

JAVA WEB DEVELOPMENT using JDBC, Servlets & JSP.

Web application - is any software application whose services can be accessed via web.

→ The communication b/w web client & web server happens using the http protocol

Eg → web client → IE, chrome, safari, firefox
web server → Apache, IIS

URL → Uniform resource locator)

static vs Dynamic Web application

Static web application → is a non-interactive web application which provides some generic information no matter which client is accessing so there is no server side programming.

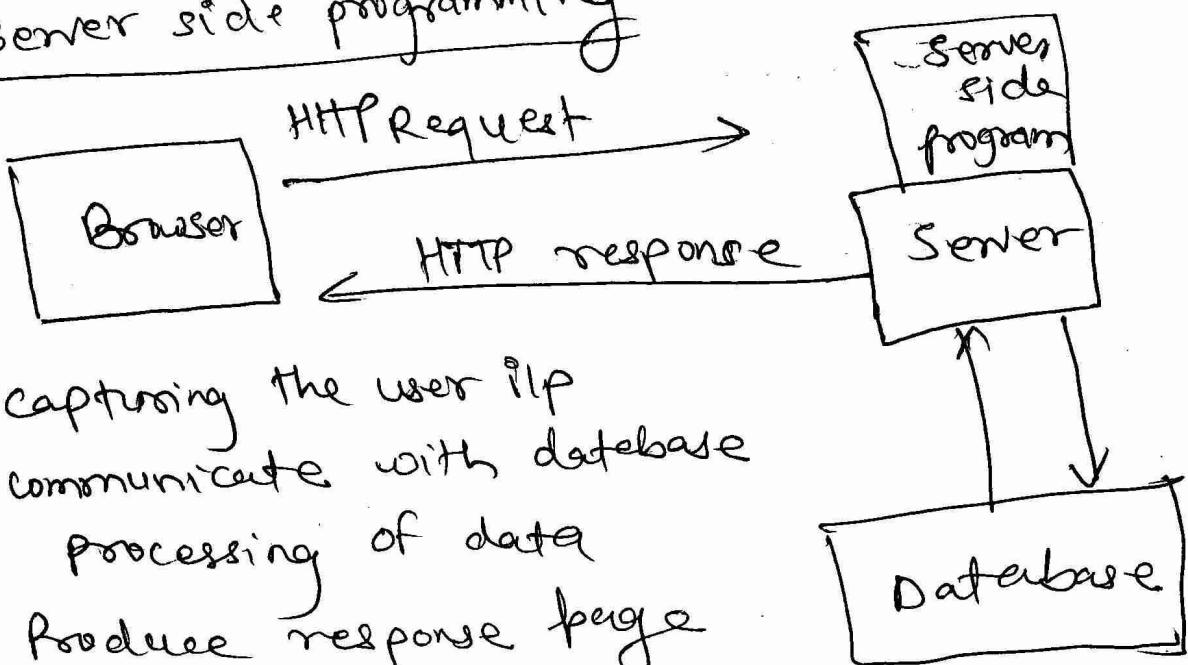
Dynamic web application → comprises of servlets, jsp's & other languages.

web client accesses dynamic web applications?

to update, save, delete information, send dynamic response which it creates back to browser

