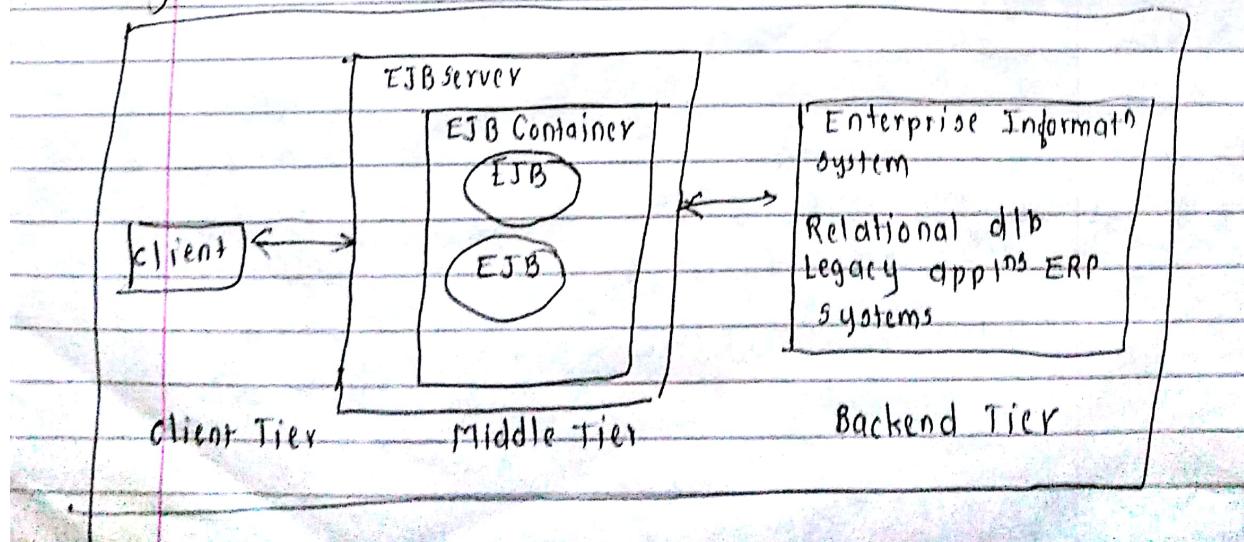
- EJB stands for Enterprise Java Beans.
- An EJB is a server-side component which contains the business logic of an appn.
- Business logic is the code written to fulfill the appn purpose.

e.g. In an Inventory system.

- EJB is an important part of a J2EE platform. The architecture of J2EE platform is component based which provides multi-tier & distributed features to the appns at enterprise level.
- EJB provides an architecture which is used to develop & deploy appns which are component based
- EJB provides robustness, high scalability, and high performance for the appn.
- EJB architecture is used to create server-side, multi-tier, user, secure, scalable, transactional & enterprise-level appns.
- EJB architecture supports client server or completely distributed n-tier appns.

a) Components of EJB architecture -

- 1) EJB server(s)
- 2) EJB container(s)
- 3) EJB clients
- 4) Enterprise Beans



1. EJB Server -

- The main functionality of EJB Server is to provide execution environment for EJB containers.
- EJB Server provides the system services such as load balancing, device accessing & multiprocessing for either EJB containers.
- The EJB Server acts as transaction processing monitor to provide transaction services.
- The EJB server also provides other services like security.

2. EJB Container -

- The EJB container is an interface in between EJB & low-level, platform-specific functionality which supports the Bean; the EJB Server has hosted EJB Clients.
- Using the proxy objects, the clients interact with beans.

B) 4. Execution -

- EJBs are deployed in an EJB container, typically within an app'n server.
- The EJB container is responsible for ensuring the client code has sufficient access rights to an EJB.

* Message Beans :-

- Message Beans (MB) are business objects whose execution is triggered by messages instead of by method calls.
- The message Driven Beans is used to provide a high level ease-of-use abstraction for the lower-level JMS (Java Message Service) specification.
- It may subscribe to JMS message queues or msg. topics, which typically happens via the activationConfig attribute of the MessageDriven annotation. They were added in EJB to allow event-driven processing.
- Unlike session beans, an MDB does not have a client view i.e. clients cannot look-up an MDB instance.
- An MDB just listens for any incoming msg on, for eg, a JMS queue or topic & processes them automatically.

* Session Bean -

- One way to think about the appn logic layer (middle tier) in the sample arch. shown in fig-1. is as a set of objects that, together, implement the business logic of an appn.

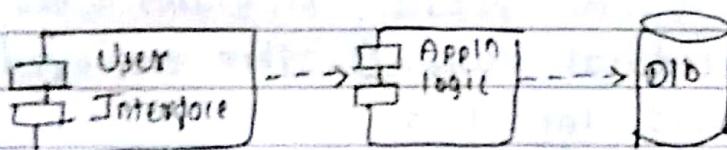


Fig-1 The classic model of a multilayered or layered arch.

- Session beans are the construct in EJBs designed for this purpose. As shown in fig-2 there may be multiple session beans in an appn. Each handles a subset of the appn's business logic.
- A session bean tends to be responsible for a group of related functionality.

