Jsp (Java Server Pages)

List of server side web technologies

=>These are given to develope server side web comps

- servlet ---> from sun Ms/Oracle corp
- jsp ----> from sun Ms /Oracle corp
- asp ----> from MicroSoft
- asp.net --> from MicroSoft
- php ----> from apache
- cold fusion ---> from adobe
- and etc...

What made Sun Ms to create and release jsp (java server pages) when they have already got Servlet Technology?

Ans) In the initial of days to servlets no one like servelt becoz it does not support tag based programming though it is having more features when compare to asp (active server pages).. So it failed to attrat non-java programmers work with servlets.. in the development of web application..

To solve this problem Sun Ms has created and released Jsp (java Server pages) supporting tag based programming and internally uisng servlet programming .So it has attracted both java and non-java programmers..

Initial few days Programmers have used only jsp to develop web applications.. later they started using both servlet and jsp in web application development..

note:: For Every jsp prg/file/comp (.jsp file) internally one equalent Sevlet comp will be generated which is also called JES class (Jsp equalent SErvlet class)

What is the difference b/w Servlet and Jsp?

Servlet	Jsp
a) Does not support tag based programming	a) supports
b) Not suitable for non-java programmers	b) Suitable for both java , non-java programmers
c) Exception handling is mandatory here	c) optional here becoz the jsp equalent servlet (generated internally) will take care of this part
(d) ServletContainer is required to execute Servlet comp	d) servlet container +jsp container is required to execute jsp comp and its equalent Servlet comp
(e) must be placed in private area of the application (WEB-INF/classes)	e) can be placed either in private area or in public area of the web application (outside or inside of WEB-INF folder)
(f) Here we should mixup presentation logic (html code) with b.logic (java code) eg:: pw.println(" hello "); (java code / html code)	(f) Allows to seperate html code from java code... .jsp file: hello (html code); <% int a=10; int b=a*a; %> (scriplet tag having java code)
(g) Placing html code in servlet programming is error prone..	(g) is not error prone..
(h) No Implicit objects (request,response,Servletconfig,Servletcontext and etc.. are not implicit objs they ServletContainer created readymade or built-in objects becoz we need write some code to access them)	h) 9 implicit objs are given .. (these objects can be used directly with out writing any java code to access them)
(i) maping servlet comp with url /url pattern is mandatory	j) mandatory when the jsp comp is placed in private area and optional when the jsp comp is placed in public area.
(j) The modifications done in Servlet comp will reflect only after recompilation of servlet comp and reloading of the web application	(j) The modifications done in web application will reflect dynamically..
(k) Learn curve bith high	k) Learing curve is small.

In standalone Apps we can " this","super" as the implicit objs/ref variables becoz we do not create them(i.e created by jvm) and do not write any code to access them.

In Standalone Apps main(-) String args[] , catch block any Exception obj are not implicit objs becoz unitil we place main(-) and catch block we do not them.. so these are called just called JVM created readymade objects..

Servlet

- ->It is java web technology
- ->version :: 4.0.1 (latest) (compitable with jdk1.8+)
- -> servlet api packages :: javax.servlet, javax.servlet.http, javax.servlet.annotation ,javax.servlet.descriptor
- -> Servletcotnainer is required to exeecute Servlet comps

Jsp

- ->It is java based web technology
- ->version:: 2.2 (latest) (compitable with jdk1.8)
- ->jsp api packages are :: javax.servlet.jsp , javax.servlet.jsp.tagext , javax.servlet.jsp.el
- -> Jsp container +Servlet Container is required for executing jsp comps.

note: Jsp container gives jsp page complier which jsp into an equalent servlet comp.. (JES class)

In Tomcat server

- Servletcontainer name :: CATALINA
- Jsp container name :: JASPER
- Jsp page compiler name :: JspC (org.apache.jasper.JspC)
- <Tomcah_home>\lib folder gives
  - servlet-api.jar (representing servelt api packages)
  - jsp-api.jar (representing jsp api packages)
  - japser.jar (representing jsp container)
  - catalina.jar (representing servlet container)

Jsp programming

- ->gives built-in tags
- ->allows to work with third party tags
- ->allows to develope custom tags

What is the difference b/w html and jsp?

html	jsp
(a) html is client web techology	(a) jsp is server side web technology
(b) html files are static web comps generating static web pages	(b) jsp files are dynamic web comps generating dynamic webpages
(c) html is given by WHATWG and maintained by w3c	(c) jsp is given by Sun Ms (oracle corp)
(d) to execute html code we need html interpeter (part of browser)	(d) To execute jsp code we need jsp container which internally uses Servlet container
(e) html coding is not strictly typed coding (errors will be ignored)	(e) jsp coding is strictly typed coding
(f) html tags and attributes are not case-sensitive	(f) jsp tags and attributes are case-sensitive
(g) Does not allow to place java code in html file	g) allows to place java code in jsp files
(h) does not allow to work with user-defined, third party tags	(h) allows to work with user-defined, third party tags.

w3c :: World Wide Web Consortium
whatwg::Web Hypertext Application Technology Working Group

The List of implicit objs in jsp page

- request,response
- page,pageContext
- config, application
- out , session, exception

The Structure of jsp file/page/comp
==========
=>every jsp file/page/comp should have extension of .jsp
=> In jsp page free style programming is possible..
=>Servlet comp development needs traditional java coding..

<filename>.jsp

```
<  ........
   ....... html code
   ........
    ........>

<%  ....
    ....  java code
    ....
    .....
      %>

    .....
     ... //ordinary text
     .....   (plain text)
<%  ....
    ...  //java code
    ....
    .... %>
```

scriptlet

scriptlet

Template text = html code +plain Text

In our jsp page the template text output is static output where as the java code placed in scriptlet generates dynamic output.

=>jsp pages generate dynamic web pages containing both static and dynamic outputs, The template text of jsp pages gives that static content where as the java code of jsp pages gives dynamic content.

first.jsp (jsp page)

```
<b> first jsp page </b> <br>
<% java.util.Date d=new java.util.Date();
    out.println(d); %>
<br>
end of jsp page
```

template text

java code placed in scriptlet

template text

Dynamic webpage of first.jsp

```
first jsp page
thursday ,20  Aug 2020, IST 11:33 am
end of jsp page
```

scriptlet is a jsp supplied tag that allows to place java code..

Procedure to develop and deploy jsp comps based java web application
==========
step1) create deployment directory structure

```
E:\JspApp1
    |------>WEB-INF  (optional)
    |            |---->web.xml  (optional)
    |--->first.jsp
```

web.xml
==========
<web-app/>

first.jsp
=======
same as above code

step2) Deploye the web application..

copy E:\JspApp1 folder to <tomcat_home>\webapps folder..

step3) start Tomcat server
use <Tomcat_home>\bin\tomcat9.exe file

step4) check the web application.

http://localhost:3030/JspApp1/first.jsp

note:: Since we are placing jsp comp in the public area of the web applications,so its cfg in web.xml file is optional..This time jsp page can be requested using its file name.

=>no need of adding jsp-api.jar file to classpath becoz for jsp page (.jsp file) there is no compilation at command prompt.. but it will be translated into JES class internally and JES class will be compiled using javac and later it will be executed.