a dynamic web application runs on a web container not web server

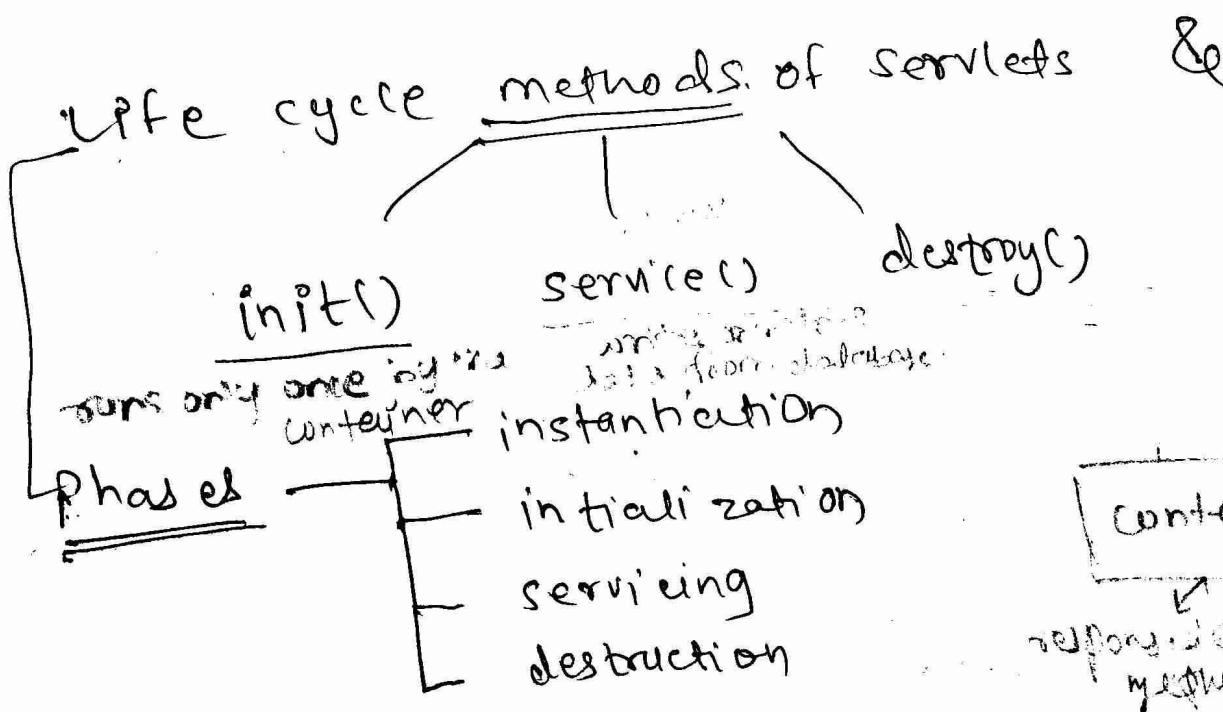
server side programming



capturing the user input
communicate with database
processing of data
produce response page

Handling response page to server

- following is a part of dynamic web application but not static ones.?
 - Server side programming
 - following is not required for static web application
 - Database



Servlets - servlets is a technology in the Java EE standard that allows us to develop dynamic web application using Java.

Servlets

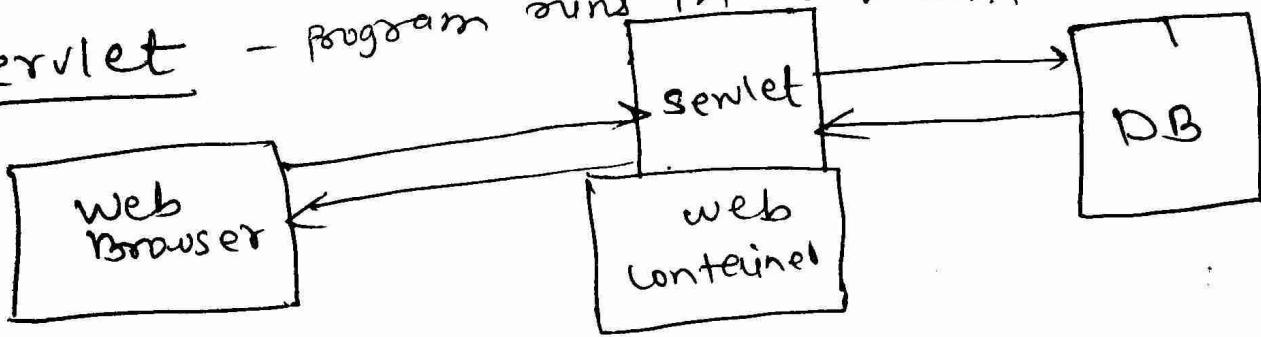
API

Specification

- set of interfaces & classes

- set of rules written in plain english

Servlet - program runs in web container



Servlet Annotations

3.0

- @ web servlet
- @ webInitParam
- @ webFilter
- @ web Listener

Code the Servlet

1st part

Class → package name → superclass (Browser → searchGenericServlet → finish)