- There are 2 types of session beans, which are defined by their use in a client interface:
 - 1) Stateless - These beans do not declare any instance variables, so that the methods contained within can act only on any local parameters.
 - 2) Stateful - These beans can hold client state across method invocations. This is possible with the use of instance variables declared in the class defn.

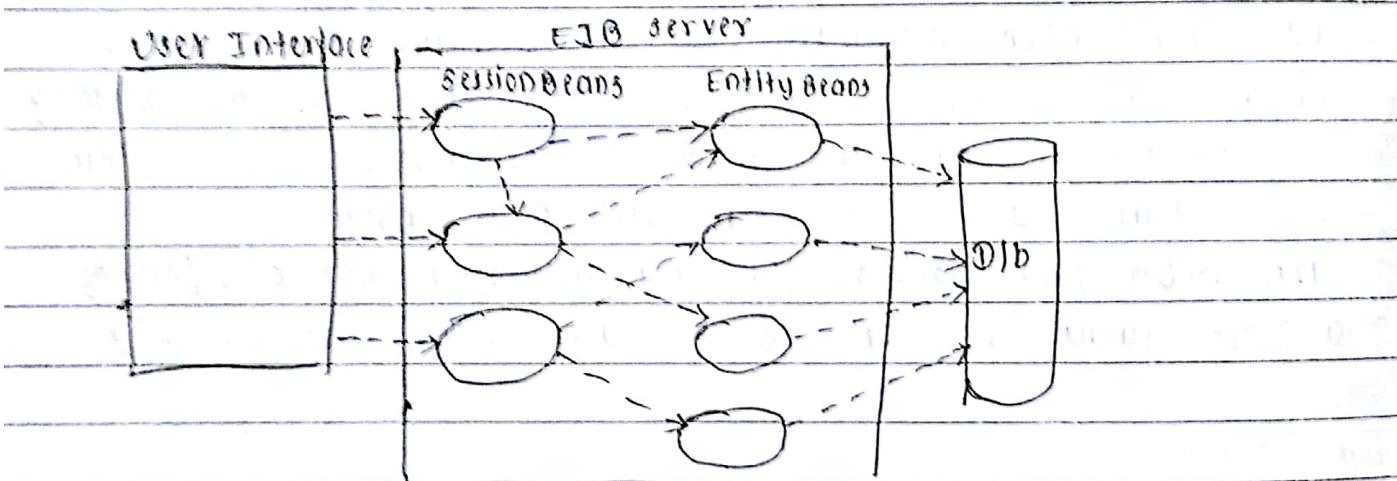


Fig. 2 Session & entity beans in an appln.

* Entity Beans -

- Before Object orientation became popular, programs were usually written in procedural lang. & often employed relational dbques to hold the data.
- The use of entity beans is one way to get the best of both of these worlds, for the foll. reasons:
 - Entity beans are objects, and they can be designed using object-oriented principles & used in appln as objects.
 - As shown in fig. 2 it is a good practice to call only sessions beans directly from the client, & to let the session beans call the entity beans.

* XSLT :-

- XSLT is an XML-related technology i.e. used to manipulate & transform XML documents. The acronym XSLT stands for Extensible Stylesheet Lang. Transformations, which is a mouthful to say, but it sounds more complicated than it is.
- With XSLT, can take an XML document & choose the elements & values you want, then generate a new file with your choices.
 - Because of XSLT's ability to change the content of an XML document, XSLT is referred to as the stylesheet for XML.

* Java Web Services :-

- Web services are client & server apps that communicate over the World Wide Web's (www) HTTP. As described by the W3C (World wide Web Consortium), web services provide a standard means of interoperating betw. silw apps running on a variety of platforms & frameworks.
- Web services are characterized by their great interoperability & extensibility.
- Web services can be combined in a loosely coupled way to achieve complex operatn.

* Types of Web Services -

- On the conceptual level, a service is a silw component provided thr a nw-accessible endpoint. The service consumer & provider use messages to exchange invocatn request & response informatn in the form of self-containing documents that make very few assumptions about the technological capabilities of the receiver.
- On the technical level, web services can be implemented in various ways.

- Two types of web services discussed below as "big" web services & "RESTful" web services.

1. "Big" Web Services -

- In Java EE, JAX-WS provides the functionality for "big" web services. Big web services use XML messages that follow the Simple Object Access Protocol (SOAP) standard, an XML lang. defining a msg. architecture & msg formats.

2. RESTful Web Services -

- In Java EE, JAX-RS provides the functionality for representational state Transfer (RESTful) web services. REST is well suited for basic, ad hoc integratn scenarios, - RESTful web services, often better integrated with HTTP than SOAP-based services are, do not require XML messages.

* Types of WS -

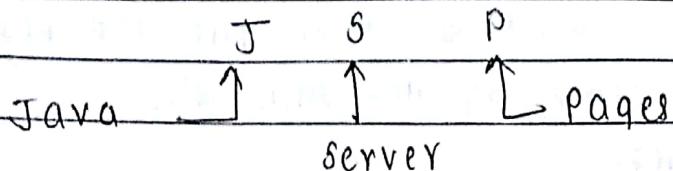
1. SOAP - SOAP stands for Simple Object Access Protocol. SOAP is an XML based industry standard protocol for designing & developing web services. Since it's XML based, it's platform & lang. independent.
2. REST - It is an architectural style for developing WS. It's getting popularity recently because it has small learning curve when compared to SOAP.