How the modifications done jsp page are reflectintg direcly?
Ans) For every modification done in jsp page-->Internally one new JSP equalent Servlet will be generated and this servlet comp class will be instantiated by destructing existing object and new object will be created based the new .class file...So the modifications will reflect automatically

Diffrent types of objects in Jsp programming (Two types objects)
==========
(1) implicit objs
-->these objects are create in JES class automatically.. can be used in any part of jsp page with shpport of scripting tags..
->Jsp givesn 9 implicit objs request,response,page,PageContext ,config,appliation, out, session ,exception

(2) Explicit objs
-->These objs are created by the Programmer manucally..

scriptlet

```
<%
  java.util.Date d=new java.util.Date();
  out.println(d);
%>
```

->out here is inmplicit obj.
->java.util.Date obj(d) is explicit object..

=>Jsp life cycle are called through servlet life cycle methods.. becoz every jsp page is internally an servelt comp , more jsp page execution is nothing internally generated Servlet comp exectution.

Jsp container raises 3 lifecy cyle events and calles life cycle methods according throguh jsp life cycle methods.

a) Instantiation event (raises when container creates JES class object)
calls jspInit()/_jspInit() as life cycle methopds through Servlet life cycle method init(ServletConfig cg).

note:: jspInit() is given for programmer related intiialization logics like creating jdbc con object
_jspInit() is given by container to place related intialization logics with respect to JES class

b) request processing event..
=> raises this event when JES class object ready to process the request...
=> calls _jspService(-,-) of JES cass as life cycle method through servlet life cycle mthods using public service(req,res) and protected service(req,res)..

c) Destruction event ::
==========
-> raises this event when JES class object is about to destroy...
->Cotnainer calls jspDestroy()/_jspDestroy() methods as life cycle ethods through Servletl life cycle method called destroy() method..

jspDestroy() -->To place Programmer related uninitialization logic
eg:: closing jdbc con object
_jspDestroy() -->to place container specific unintialzation logic..

note:: "_"symbol in the JES class indicates..that class names, method names are not generated by the Programmer.. theu are just placed for container..

=>In tomcat web server the JES class for first.jsp of JspApp1 web application will come in E:\Tomcat 9.x\work\Catalina\localhost\JspApp1\org\apache\jsp folder having names first_jsp.java (source code) firrst_jsp.class (compiled code)

package name

Web app name..

=>Every JES class extends from Contaier supplied class and that class extends from HttpServlet (AC)

=> The super class JES class is Container specific i.e will change container to container

=> In case of Tomcat server the super class of JES class is org.apache.jasper.runtime.HttpJspBase

=>By default every JES class contains

a) _jspInit() method b) _jspDestroy() method c) _jspService(-,-) method

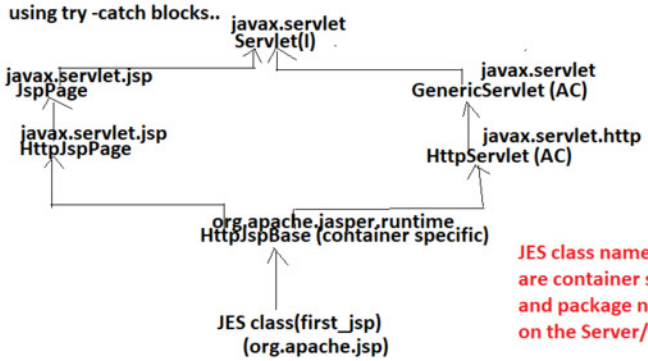
note: jspInit() , jspDestroy() methods will come in JES class only when they are defined by prgrammer in jsp file..

=>The Template text placed in jsp page goes to _jspService(-,-) and becomes the argument values of out.write(-) methods

=> The java code placed in scriptlet goes to _jspService(-,-) as it is.

=> all implicit objs created in _jspService(-,-) method as Local variables..

=> The java code that goes to _jspService(-,-) will automatically takes care of exception handling using try -catch blocks..



JES class name and its super class name are container specific i.e thier names and package names will change based on the Server/Container we use.

=>The super class of JES class contains servlet life cycle methods definitations calling jsp life methods internally.. and also jspInit() ,_jspInit() , jspDestroy(), _jspDestroy() methods with Null Method definitations.. (empty method definitation to comple the flow).

For Instantation event

==========

container calls init(cg) method on JES clas obj -->since not there in JES class the init(cg) method JES super class executes (HttpJspBase) ---> the init(cg) method of JES super class calls jspInit() and _jspInit() methods. --> _jspInit() of JES class and jspInit() method of JES Super calss (HttpJspBase) methods will execute..

For requst processing event

==========

Container calls 1st service(-,-) method ServeltRequest ,ServletResponse objects as args on JES class object--> since not available in JES class it will search all the classes of inheritence hierachy and finds it HttpServlet class (super super class of JES class) --> 1st service(-,-) of HttpServlet class calls 2nd service(-,-) methods and it finds it super class of JES class (HttpJspBase class) and this method interally calls _jspService(-,-) method and _jspService(-,-) of JES class executes..

For destruction event

==========

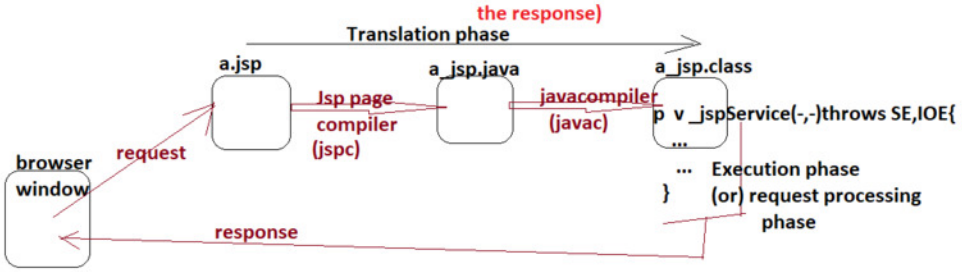
Container calls destroy() method on JES class obj ---> since not available in JES class ---> the destroy() method of JES super class will execute --> and that internally calls jspDestory() and _jspDestroy() methods ---> jspDestroy() of JES super class executes and _jspDestroy() method of JES class will execute..

Two phases of Jsp execution

==========

a) Translation phase (Translates jsp into an equalent Servlet comp source code and byte code)

b) Execution phase/request proccesing phase (the _jspService(-,-) method of JES class will execute to process the request and to generate the response)