Package —

```
import java.io.IOException;
```

```
public class HelloWorldServlet extends HttpServlet {
```

@Override

```
→ public void service (ServletRequest request,  
                     ServletResponse response) throws ServletException,  
                                         IOException {
```

```
    response.setContentType ("text/html");
```

```
    PrintWriter out = response.getWriter();
```

```
    out.println ("html");
```

```
    out.println ("<body>");
```

```
    out.println ("<h1> Hello Servlet World </h1>");
```

```
    out.println ("</body>");
```

```
    out.println ("</html>");
```

g

→ If now go to web.xml file to configure

```
<servlet>
```

```
<servlet-name> HelloServlet </servlet-name>
```

```
<servlet-class> com.bharat.training.servlets.HelloWorldServlet
```

(refractor → rename → or - copy newname →)

```
</servlet-class>
```

```
</servlet>
```

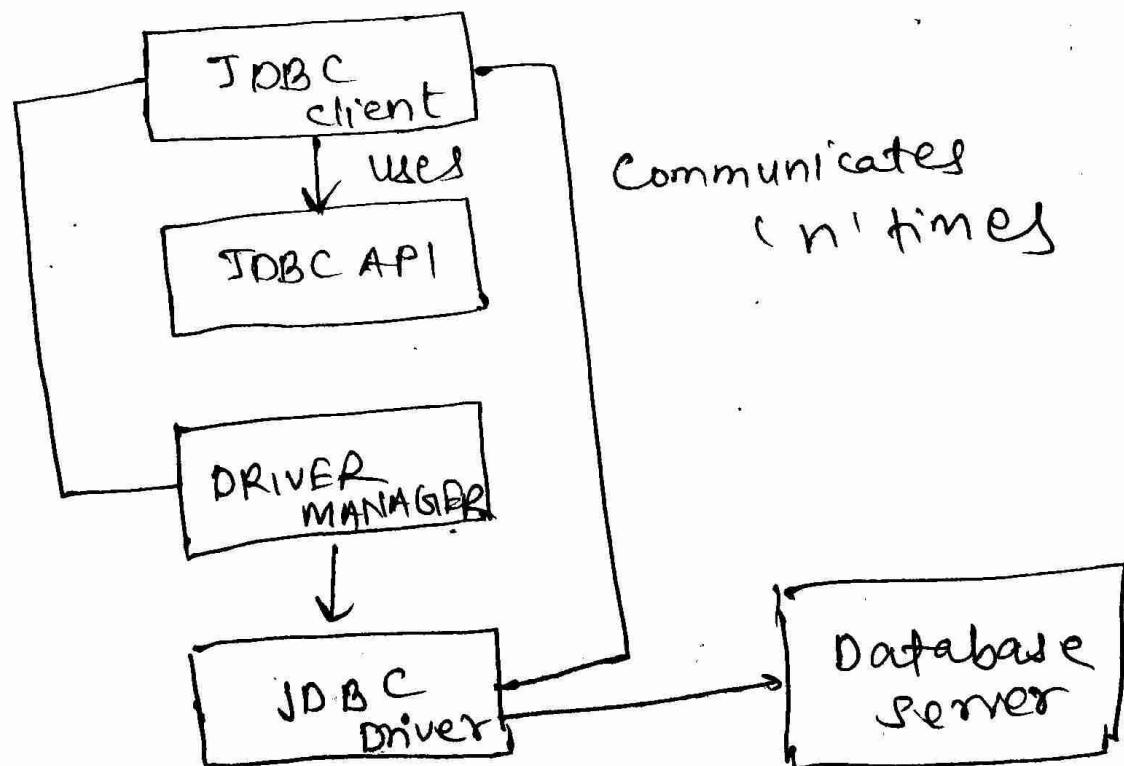
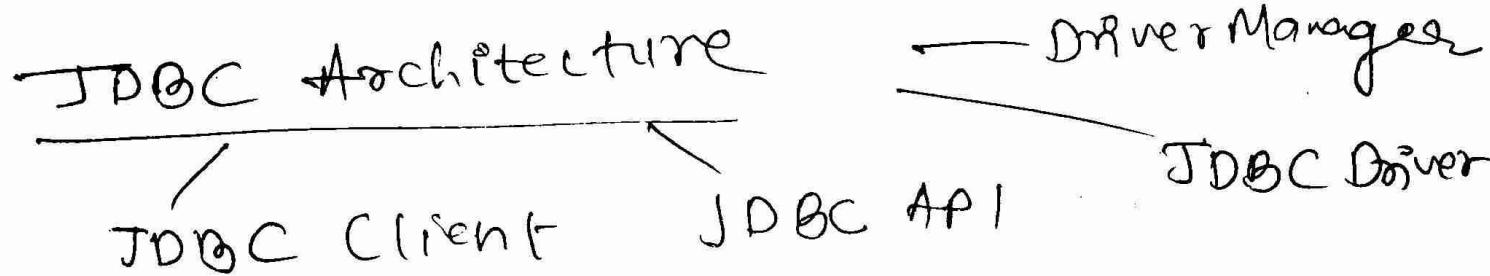
```
<servlet-mapping>
```

```
<servlet-name> HelloServlet </servlet-name>
```

```
<url-pattern> /hello </url-pattern>
```