- Resources are core concepts of RESTful WS & they are uniquely identified by their URIs.
- Java provides its own API to create both SOAP as well as REST web services.

1. JAX-WS : JAX-WS stands for Java API for XML Web services. JAX-WS is XML based Java API to build web services server & client appn.

2. JAX-RS : Java API for RESTful web services (JAX-RS) is the Java API for creating REST web services. It uses annotations to simplify the development & deployment of web services.

* JSP :-



- JSP stands for Java Server Pages. Jsp is a server side Java technology used to create dynamic contents just like Servlet.
- JSP is an extension to servlet & provides more functionality as compared to servlet.
- Advantage - A JSP page is combination of HTML tags & JSP tags. The JSP pages are comparatively easy to maintain than those of servlet because it is possible to separate designing & development in JSP.
- It can perform various operations like accepting input from user thru the forms of web pages, present records from any data source on the server & create dynamic web pages.

* Lifecycle of JSP :-

- There are 4 phases in the life cycle of JSP which are used to process the JSP page. These are shown in fig. below.

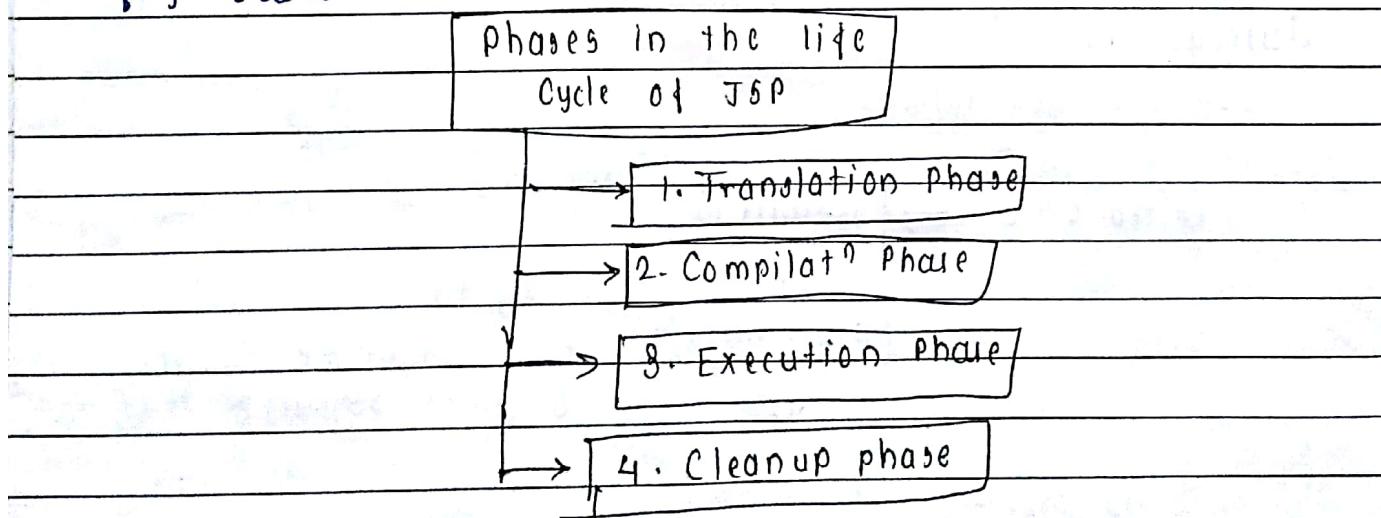


Fig. phases in life cycle of JSP.

1. Translation Phase → At first time when there is a request for JSP page, the JSP page is translated into servlet.

2. Compilation phase -

- Here the JSP engine checks whether the JSP page request is for the 1st time. If yes then the generated servlet is compiled.
- The compilation is done only if the page is requested for the 1st time or any changes have been made in it.

3. Execution phase -

- In this phase the JSP servlet methods are executed.
- a) `public void jspInit()` -
- This method is called when the JSP is initialized before serving any requests. This method is same as of the `init()` method of servlet.
- b) `public void jspService throws ServletException, IOException` -
- This method is basically corresponds to the body of the JSP page. This method is implemented by the JSP container.
 - This method is called repeatedly for each client request.

4. Cleanup phase -

This phase indicates the end of JSP life cycle. In this phase the JSP is destroyed & removed from the container.

c) `jspDestroy()` -

- This method get called when the container is about to destroy the JSP. This method is similar to servlet's `destroy()` method.