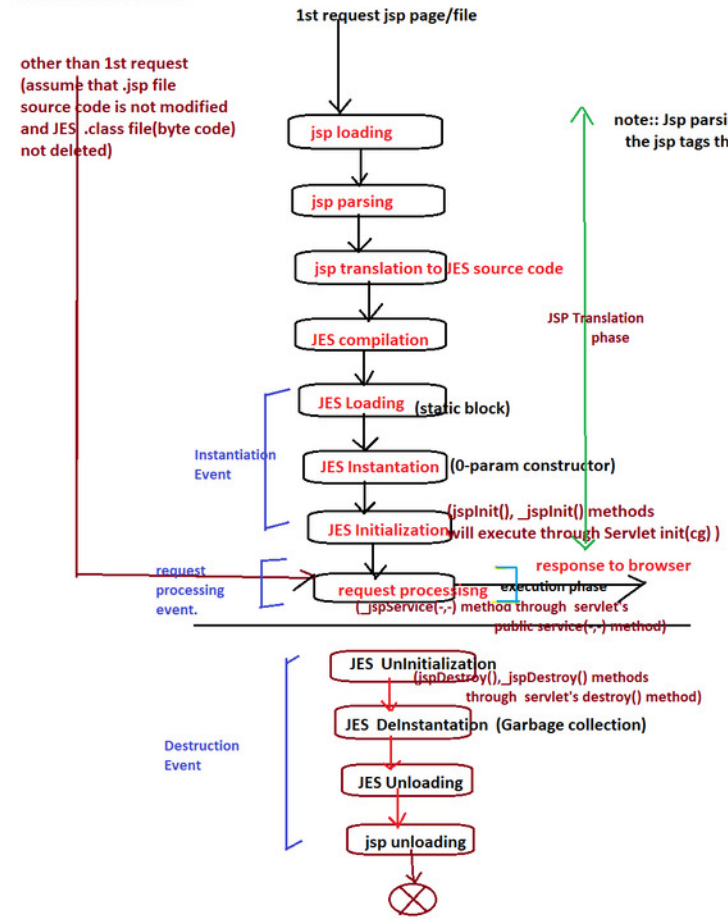


=>The request given to jsp page participates directly request processing phase/execution phase if the source code jsp page is not modified before the request and the byte code JES class is already availble otherwse the request given jsp page first participates translation and later particpated in request processing/execution phase.

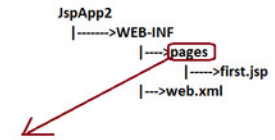
note:: if we just delete soruce file of JES class.. then also next request directly particpates in execution phase.

Jsp life cycle diagram

==========



=>if jsp page is placed in private area of web application (WEB-INF and its sub folders)then jsp file cfg in web.xml file is mandatory having url pattern)




```
web.xml
==========
 <web-app>
   <servlet>
      <servlet-name>f</servlet-name>
      <jsp-file>/WEB-INF/pages/first.jsp </jsp-file>
    </servlet>
   <servlet-mapping>
       <servlet-name>f</servlet-name>
        <url-pattern>/firsturl</url-pattern>
    </servlet-mapping>
 </web-app>
```

Jsp file configuration

mapping jsp file with url /url pattern

http://localhost:3030/JspApp2/WEB-INF/pages/first.jsp (invalid)

http://localhost:3030/JspApp2/firsturl (valid)

note:: Once the jsp is cfg in web.xml file the container automatically enables <load-on-startup>

This <load-on-startup> makes Container to perform jsp loading, jsp parsing, jsp transaltion, jES compilation JES Loading ,JES instanatiation & JES Initialization activities either during server startup or during the deployment of the web application.

Q) Can we cfg url pattern for the jsp page that is there in public area ?

Ans) Yes --But not required..

Q) Can we cfg multiple url patterns for jsp page?

```
<servlet-mapping>
    <servlet-name>f</servlet-name>
    <url-pattern>/firsturl</url-pattern>
    <url-pattern>/firsturl1</url-pattern>
</servlet-mapping>
```

@) Can we cfg jsp page /file using Annotations?

Ans) Not Possbile .. In Jsp page there will be no classe definitations to write @WebServlet

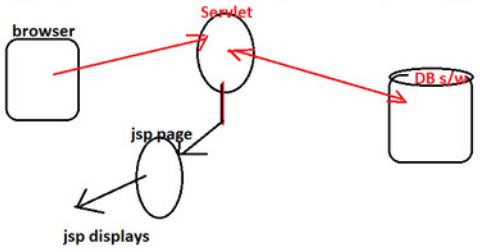
What is the advnatage of placing jsp page in private area?

Ans)=> To protect source code access from outsiders web application or web server (Indirectly from endusers)

=>Useful to hide the technology of web application from enduser...

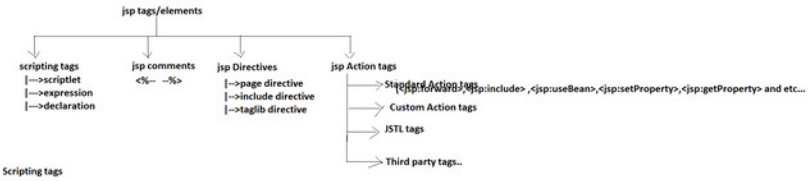
=> To get automatic <load-on-startup> advantages (we should cfg in web.xml)

=> if jsp page displaying request scope data collected servlet comp then direct request to jsp page will display "null" (ugly) values ,To avoid this ugly values keep jsp in private area.. So no one can give direct request to jsp page..



Q) if i directly modify the source code of JES class (with /with out compilation) can u tell me what happens?

Tags /elements in jsp
==========



Scripting tags
==========
=>These tags are given to place java code in jsp page
=>The java code placed in jsp tags is called script code .. So the tags are called scripting tags

a) scriptlet (<% .... %>)
b) expression (<%= .... %>)
c) declaration (<%! ... %>)

scriptlet
========
=>This tag is given to place java code in jsp page..
=> The java code placed in scriptlet goes to _jspService(-,-) of JES class as it is .

standard syntax
==========
<% ....
.... %>

xml syntax
==========
<jsp:scriptlet>
...
</jsp:scriptlet>

=>The variables declared in scriplet becomes the local variables of
_jspService(-,-) method in JES class

first.jsp
========
<% int a=10;
out.println("square::"+(a*a));
%>

In JES class
==========
public class first_jsp extends ....{

p v _jspService(req, res) {
... //implicit objs
...
int a=10;
out.println("square::"+(a*a));
}
}

In scriptlets we can use the implicit objs becoz they are also coming as the local variables in _jspService(-,-) method of JES class

first.jsp
========
<%
out.println("browsr name::"
+request.getHader("user-agent"));
%>

first_jsp.java
==========
public class first_jsp extends .....{

p v _jspService(req,res)throws SE,IOE{
....
... //implicit obj
...
out.println("browsr name::"
+request.getHader("user-agent"));
}
}

=>In scriptlet , we can call methods .. but we can not define methods.. as java does not support nested method definitations

first.jsp
=======
<% public void m1(){
} %>
gives error

In JES class

p v _jspService(req,res)throws SE,IOE{
... //implicit objs
...
public void m1(){
}
}
Error as java does not support nested method definitation.

class inside class :: yes
class inside interface ::yes
interface inside interface:: yes
interface inside class ::yes

class inside method :: yes (Local inner class)
interface inside method :: no

first.jsp
======
<% class Test{
} %>

In JES class

p v _jspService(req ,res){
... //implicit objs
...
class Test{
}
}

first.jsp
======
interface
<% Test{
} %>

In JES class

p v _jspService(req ,res){
... //implicit objs
...
interface Test{
}
}

=>java supports local inner classes ,but java does not supprt local inner interfaces..

=>In real projects _jspService(-,-) method maintains b.logic or request processing .. So we can plce b.logic or request procesing in the scripet of jsp programming.