<RequestDispatcher>

- If a database connection needs to be created which servlet method is good method to put that code in to?
- init
- following is not a parameter Type passed to the servlet's Service method?
- HTTP Session
- following element ~~is~~ can be used in the ~~web~~.xml file to assign a URL to a servlet.
- url-pattern



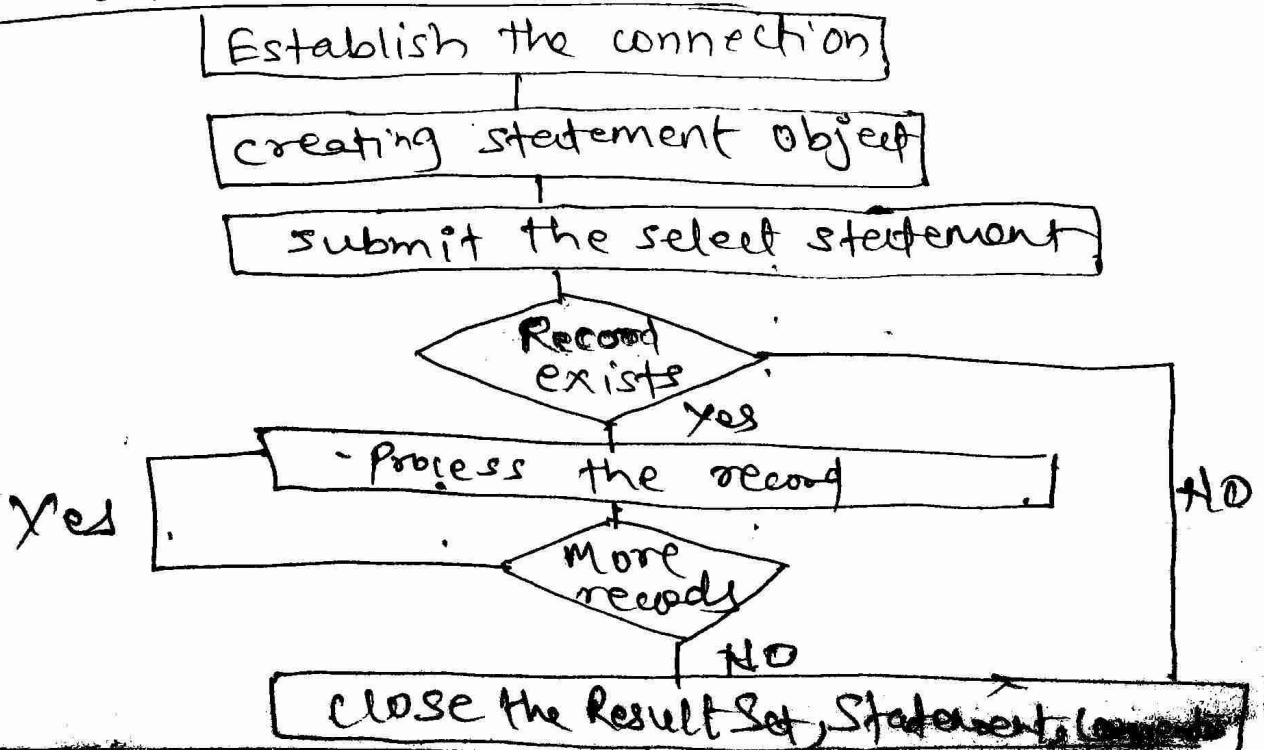
- following is not a part of the JDBC architecture?
 - JDBC server
- which Java package has all the JDBC API classes
 - java.sql
- which component helps us to establish a Connection?
 - Driver Manager

JDBC Statement -

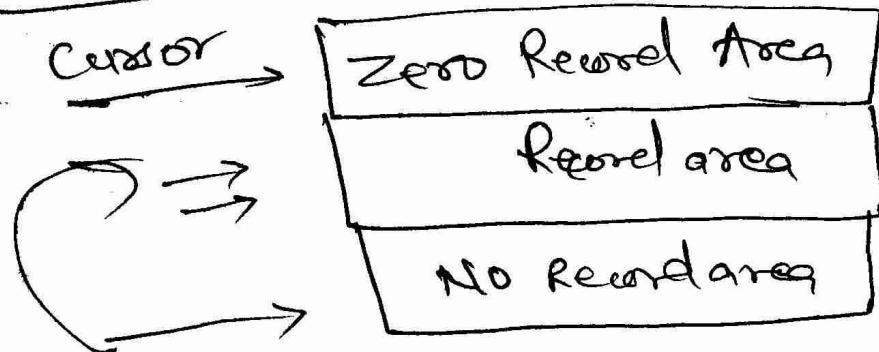
```

Statement stmt = con.createStatement();
int result = stmt.executeUpdate("...");
ResultSet rs = stmt.executeQuery("...");
  
```

JDBC steps to read Data



JDBC ResultSet → object oriented representation of data in the database, of table records



- moves the cursor
- Return Boolean

get ~~xxx~~ ()
(int, char, etc)

Note - DriverManager.getConnection(url, user, password)

- the DriverManager classes getConnection method takes how many parameters?
- 3 (Three)
- which interface can be used to execute sql statements?
- Statement
- which method on the ResultSet allows us to navigate the records?
- next()
- Which version of java has introduced try blocks with resources?
- Java 7.