JSP life cycle methods

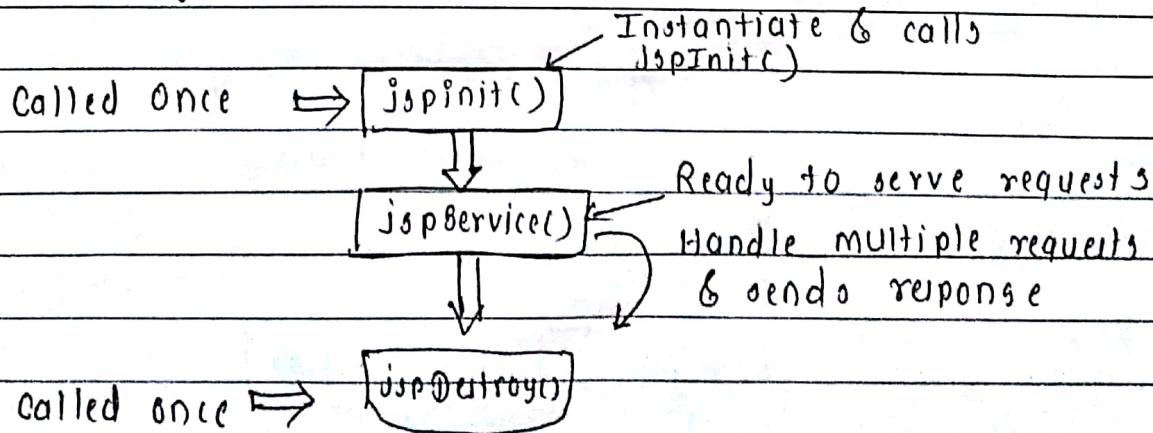


Fig. JSP life cycle method

* Elements of JSP Page -

In JSP there are 3 important types of elements available.

Important types of JSP

1. Scripting Elements

2. Directives

3. Actions & templates

A) Scripting Elements -

- Scripting elements are used to include scripting code within the JSP. These elements help to perform various operations like declaring variables & methods, include implementation code & evaluate an expression.

• Types of Scripting elements -

Types of Scripting elements

A. Declaratn

B. Scriptlets

C. Expressions

D. Comments

Fig. Types of Scripting elements

A) Declaratn -

- The declaratn section is used to declare class-wide variables & methods.
- This is block of Java code in JSP. When the JSP page is initialized all the declarations in the file get initialized.
- These members have class scope means they are accessible within the JSP anywhere to other declarations, expressions or code.

- The declaratⁿ sectⁿ is enclosed in <%! and %>.

b) Scriptlets -

- The scriptlet is a block of java statements. This block is executed when the client request is processed by the JSP.

- The scriptlet generates O/P stream to send to client.

- In a JSP there may be multiple scriptlets as per the requirements.

- The scriptlet sectⁿ is enclosed in <% and %>.

<% Java statements %>

c) Expressions -

- The expressⁿ o/p a value in the response stream which is to be sent to client.

- Expression is considered as shorthand notatⁿ of scriptlet.

- After evaluatⁿ of the expressⁿ, the result is converted into string format & displayed to the client.

- The expressⁿ is enclosed in <%= and %>

d) Comments -

Use = Comments are used to give informative text in the JSP. The comments are ignored by the JSP container.

Syntax -

<%-- This is JSP comment --%>

e) Directives -

The directive in JSP serves as messages & directions to the JSP container. It instructs the container the way to handle some important aspects of the JSP processing.

- The entire structure of the servlet . class is affected by the directives.

Syntax —

`<%@ directive attribute = "value" %>`

* Types of directive tag —

1) The page Directive —

It gives instructions to the JSP container.

These instructions affect the entire current JSP page.

Syntax —

`<%@ page attribute = "value" %>`

2) The include Directive —

The include directive is used to include external files during the translation phase.

Syntax —

`<%@ include file = "relative url" %>`

* JSF (Java Server Faces) :-

- JSF is a Java-based web appn framework intended to simplify development integration of web-based user interfaces.

- Java Server Faces is a standardized display technology, which was formalized in a specification thru the Java Community Process.

- What is JSF -

JSF is a MVC web framework that simplifies the construction of User Interfaces (UI) for server-based apps using reusable UI components in a page.

- JSF enables the reuse & extension of the existing standard UI components .

* Benefits -

- JSF reduces the effort in creating & maintaining apps, which will run on a Java appn server & will render appn UI on to a target client, JSF facilitates web appn development by -

- providing reusable UI components.
- marking easy data transfer b/w UI components
- managing UI state across multiple server requests
- Wiring client-side event to server-side appn code.

* JSF Architecture -

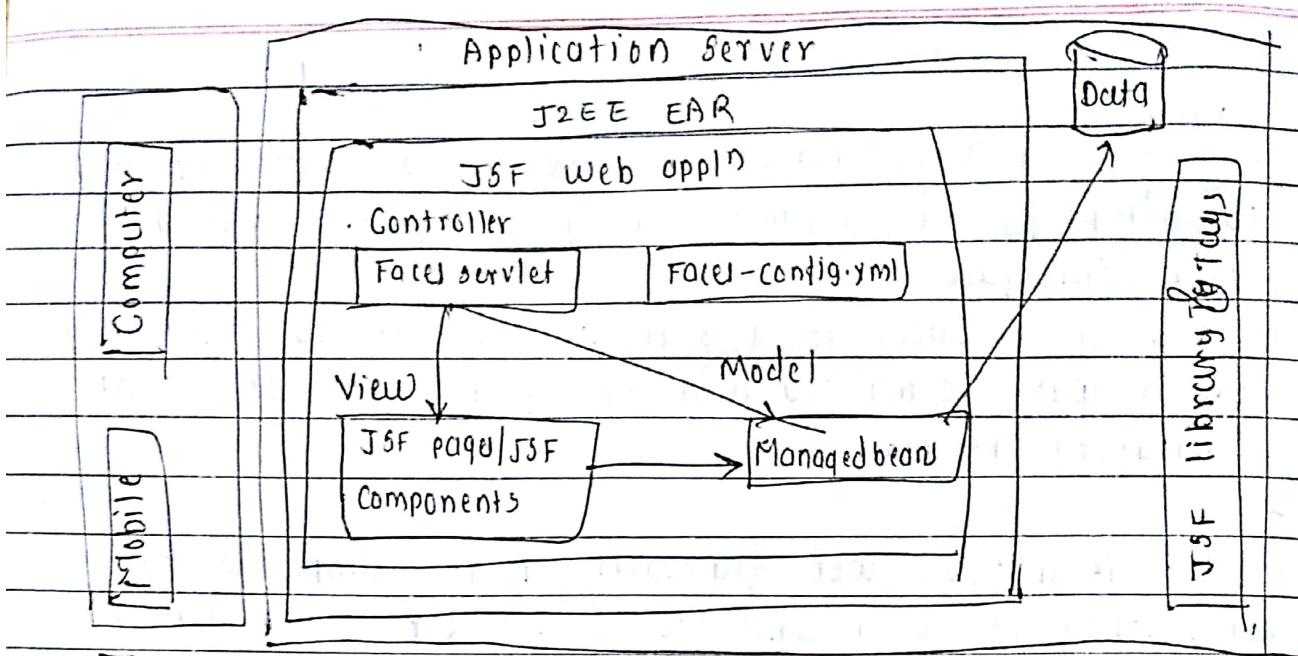
- JSF appn is similar to any other Java technology based web appn; it runs in a Java servlet container, & contains -