Using xml syntax
==========
<jsp:scriptlet>
int x=10;
int y=20;
int z=x+y;
out.println(z);
</jsp:scripet>

In JES class
==========
p v _jspService(req,res)throws SE,IOE{
...
... //implicit objs
...
int x=10;
int y=20;
int z=x+y;
out.println(z);
}

we should work with "<" symbol effectively while dealing with xml systax based scriptlet becoz we use at as java operator .. becoz of xml sytax .. it will be take them begining of xml tags sytax

Problem::
<jsp:scriptlet>
int a=10;
int b=20;
boolean flag=a<b;
out.println("result::"+flag);
</jsp:scriptlet>
gives problem.

1
solution:: work with standads syntax::

<%
int a=10;
int b=20;
boolean flag=a<b;
out.println("result::"+flag);
%>

Xml syntax will tags also be transalted to standard syntax, So it better work with stadard syntax.

solution2:: (recomanded)

<jsp:scriptlet>
<![CDATA[
int a=10;
int b=20;
boolean flag=a<b;
out.println("result::"+flag);
]]>
</jsp:scriptlet>

note:: <![CDATA[..]]> makes xml parser/xml reader to read given body as it is with out applying any xml meaning.. on it.. (asks parser to create given get body as text content.

What is diff b/w out.write(-) and out.print(-) methods?

Ans) out.write(-) can not display "null" value... so JES uses it to display temlate text content on the browser..
out.print(-) can display "null" values... so JES uses it to display java code results on the browser which may have null values..

<%
String s=null;
out.print(s); //display null
out.write(s); // throws exception
%>

=>In one jsp page we can place multiple scriptlets .. having both xml , standard syntaxes..

Expression tag

=>this is given to evaluate given expression and to display results on to the browser..
=> arithmetic operation ,logical operation, method call, instantiation and etc.. falls under expresions..

standard syn::
```
<%= <expression> %>
```

xml syntax::
```
<jsp:expression>
    <expression>
</jsp:expression>
```
in

=>The code placed expression tag automatically goes to _jspService(-,-) of JES class and becomes the argument value of out.print(-) method

first.jsp
```
<% int a=10; %>
a value :: <%=a %>  <br>
square value :: <%=a*a %> <br>
is a =10 ? :: <%= (a==10) %>
```

JES class
```
public class first_jsp extends ....{

  p  v  _jspService(req,res)throws SE,IOE{
       ....
       ....  //implicit objs
       ...
    int a=10;
    out.write("a value ::");
    out.print(a);
    out.write("<br>");
    out.write("square value ::");
    out.print(a*a);
    out.write("<br>")
    out.write("is a=10?");
    out.print( (a==10));
   }
}
```

In expression tag we can use implicit objects.. becoz the implicit objects are local variables in _jspService(-,-) method ang the code expression tag also goes there..

first.jsp
```
browser name::  <%=request.getHeader("user-agent") %>
```

note:: request headers carry more info about client/browser along the with request having fixed header names like user-agent, referer,accept,accept-language and etc.. fixed names

user-agent :: hold browser software name
referer :: current request url
accept :: holds mime types supported by browser s/w
accept-language :: holds languages supported by the browser and etc...

In JES class
```
p   v    _jspService(req,res)throws SE,IOE{
   ...
   ...
  out.write("browser name::");
  out.print(request.getHeader("user-agent"));
  }
```

What is diff among request parameters , request headers and request attributes ?

req parameters	request headers	request attributes
a)useful to gather enduser supplied inputs like from data/query string data being from servlet comp	a)useful to gather more details about client machine its browser s/w being from servlet comp	a) useful to pass additional data from one web web comp to another web comp when they are using same request object.
b) request param names and values are user-defined (i.e not fixed)	b) request header names are fixed but values will be changed based on Client machine its browser s/w..	b) request attribute names and values are programmer choice (not fixed-user-defined)
c)Servletcontainer puts request parameters into request object automatically	c) ServletContainer keeps request headers into request obj automatically	c) programmers keepts/creates request attributes by calling req.setAttribute(-) methods
(d) To read req param values use req.getParameter(-) method	(d) to Rea.. req header values use req.getHeader(-) method	(d) To read request attribute values use req.getAttribute(-)
(e) once the req param is added to request , it can not be modified,deleted	(e) once the req header is added to request , it can not be modified,deleted	(e) once the req attribute is added to request , it can be modified and deleted..
(f)optional in the request	(f) mandatory in the request	(f) optional in the request

Using expression tag we can not define the method.. but we can call the method (that method should have other than void as the return type)

first.jsp
```
<% String s="hello";  %>
<%=s%>  value length is :: <%=s.length() %>
```

note:: do not place ";" at the end fo expressions that we place in expression tag.

In JES class
```
p   v    _jspService(req,res)throws SE,IOE{
     .....
     .....
     String s="hello";
     out.print(s);
     out.write(" value length is ::")
     out.print(s.length());
  }
}
```

first.jsp
```
current in milli seconds<%= System.currentTimeMillis() %>
```

In JES class
```
p  v  _jspService(req,res)throws SE,IOE{
     ....
     .....
   out.write("current time milliseconds");
   out.print(System.currentTimeMillis());
  }
```

=>we can use expression tag for instantation and to display initial data of the object/instance on to the browser..

first.jsp
```
system date and time ::
    <%=new java.util.Date() %>
```

In JES class
```
p  v  _jsp(req,res)throws SE,IOE
{
 .....
 .....
 out.write("system date and time");
 out.print(new java.util.Date());
}
```

note:: if we use expression tag effectiverly, then there is no need of using out.println(-) /print(-) methods in our total jsp programming..

note:: we can not define methods , classes ,interfaces ,enums ,annotaions using expression tag...

Using xml sytax

first.jsp
```
<jsp:scriptlet>
     int a=10;
     int b=20;
</jsp:scriptlet>
sum is :: <jsp:expresion> a+b </jsp:expression>
```

In JES class
```
p   v   _jspService(req,res)throws SE,IOE{
     ....
     ....
     int a=10;
     int b=20;
     out.write("sum is ::");
     out.println(a+b);
  }
```

Xml syntax based expression tag gives problem with "<" symbol and we can not solve that problem even using <![CDATA[..]]>

```
<jsp:scriptlet>
   int a=10;
     int b=20;
</jsp:scriptlet>
 a < b   ? <jsp:expression> <![CDATA[ a <b  ]]></jsp:expression>
```

<jsp:expression> a<b </jsp:expression> gives error

<%= a<b %>

The "<" symbol problem at template text can be solved by < entity of html file.. but in jsp it can be solved by using standard syntax , not by using xml syntax..

it is always recomanded to evaluate multiple expressions using multuple exression tags..

```
<% int a=10; int b=20; %>
  sum  :: <%=a+b %> <br>
  sub  :: <%=a-b %>
```

Good practice..

The wrost , non-recomanded alternate is ::
```
sum,sub are :: <%=(a+b) +", "+(a-b) %>
```

note:: we can have 0 or more expression tags in our jsp page having both standard , xml syntaxes...