GET vs Post

GET

- used by default
- has 3 cases → user specified URL
→ user click on hyperlink
→ user submits the form
- used to get the data
- only have headers no body

Every string

No sensitive data should be send

has restriction on the data

we can Idempotent.

i.e we can execute get n no. of time it'll not affect state of data

POST

- it is to be mentioned explicitly.
`<form method="post">`
- meant to posting data
 - Body present
 - payload / Body
 - can be sent safe
 - set any length of data
 - No restriction
- not Idempotent

Note - To create the database connection simply use `DriverManager.getConnection()`

Create static HTML (refer DynamicPro-servlet)

```
protected void doGet(HttpServletRequest request,  
                      HttpServletResponse response) throws ServletException  
, IOException {  
    System.out.println("doPost()");
```

```
try {
    Statement statement = connection.createStatement();
    ResultSet resultSet = statement.executeQuery("select * from user");
    PrintWriter out = response.getWriter();
    out.print("<table>");
    out.print("<tr>");
    out.print("<th>");
    out.print("First Name");
    out.println("</th>");
    out.print("<th>");
    out.print("<td>");
    out.print("Email");
    out.println("</td>");
    out.print("<th>");
    out.print("<tr>");
    out.print("<td>");
    out.print("</td>");
    out.print("</table>");

    catch (SQLException e) {
        e.printStackTrace();
    }
}
```

(This will print time if you want to print again use while loop :-

```
while(resultSet.next()){
    out.println("<td>");
    out.print(resultSet.getString(1));
    out.println("</td>");
    >
```

Init params - <Servlet>

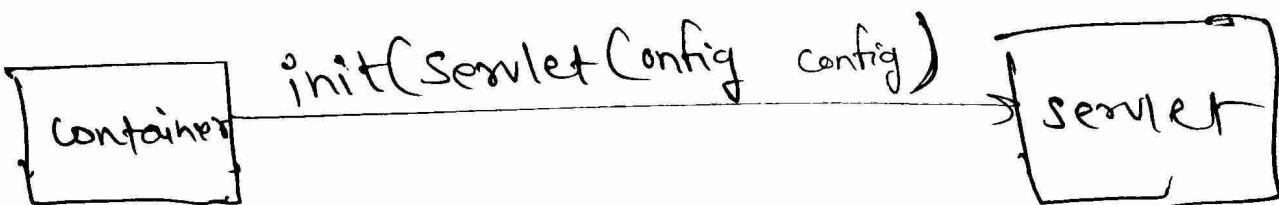
Web.xml

```

< servlet-name >
< servlet-class >
< init-param >
  < param-name > dbuser < /param-name >
  < param-value > root < /param-value >
  < /init-param >

```

Main advantage of init param is you do not have to hard code values.



String User = config.getInitParameter("dbuser");

Init Params

Name	file location
value	c:\Temp
web.xml	& servlet initialization-

configuring Init params using Annotations -

say package com.bharath.user.servlets;

import java.io.IOException;

@WebServlet (urlPatterns = "/addServlet", initParams =

```
{ @WebInitParam(name = "dburl", value = "jdbc:mysql://127.0.0.1/test"),
  @WebInitParam(name = "dbUser", value = "root"),
  @WebInitParam(name = "dbPassword", value = "test") })
```