i) Java Beans components as models containing appn specific functionality & data.

e) A custom tag library for representing event handlers & validators.

o) A custom tag library for rendering UI components.

4) Server side helper classes.



- There are controllers which can be used to perform user actions. UI can be created by web page authors & business logic can be utilized by managed beans.
- JSF provides several mechanisms for rendering an individual component. It is upto the web page designer to pick the desired representation.

* SOA (Service Oriented Architecture) :-

- The SOA is an architectural design which includes collection of services in a network which communicate with each other.
- The complexity of each service is not noticeable to other services.
- The service is a kind of operation which is well defined, self contained & that provides separate functionality such as checking customer account details, printing bank statements etc & does not depend on the state of other services.

• Why to Use SOA ? -

- SOA is widely used in market which responds quickly & makes effective changes accd to market situations.
- The SOA keeps secret the implementation details of the subsystems.

- It allows interactⁿ of new channels with customers, partners & suppliers.

• Features -

1. SOA uses interfaces which solves the difficult integratⁿ problems in larger system.
2. SOA communicates customers, providers & suppliers with msg messages by using XML schema.
3. It uses the msg monitoring to improve the perform measurement & detects the security attacks.

• Advantages -

1. SOA allows reuse the service of an existing system alternately building the new system.
2. It allows plugging in new services or upgrading existing services to place the new business requirements.
3. It can enhance the performance, functionality of a service & easily makes the system upgrade.

• Disadvantages -

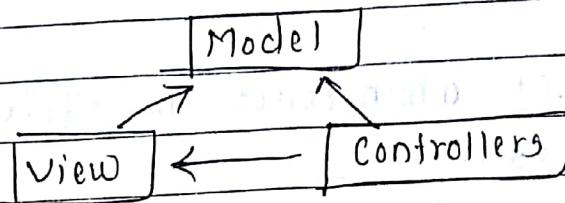
- 1) SOA requires high investment cost
- 2) There is greater overhead when a service interacts with another service. which increases response time.
- 3) SOA is not suitable for GUI appn.

* MVC -

The Model-View-Controller (MVC) is an architectural pattern that separates an appn into 3 main logical components: the model, the view, and the controller.

Each of these components are built to handle specific development aspects of an appn.

MVC Components -



1) Model -

- The model component corresponds to all the data-related logic that the user works with.
- This can represent either the data that is being transferred betn the view & controller components or any other business logic-related data. eg. a customer object will retrieve the customer info. from the db, manipulate it & update its data back to the db or we