Declaration tag

=> The code placed in this tag goes to outside of _jspService(-,-) method in JES clas
=> This tag is useful to declare global variables , to define methods and to define jspInit() ,jspDestory() methods..

standard syn:

```
<%!
   ....
   ....
%>
```

xml syntax:

```
<jsp:declaration>
   ....
   ....
</jsp:declaration>
```

=>Varaibles declared in in delaration become global variables in JES class

first.jsp

```
<%! int a=10; %>

square value :: <%=a*a %>
```

In JES class

```
public class first_jsp exstends ....{
  int a=10;

  p v _jspService(-,-)throws SE,IOE{
    ...
    ... //implicit objs
    ...
    out.write("square value");
    out.print(a*a);
  }
}
```

What is the diff b/w the variable declared in declaration tag and the variable declared in scriptlet?

=>Declaration tag variables acts as global variables in JES class.. (class level property)
where scriplet tag variable acts as local variable in _jspService(-,-) method..

How to differentiate variable names in scriptlet , if the declaration tag variable name is matching with scriptlet tag variable name?

ans) we can either "this" operator or "page" implicit obj for differentiation

Not recomanded becoz type casting with Server specific class name is required..

first.jsp

```
<%! int a=10; %> //global
<% int a=20; %> //local
 global variable value :: <%=this.a %> <br>
 global variable value :: <%=((first_jsp)page).a %> <br>
 local variable value :: <%=a %>
```

final java.lang.Object page = this;
(in jes class)

Can use implicit objs of jsp in declaration tag code?

Ans) not possible becoz they are not visible. all implicit objs are local variables in _jspService(-,-) method, so they are not visible in declaration tag code that goes outside of _jspService(-,-) method.

```
<%! String brname=request.getHeader("user-agent"); %>
browser name:: <%=brname %>
```

org.apache.jasper.JasperException: Unable to compile class for JSP:
An error occurred at line: [2] in the jsp file: [/first.jsp]
request cannot be resolved

we can use declaration tag for method definations and we can call these either using scriplet or using expression tags -only when the return type is other than void

first.jsp

```
<%! public int sum(int a,int b){
      return a+b;
    }
%>

result is :: <%=sum(10,20) %>
```

JES class

```
public class first_jsp extends ....{

  public int sum(int a,int b){
    return a+b;
  }

  p v _jspService(req,res)throws SE,IOE{
    ...
    ...
    ...
    out.write("result is ::");
    out.print(sum(10,20));
  }
}
```

=>The method call that we place in expression tag must not have void as the return type other wise exception will be thrown..

```
<%=Thread.currentThread().start() %>
```

return type is void , So
org.apache.jasper.JasperException: Unable to compile class for JSP:
An error occurred at line: [8] in the jsp file: [/first.jsp]
The method print(boolean) in the type JspWriter is not applicable for the arguments (void)

Xml syntax of declaration tag is having "<" symbol problem and we can solve that problem by using <![CDATA ....]]>

```
<jsp:declaration>
 public String findBig(int a,int b){
   <![CDATA[
      if(a<b)
        return b+" is big";
      else if(b<a)
        return a+" is big";
      else
        return "both are equal";
   ]]>
 }
</jsp:declaration>

result :: <%=findBig(10,20) %>
```

Problem is only with "<" symbol not with ">" symbol.

Programmer can use declaration to place jspInit() and jspDestory() method definitations having programmer supplied initialization , unintialization logics..

first.jsp

```
<%! public void jspInit(){
      System.out.println("jspInit()");
} %>

<% int a=10;
      System.out.println("_jspService(-,-) method");
%>
square value :: <%=a*a %>1

<%! public void jspDestroy(){
      System.out.println("jspDestroy()");
} %>
```

=>jspInit() method contains programmer specific initialization logics like creating jdbc con object and etc..
=>jspDestroy() method contains programmer specific uninitialization logics like closing jdbc con object and etc..

Code Demonistrating Jsp life cycle activities

```
<%!
static {
      System.out.println("Static block");
}

public first_jsp(){
      System.out.println("0-param constructor");
}
%>

<%! public void jspInit(){
      System.out.println("jspInit()");
} %>

<% int a=10;
      System.out.println("_jspService(-,-) method");
%>
square value :: <%=a*a %>

<%! public void jspDestroy(){
      System.out.println("jspDestroy()");
} %>
```

(for JES class Loading)
(for JES instantiation)
(for JES initialization)
(for request processing)
(for JES uninitialization)

note:: We can place method , class , internface definitations in declaration tags.
(makes it as inner class) (makes it inner innerface)

note:: In one jsp page we can multiple declaration tags having both xml, standard Syntaxes...

Can we place _jspInit() , _jspService(-,-) and _jspDestroy() method definitations in the delaration tag of jsp page?

Ans) Since same are already available in JES class.. So our methods (our _jspXxx()) become duplicate methods in JES class.. Java does not support duplicate methods .. but it supports overrloaded methods..

```
<%! public void _jspInit(){
        ...
    } %>
```

error...

Can we place servlet life cycle method definitations in the declaration tags of jsp page?

Ans) no .. becoz all servlet life cycle methods are given as final methods in the super of JES class (HttpJspBase in case of Tomcat server) and we can not override final methods in the sub classes...

=>servlet life cycle methods placed declaration goes JES class becomes overriding methods of JES super class final method.. which is an error..

=>In the supper class JES class serlet life cycle methods are intentionally given as final methods , So that no developer will think about using servlet life cycle methods directly.. with out using jsp life cycle methods...

```
<%! public init(ServletConfig cg){
        ...
    } %>
```

error...

note:: we can use all the 3 scripting tags in one jsp page in any order having either standard or xml syntax or both syntaxes...

Example App that uses all 3 scripting tags togather in single jsp page

3 scripting tags are
a) scriptlet
b) expression
c) declaration tag.

second.jsp

```
<%! public String generateWishMessage(String user){
      //get System date and time
            java.util.Calendar cal=java.util.Calendar.getInstance();
            //get current hour of the day
     int hour=cal.get(java.util.Calendar.HOUR_OF_DAY);
            //generate wish message
            if(hour<12)
                return "Good Morning::"+user;
            else if(hour<16)
                return "Good afternoon::"+user;
            else if(hour<20)
          return "Good evening::"+user;
            else
         return "Good night::"+user;
   }
          %>
```

Declaration tag

Template text

```
<h1> welcome to jsp page </h1>
<h2> date and time :: <%=new java.util.Date()%> </h2>
```

template text

expresssion tag

scriptlet

```
<% String uname=request.getParameter("uname"); %> <br>
<b> wish Message is ::</b> <%=generateWishMessage(uname) %>
```

(Template text)

expression tag

request url :: http://localhost:3030/JspApp1/second.jsp?uname=raja

Procedure to develope jsp page based web application using eclipse

step1) create dynamic web project
file -->new -->Dynamic webproj --> name: ... , ...

```
JspApp3 (EDWP)
  |--->webcontent
        |---->second.jsp
        |---->WEB-INF
              |--->web.xml
```

note:: if file menu -->new is not showing Dynamic WebPRoject option.. then
a) try to change project prospective to Java EE

b) Install Enterprise java developer tool plugin .. usisng eclipse market place...

step2) make sure that Tomcat server is cfg with eclipse IDE..

step3) develop second.jsp and web.xml files...

step4) Run the web application...

step5) Rigght click on the Project ---> run as ---> choose server ....

note:: if we create Dynamic webProject after configuring server to the IDE.. then there is no need of adding servlet-api.jar file to CLASSPATH/BuildPATH..otherwise we need to add servlet-api.jar file to BUILDPATH/CLASSPATH .. explicitly..

note:: Eclipse IDEs uses its own copy of Tomcat server in the workspace folder .. The JES class for given jsp page will be generated in the following place of workspace folder...