Public class CreateUserServlet extends HttpServlet {

private
private

ServletConfig uses access to all the Init parameters

```
public void init (ServletConfig config) {
    try {
        System.out.println("init()");
        Class.forName ("com.mysql.jdbc.Driver");
        connection = DriverManager.getConnection(config.getInitParameter ("dbUrl"),
                                              config.getInitParameter ("dbUser"),
                                              config.getInitParameter ("dbPassword"));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

every servlet has its own config Servlet

- which servlet method is a good place to create a database connection?

→ init

- The driver jar file should be copied to which folder under the web application

→ WEB-INF/lib

- following methods one the HttpServlet response obj can be specify the type of data that will be sent in the response?

→ setContentType

- which interface can be used to access init parameters?

→ ServletConfig

- In the web.xml what is the parent element of the init-param element?

→ servlet

- which annotation can be used inside the WebServlet annotation to configure init parameters?

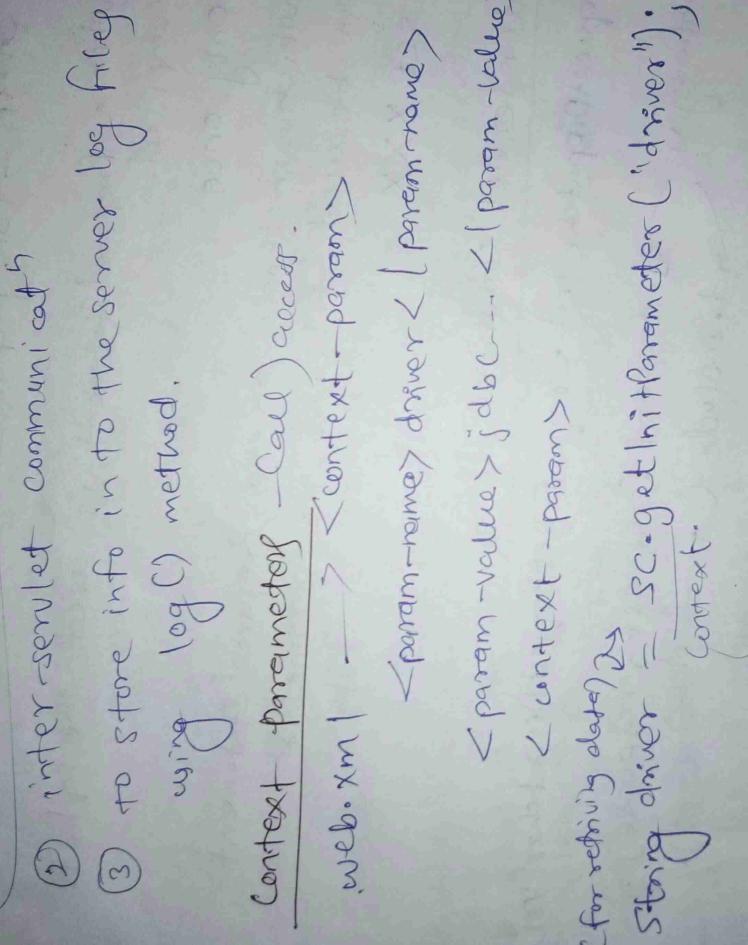
→ WebInitParam

Servlet Context (init method with config)