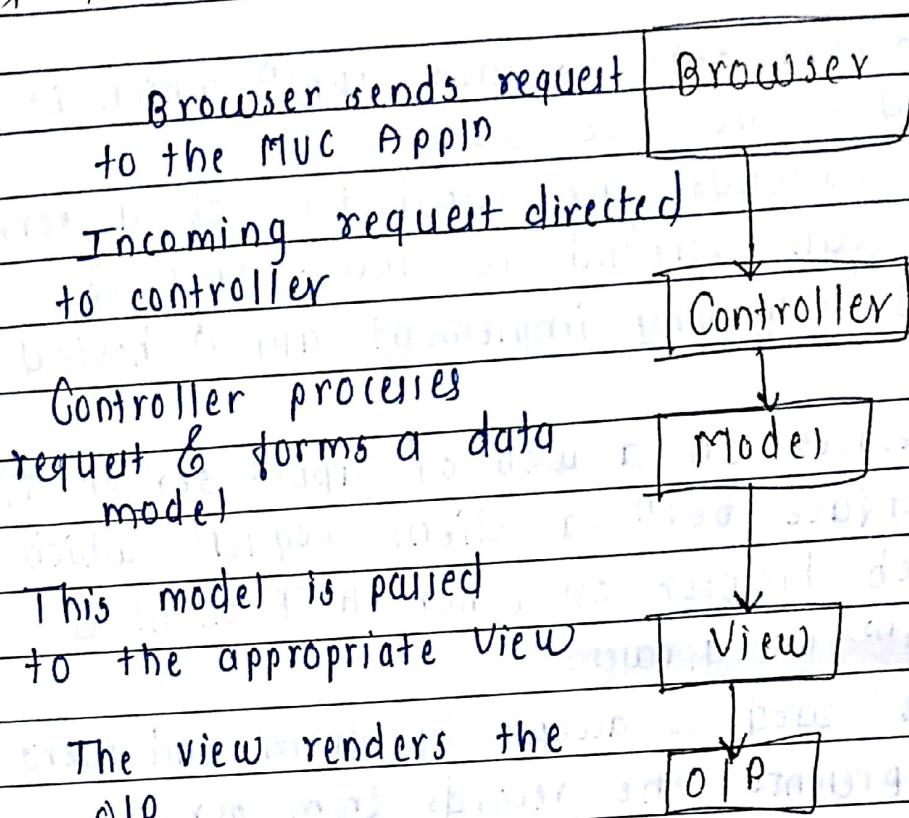
2) View -

- The view component is used for all the UI logic of the appn.
- eg. the customer view will include all the UI components such as text boxes, dropdowns etc. that the final user interacts with.

3) Controller -

- Controller acts as an interface betn Model & View components to process all the business logic & incoming requests, manipulate data using the Model component & interact with the Views to render the final output.

* MVC Flow Diagram -



Flow steps -

- Step 1 - The client browser sends request to the MVC appn.
- Step 2 - Global.asax receives this request & performs routing based on the URL of the incoming request using the RouteTable, RouteData, UriRoutingModule & MvcRouteHandler objects.
- Step 3 - This routing operatⁿ calls the appropriate controller & executes it using the IControllerFactory object & MvcHandler object's Execute method.
- Step 4 - The controller processes the data using Model & invokes the appropriate method using ControllerAction Invoker object.
- Step 5 - The processed model is then passed to the View, which in turn renders the final o/p.

* Java Servlet :-

- Servlet is a server side java appn which is used to create dynamic web pages.
 - A Java servlet extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement apps hosted on Web servers.
 - The servlets executes on a web or appn server. It works as an interface b/w a client request which comes from a Web browser or other HTTP client & server side apps or dbms.
- Use - Servlets are used to accept ip from end users thr the form, presents the records from server database or any other storage media & create dynamic web pages.

- Need of Servlet -
- Servlet is considered as an integral part in J2EE web apps. The server side component of a servlet provides a great mechanism for development of server side web apps.
- It plays significant role in the explosion of Internet with the help of its reusability, performance & scalability.
- We can create server side programs or apps fast & efficiently by using servlets.
- The access to any J2SE & J2EE APIs is available to servlet & it can take advantage of powerful JDK of Java.
- Servlets are the server side scripts which are component based, platform independent which creates the Web based apps.

* Advantages of Servlet

1. Portability - Servlets are written entirely in Java. As Servlets are part of Java, they are portable on the basis of OS & server implementation.

- Servlets follows the Java Rule "Writing Once, Run Anywhere" (WORA), because it is possible to develop a servlet on Windows machine with the help of tomcat server.

- As Servlets are tremendously portable they can be executed in any platform. Hence Servlets are platform independent.

2. Powerful -

- Servlet can handle no. of complex tasks which were difficult for CGI.

for eg. Servlet can directly interact with web server, Servlets can share data or info. among each other. Servlets can easily make db connection pools which can be implemented easily.

3. Efficiency -

- The invocation of Servlet is highly efficient than CGI scripts. When the Servlet is loaded in the server it remains persistent in the memory of the server as a single object instance.

4. Safety -

- As Servlets are part of Java, they inherit the strong type safety of JDK. The automatic garbage collection mechanism & lack of pointers in the Java environment protect the Servlets from the problems of memory management.

5. Integration -

- Servlets are strongly integrated with the server. Server can be used by the servlet to perform various operations.

6. Extensibility -

- The servlet API is designed purposely in such a

that it can be easily extensible. Java is robust, well-designed, object oriented lang. which supports the extended features.

7. Inexpensive -

- Servlet are executed with the help of web servers which are available without any cost for personal as well as commercial purpose.

8. Secure -

Servlets are server side components which run thru the web servers. Servlets can inherit the security of the web server.

9. Performance -

- As compare to CGI, Servlets are relatively very faster because in CGI for every script, a new process is generated.

④ Disadvantages of servlet -

1. The process of servlet designing is complicated & slows down the appn.
2. The complex business logics written make it difficult to understand the servlet.
3. To run the servlets, there is requirement of Java Runtime Environment on the server.

* /Architecture of servlet -

- A servlet is a class, which is implemented from the inbuilt interface javax.servlet.Servlet.

- Classes of Java
There are 2 main classes provided by Java which are extended from javax.servlet.Servlet.

Classes of java -

javax.servlet.Servlet

javax.servlet.GenericServlet

javax.servlet.http.HttpServlet

Fig. Classes of Java

* Architecture of Package -

a) Servlet

Life Cycle of servlet -

- The life cycle of servlet is considered as a series of steps during which a servlet comes across the its life span, beginning from loading process till last step of destroying.
- The lifecycle of servlet is a simple object oriented design.
- Web server - This server is also called as HTTP server. The main functionality of web server is to handle the requests means to accept the client requests & generates the response.
- Web Container - This is also called as Servlet Container or servlet engine.