G:\Worskpaces\advjava\NTAJ1113\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\work\Catalina\localhost\JspApp3\org\apache\jsp

web application name

pkg name

|--->second_jsp.java (JES source code)
|--->second_jsp.class (JES compiled code)

Comments in jsp page
==========

A jsp page can have 3 types of comments

a) html comments/xml comments (<!-- .... -->)
->To commnent template text and xml syntax based jsp tags of jsp page
--> these comments are recognized and processed by html interpeter

b) jsp comments (<% -- ..... --%>
->To comment standard syntax jsp tags of jsp page
->these comments are recognized and processed by jsp page compiler

c) java commnets (// -single line , /* .... */ - multiline)
|--->To comment java code of scripting tags
|--->These comments are recognized and processed by java compiler...

=>html commnets of jsp page are called output comments becoz they come to browser along with the response code/output code.
=>jsp commnets are called hidden commnents becoz they are visible only in the jsp page, not in other phases of jsp execution...
=>java comments are called scripting commnets becoz they are useful to comment script code placed in scrpting tags of jsp page..

Comment ========	Visibility: In JES source code	In JES byte code	In output code goes to browser	output (webpage on the browser)
jsp comment (<%- .... --%>	no	no	no	no
html comments <!-- ..... -->	yes	yes	yes	no
java comments // or /*... */	yes	no	no	no

Can we comment jsp tags with html comments?
Ans) Possbile but not recomanded ...becoz it makes jsp code to gnerate the output by executing the code and that output will be commented..

Can we comment html code/template text with jsp commnets?
Ans) possbile and recomanded also..

Can we use scripting/java comments to comment html code/jsp code?
Ans) not possible...

Scopes in jsp Programming (Talks about the Visibility of data)
==========

a) page scope (Specific to current jsp page)
b) request scope (specific to each request --Visible through out request)
c) session scope(specific to each browser s/w of a client machine)
d) application scope (specific to each web application i.e visible in all web comps of web application)

note:: applicationScope means data is visible in all web comps of a web application and not specific any browser and any request..
note:: sessionScope means data is visible in all web comps of a web application with respect to single browser s/w of a client machine..

Test t= new Test();
reference type — Object type

Object obj = new Test();
refernce type — Object type

Xyz x=new Test()
reference type — Object type

Test(c) is implementing Xyz(I)

implicit obj	reference type	scope
request	javax.servlet.http.HttpServletRequest(I)	request
response	javax.servlet.http.HtttpServletResponse(I)	response/request
page	java.lang.Object (c)	page
pageContext	javax.servlet.jsp.PageContext(AC)	page
session	javax.servlet.http.HttpSession(I)	session
config	javax.servlet.ServletConfig(I)	page
application	javax.servlet.ServletContext(I)	application
out	javax.servlet.jsp.JspWrite(AC)	page
exception	java.lang.Throwable(C)	page

note:: we do not create these implicit objs, the jsp cotainer creates them having fixed reference type given by servlet/jsp api.. and varying object type based on the container/server we use.

Exmaple :: the reference type of " request " obj is type alwyas javax.servlet.http.HttpServletRequest(I) .. but its object type is specific to each server implementing "HttpServletRequest(I)".

request obj class name:: <%=request.getClass() %>

(Tomcat server)
request obj class name :: class org.apache.catalina.connector.RequestFacade
⇩
implements HttpServletRequest(I)

(Wildfly server)
equest obj class name :: class io.undertow.servlet.spec.HttpServletRequestImpl
⇩
implements HttpServletRequest(I).

Important observations
==========

(a) Exception handling is optional only for the code that goes to _jspService(-,-) of JES class.. For remaining code we need perform exception handling manually.. i.e for the code placed in declaration tag we need to perfrom exception handling explicitly becoz this code goes to outside of _jspService(-,-) in JES class.

```
<%! public void jspInit(){
        try{
          Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch(Exception e){
        }
    } %>
```

JES class
=========

```
public class first_jsp extends ....{

  public void jspInit(){
        try{
          Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch(Exception e){
        }
  }
  public void _jspService(req,res)throws SE,IOE{
      ...
      ...
  }
}
```

When Exception will be raised?

Ans) It will be raised for rum time problems.. and causes abnormal termination in the execution of the Application.

What is the meaning the handling exception?

=> placing try/catch block for the code that raises exception is called exception handling.. i.e when exception raised it will not terminate the application rather control goes catch block and further statements will execute.

=>Exception handling also useful to convert system generated technical messages to enduser specifics non-technical messages..

What is the diff b/w checked exception and unchecked exception?

Chekced Exception	Unchcked Exception
(a) catching and handling exception or declaring the excpetion to be thrown is mandatory otherwise causes compile time error	(a) catch and handling exception is optional if not caught and handled it propagates the exception to caller
(b) does not propagate the exception by default..we must explicitly enable this using "throws"	(b) supports exception propagation by default if that exception is not caught and handled.
(c) These classes sub direct sub class of java.lang.Exception	(c) These classes are direct or indirect sub classes of java.lang.Runtime class..

note:: Both checked and unchecked exceptions are run time errors....

note:: In realtime, we use unchecked exceptions most of time to enjoy exception propagation/ passing in layered applications

---

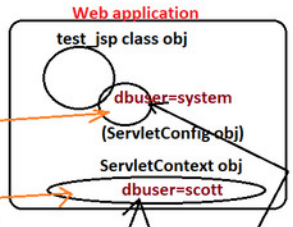
=>ServletConfig, ServletContext (application) objects visible in all the life cycle methods servlet comp or JES class

test.jsp
=======

```
<%! public  void jspInit(){
     String dbuser=config.getInitParameter("dbuser");
        (or)
     String dbuser=application.getInitParameter("dbuser");
  } %>
```

JES class

```
public class test_jsp extends .....{
     public  void jspInit(){
        ServletConfig cg=getServletConfig();
        SErvletContext sc=getServletContext();
        String dbuser=config.getInitParameter("dbuser");
           (or)
        String dbuser=application.getInitParameter("dbuser");
     }

     public  void _jspService(req,res)throws SE,IOE {
        //implicit objs /reference variable
        ...
        ...
        application , conflg
        ...
        }
     }
```



=>Instead calling implicit objs of jsp...it is better to call them as implicit reference variables becoz they are pointing to container create objs like request,response , SevletContext ,ServletConfig, HttpSession objs and etc..

=> when u can not access Container created objs in any part of Jsp,Servlet comps using implicit reference variables/objects then u r own reference variables accessisng them as shown above.. (cf,sc)

web.xml
========

```
<web-app>
  <servlet>
     <s-n> t</s-n>
     <jsp-file>/test.jsp </jsp-file>
      <init-param>
          <param-name>dbser </param-name>
          <param-value> system </param-value>
      </init-param>
     </servlet>

     <sevlet-mapping>
        <s-n>t</s-n>
        <url-pattern>/testurl </url-pattern>
     </servlet-mapping>
     <context-param>
        <param-name>dbuser</param-name>
         <param-value> scott </param-value>
      </context-param>
</web-app>
```

=>Instead of hard coding technical input values directly in the jsp page, we can get them from web.xml file through ServletConfig ,ServletContext objs.