```
public void init(ServletConfig config) {  
    servletContext context =  
        config.getServletContext();  
}
```

uses of ServletContext

- ① share & manipulate data
setAttribute , getAttribute , removeAttribute()
- ② inter-servlet communication
- ③ to store info into the server log file
using log() method.



try {
 ServletContext context = config.getServletContext();
 context parameters are a child element of
 which element in the web.xml?

→ webapp
using which interface in the Servlet API can we
access the context parameters?

→ ServletContext

- context parameter can be accessed
- through out the applicat)

Prepared Statement

Statement = Con. prepareStatement ("insert --")

- when a PreparedStatement is used the SQL is compiled.
- only once
- which methods on the PreparedStatement can be used to bind the parameters

→ setXXX

Inter servlet communication

RequestDispatcher rd = request.getRequestDispatcher("obj");

forward → rd.forward (request, response);
include → rd.include (request, response);
→ mobile

- following interface in the servlet API can be used to do inter servlet communication.

→ RequestDispatcher

- method not used for inter servlet communication.

→ next

Initialization of servlet the server even a
but if the container initializes the server even a
client request comes in that is called pre-
load on web.xml file
initialization by using
it takes numeric ~~as~~ value as it's data

< servlet >
 < servlet-name > --
 < servlet-class > --
< load-on-startup > 1 < /load-on-startup >

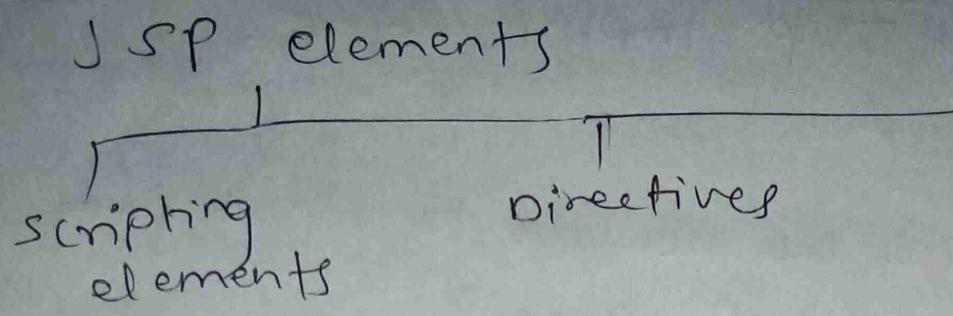
< / servlet >
Example → ~~CXF~~ or Jersey , Spring MVC framework

(the lesser the number the higher priority)
write.
what is the name of the attribute on the
WebServlet annotation that should be used
for pre initialization?

→ LoadOnStartup
which of following element can be used in the
web.xml to configure pre initialization?

→ load-on-startup
→ Listener → Event handling in web application
Request ; session , context , sync

To create class UserCountListener implement HttpSessionListener
distress

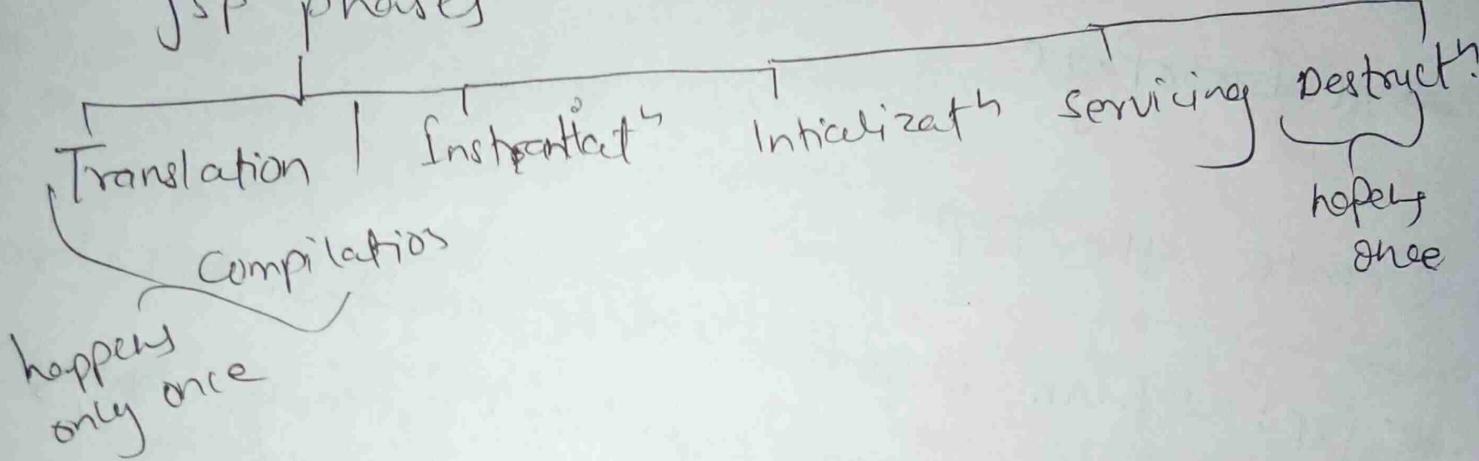


JSP lifecyle

```

    jspInit()
    jspService()
    jspDestroy()
  
```

JSP phases



JSP scripting Elements -

Declaration

`<%!`

`-- --`

`%>`

Expression

`<% =`

`-- --`

`%>`

Scriptlet

`<%`

`-- --`

`%>`

JSP Directives

Include

`<%@ include file = "%>`

Page

taglib