* Life Cycle of servlet -

• Steps of Life Cycle of servlet -

The LC of servlet undergoes 5 main steps as follows:

Step 1: Load servlet

↓
Step 2: Create servlet instance

↓
Step 3: Call init() - once

↓
Step 4: Call service() - repeatedly for each client request

↓
Step 5: Call destroy() - once

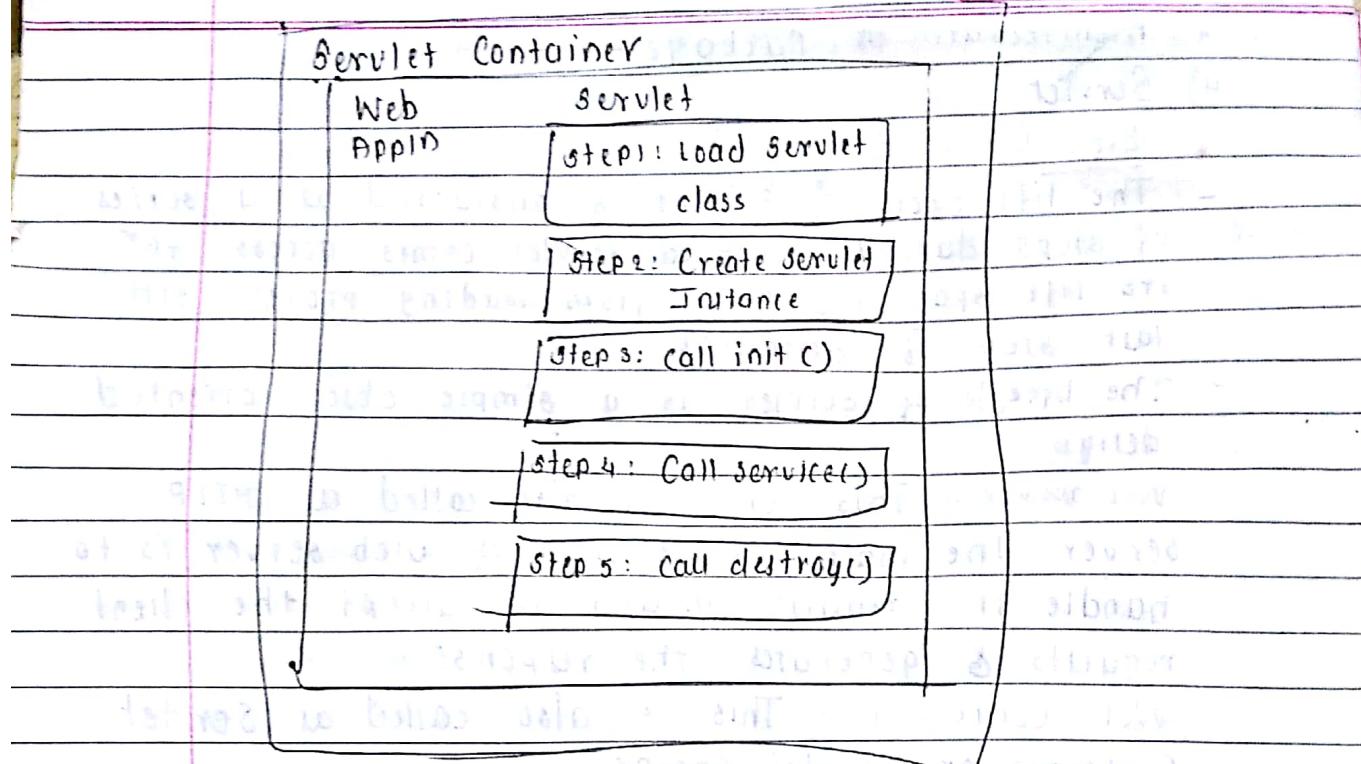


Fig. Life Cycle of servlet.

Step 1: Load servlet

When the service of web server like Apache Tomcat is started, the servlet is deployed & loaded by servlet container.

Before invoking a servlet, the servlet container should load its class defⁿ first.

Step 2: Creating instance of servlet

As soon as the servlet class is loaded, an instance of servlet class is created by the servlet container.

For every servlet, only a single instance is created which handles all the client requests.



Step 3: Call init() method

The init() method is considered as the beginning of servlet's life cycle.

As soon as the servlet class is instantiated, the server calls the init() method for instantiated the servlet.

This method helps to initialize the servlet.

The init() method is called only once in the LC of servlet.



Step 4: Call service() method

- Whenever the web server receives a client request for servlet, it initiates a new thread which invokes service() method. That means it gets called multiple times in the servlet's LC.

Step 5: Call destroy() method

- This method indicates the end of servlet's life. When servlet container shuts down which is generally happens when web server is stopped, it unloads all the running servlets from the memory & calls destroy() method for all the initialized servlets.
- This is where all the resources created by the init() method are cleaned up.

* Struts :-

Defn - Struts is an open source framework which is developed by the "Apache Software Foundation" under the project named as Jakarta.