=>if the technical input values are specific one servlet /jsp comp then use init param values..

=>if the technical input values are common for multiple servlet /jsp comps then use context aram values..

note:: the cfgs done on jsp page in web.xml file will takes place on jsp page only when that jsp page is requested through url pattern .. otherwise they will not takes place..

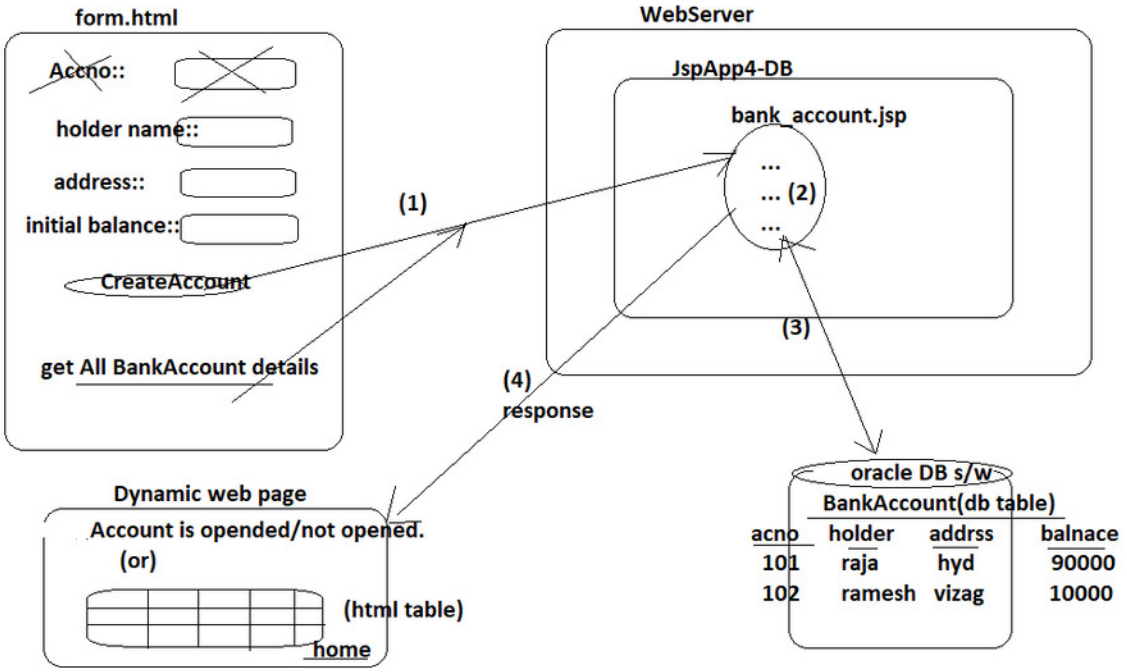
html to jsp to Db s/w

html to jsp communication possible using 3 approaches

a) using hyperlinks (place jsp file name or url pattern as the "href" attribute value of <a> tag)
b) using submit button (place jsp file name or url pattern as the "action" attribute value of <form> tag)
c) using java script (take the support of form.submit() method)

note:: jsp to Db s/w communication we need to add jdbc code in jsp page.. by also keeping jdbc driver s/w related jar file (like ojdbc6.jar) in WEB-INF/lib folder..

Example App



```
JspApp4-DB
   |----->webcontent
          |---->form.html
          |---->bank_account.jsp
          |--->WEB-INF
                |-->pages
                    |---->bank_account.jsp
                |--->lib
                     |--->ojdbc6.jar
                |---->web.xml
```

a) Differentiate logics for hyperlink and submit in the jsp page.
b) Collect jdbc properties from web.xml file as init param values.. for access ServletConfig object seperately in jspInit() method
c) invovle all 3 jsp life cycle methods jspInit(), _jspService(-,-) and jspDestroy()
d) avoid out.println(-) completely from coding .. with the support of expresson tags..
e) use all the 3 scripting tags... in jsp pages..
and etc..

```
CREATE TABLE "SYSTEM"."JSP_BANK_ACCOUNT"
 (    "ACNO" NUMBER(10,0) NOT NULL ENABLE,
      "HOLDERNAME" VARCHAR2(20 BYTE),
      "ADDRESS" VARCHAR2(20 BYTE),
      "BALANCE" FLOAT(126),
       CONSTRAINT "JSP_BANK_ACCOUNT_PK" PRIMARY KEY ("ACNO"))
```

```
CREATE SEQUENCE  "SYSTEM"."JSP_ACNO_SEQ"  MINVALUE 1000 MAXVALUE 10000000
INCREMENT BY 1 START WITH 1000
```

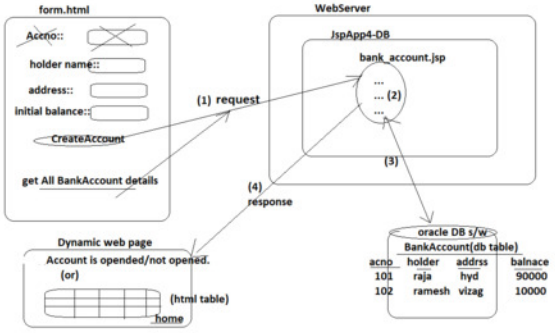
html to jsp to Db s/w
==========

html to jsp communication possible using 3 approaches
---

a) using hyperlinks (place jsp file name or url pattern as the "href" attribute value of <a> tag)
b) using submit button (place jsp file name or url pattern as the "action" attribute value of <form> tag)
c) using java script (take the support of form.submit() method)

note:: jsp to Db s/w communication we need to add jdbc code in jsp page.. by also keeping jdbc driver s/w related jar file (like ojdbc6.jar) in WEB-INF/lib folder..

Example App
==========



JspApp4-DB
|---->webcontent
|---->form.html
|---->bank_account.jsp
|--->WEB-INF
|--->pages
|---->bank_account.jsp
|--->lib
|--->ojdbc6.jar
|---->web.xml

a) Differentiate logics for hyperlink and submit in the jsp page.
b) Collect jdbc properties from web.xml file as init param values.. for access ServletConfig object seperately in jspInit() method
c) invovle all 3 jsp life cycle methods jspInit(), _jspService(-,-) and jspDestroy()
d) avoid out.println(-) completely from coding .. with the support of expresson tags..
e) use all the 3 scripting tags... in jsp pages..
and etc..

```
CREATE TABLE "SYSTEM"."JSP_BANK_ACCOUNT"
(    "ACNO" NUMBER(10,0) NOT NULL ENABLE,
     "HOLDERNAME" VARCHAR2(20 BYTE),
     "ADDRESS" VARCHAR2(20 BYTE),
     "BALANCE" FLOAT(126),
      CONSTRAINT "JSP_BANK_ACCOUNT_PK" PRIMARY KEY ("ACNO"))

CREATE SEQUENCE  "SYSTEM"."JSP_ACNO_SEQ"  MINVALUE 1000 MAXVALUE 10000000
INCREMENT BY 1 START WITH 1000
```