- Struts framework helps to create web related apps for Java environment with MVC arch.
- In 2004, the framework "struts 1" was released but it failed to fulfill the requirements of users as per current trends.
- "Struts 2" is then developed by Apache with OpenSymphony.
- We say : Struts 2 = Webwork2 + Struts 1
- Webwork2 is the framework developed by OpenSymphony.

Features of struts 2

- struts 2 provides many important features which were not at a part in struts 1.
- The imp. features of struts 2 framework are as follows :

Features of struts 2

1. Configurable MVC components
2. POJO based actions
3. AJAX support
4. Integration
5. Various result types
6. Various Tag support
7. Theme & Template support

Fig. Features of struts 2

1. Configurable MVC Components -

In struts 2 framework, the info of all the components like view & action is given in struts.xml file.

2. POJO based actions -

In struts 2, the action class is a plain old Java Object, shortly called as POJO.

3. AJAX support -

Struts 2 have a strong support to AJAX technology. It helps to send an asynchronous request.

4. Integration support -

The integratn of various frameworks like spring, hibernate, Tiles is very easy in struts appn.

5. Various Result Type -
As result it is possible to use various technologies like JSP, freemarker, & velocity in struts 2 appⁿ.
 6. Various Tag support -
There are no.of tags provided by struts 2 like UI, Data & control tags.
 7. Theme & Template support -
There are 3 types of themes supported in struts 2 appⁿ; XHTML, simple & CSS-XHTML.

Architecture of struts -

- Architecture of struts—
 - The architecture of struts is purely MVC based.
 - There are 5 core components in the MVC architect. of struts 2.

• Architecture Components -

- 1. Actions
 - 2. Interceptors
 - 3. Value stack / OGNL
 - 4. Result / Result types

5. View technologies

- View technologies
 - There are slight diff. betw struts 2 & traditional mvc framework in the manner that the role of model is taken by action instead of controller.
 - The Model, view & controller are implemented as follows:
 - Model - Implemented with actions.

Model) - Implemented with actions.

- View : - i - combination of result types & results.

Controller : a struts 2 dispatch servlet filter & interceptors.

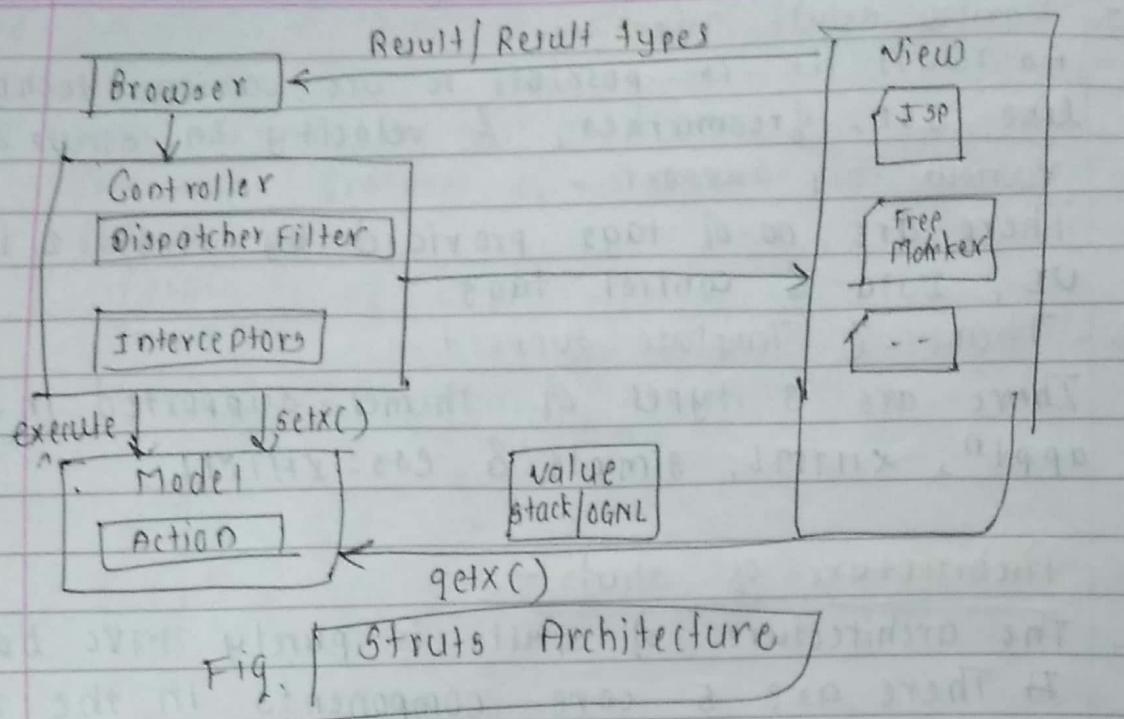
- Request life Cycle -

- Based on the above fig. the request life cycle in struts 2 can be explained as follows:

- Request is sent by user to server for some ~~resource~~ response.
In between interceptors like

- Functionalities of configured interceptors like validation, file upload etc. are applied.

- To carry out the requested operatⁿ, selected actⁿ is implemented.



- Executing an Action of struts :-
- When the HTML form is submitted in a struts 2 web appn, the ip which is sent to java class file named "Actions" is controlled by the controller.
- As soon as the executof "Actions" is completed, a resource is selected by the result to render the response.