Directive tags
==========

=>These are given to give directions to jsp compiler to generate the java code in JES class..
i.e based on the instructions given in directive tags the code in JES class will be generated.
=>3 directive tags are given in jsp page
a)page directive
b) include directive
c) taglib directive

=>These tags makes jsp container to given special instructions jsp page compiler to generate java code in JES class

directive tags standard syntax::
<%@<tag> attributes %>

directive tags xml syntax::
<jsp:directive.<tag> attributes />

page Diretive
==========

This tag given bunch of attributes to make Jsp Page compiler to add extra code in JES class by giving instructions to jsp container.

standard syntax:
<%@page attributes %>
xml syntax:
<jsp:directive.page attributes />

note::<$ @Page %> tag and its attributes are useful to provide global info jsp page..by giving instructions to jsp page compiler..

attributes
==========
info
language
errorPage
isErrorPage
extends
buffer
autoFlush
session
isThreadSafe
isELIgnored
import
contentType
pageEncoding
and etc..

info
====

=>useful to give short description of jsp page.. to explain the purpose of the jsp page
=>No default value
=>we can write description having multipe words..
=> Based on this jsp container makes the page compiler to generate getServletInfo() method in JES class..

test.jsp
========
<%@page info="this is report generation page" %>

In JES class
==========
```
p c    test_jsp extends ...{
p   String getServletInfo(){
      return " this is report generation  page";
    }
p    void  _jspService(req,res)throws SE,IOE{
      ...
      ...
    }
}
```

language
========

=>Allows to specify the script code language the should be there in underyling server as part of jsp traslation.. As of now "java" is the only language and default language that it supports.

<%@page language="java" %> // valid statement
<%@page language="c" %> // Invalid language
<%@page language="c++" %> // Invalid language

"default value is " java ..

JVM based languages are giving their own syntax and they own compiler.. but their complier give such .class files whilch can be executed by Java JVM.
eg:: groovy , kotlin, @, spark,sacal and etc..

import
======

Allows to spcify , the java packages to be imported to the JES clas...

<%@ page import ="java.sql.*,java.util.*"%>

note:: we can have multiple values as the comm seperated list of values
By detault JES class imports 3 pkgs :: javx.sevlet.*, javax.servlet.http.*, javax.servlet.jsp

<%@ page import="java.net.*,java.util.*, java.util.*" %>
Does not any error.. through we are importing same package twice..

<%@page> directive contentType

=>Allows to specify response content type by internally calling res.setContentType(-) method
=>default value is *text/html; charset=ISO-8859-1*

eg: <%@page contentType="text/plain"%>

test.jsp

```
<% response.setContentType("text/html"); %>
<%@ page contentType="text/plain"%>
```

Can you tell me which will be applied?

```
<% response.setContentType("text/html"); %>
```

These lines code always come after <%@page %> directive tag contentType attribute based response.contenType(-) method , So the explicitly called response.setContentType(-) will always override the <%@page> directive tag respoonse content type

To place the template text in different languages , we need to take the character encoding as UTF-8 along with the MIME type

<%@page contentType="text/html;charset=UTF-8" %>

eg:: test.jsp

```
<%@ page contentType="text/html;charset=UTF-8"%>

<b> hello</b> <br>
<b>□□□□ </b> <br>
<b>Hallo </b> <br>
<b>□□□□ </b>
```

note:: while test.jsp file we need to take UTF-8 as the encoding/charset type

different charsets are
a)ascii (256 chars)
b) unicode (65,535)
c)utf (7,8,16,32)
d) ISO
and etc..

Max the languages of world covered in utf-8 range..

utf-->unicode transformation format

extends

Allows us to specify, the programmer java class as the JES class's super class..But not recomanded becoz that class has to given minimum the following standards based code..
They are
a) class must extend from HttpServlet
b) should override Servlet life cycle methods calling jsp life cycle methods internally
b) should develope _jspInit(),jspDestroy(), _jspDestroy(-,-) ,_jspInit() as empty methods..
and many kore..

```
<%@ page extends="com.nt.comp.TestBase"%>
```

```
public class test_jsp extends TestBase{
  ....
  ....
  ....
}
```

This code is impossible

=>no default value this tag..

session

=>Allows us to sepcify wheather the implicit objet session will be creator not..
session="true" :: Session object will be created
session="false" :: Implicit Session object will not be created

<%@page session="false" %> (or)
<%@page session="true" %>
default value is "true"

=>if do not we use "sessiosn tracking " on our web application.. it really bad practice to enable Session object in those movies..

isELIgnored

=>Writing java code in jsp page is bad practice.. So we should avoid it or minimize it becoz the java code in jsp page kills the redability.. but to perform arithmetic and logical operations in jsp page we need java code.. To overcome this use EL.. to perform arithmetic and logical operations..

syntax :: ${<expr>} :: It evalues the expression dipslay the output on to the browser.. (it is expression tag)

eg1:: <%@ page isELIgnored="true"%>

${4+5} → Gives ${4+5) as text text..

eg2: <%@ page isELIgnored="false"%>
${4+5}
→ *Recongnizes the EL and display 9 as output.*

the default is of this attribute :: false so ${4+5} gives 9 through <%@page %> is not included..

pageEncoding

==>allows to specify encoding charset for the jsp page...
===> Instead of writing charset along with contet type we can place seperately ..

```
<%@page pageEncoding="utf-8" contentType="text/html" %>
${4+5}
<b> hello</b>
<br>
lakatsa
□□□□□□
```

buffer && autoFlush *(attributes of <%@page %>)*

=>Allows us to specify jsp buffer size
=> default is 8kb ..

Servlet comp

```
pw.println("....");
pw.println("....");
pw.println("....");
```

response obj

Servlet comp writes the output messages to response obj directly

Jsp comp /JES class

```
out.write("....");
out.println("....");
out.println("....");
out.print("....");
out.print("....");
out.write("....");
```

buffer/cache

response obj

Jsp comp writes the output messages to response obj through jsp buffer/cache.

flushing :: The Process of transfering/sending buffer/cache content to its destination like response obj is called flushing.

Jsp buffer will be flushed automatically in the following 2 situations
a) When buffer/cache is filledup
b) Jsp page execution is over but the buffer is not filled up.

To enable or disable this autoflushing use "autoFlush" attribute of <%@page%>

```
<%@ page buffer="10kb" autoFlush="true" %>
```

default value: 8kb
default vlaue "true".

note:: we can flush buffer/cache manually by using out.flush() method call.

Q) what happens of jsp page generates more than buffer size output content ?

=>if autoFlush is enabled.. no problem.. every time the buffer content will be flushed out to response obj when buffer is fillied up..

```
<%@page buffer="2kb" autoFlush="true" %>

<% for(int i=0;i<=100000;++i){
     out.print("hello"+i);
  }%>
```

if autoFlush is disabled, then IOExcepion : Jsp Bufer Over flow exception will come

```
<%@page buffer="2kb" autoFlush="false" %>

<% for(int i=0;i<=100000;++i){
     out.print("hello"+i);
  }%>
```

we can solve the above the problem by doing explicit/manual flushing of jsp buffer/cache.

```
<%@page buffer="2kb" autoFlush="false" %>

<% for(int i=0;i<=100000;++i){
      if(i%100==0)
         out.flush(); // flushes the jsp buffer for 100 iterations

      out.print("<br> hello"+i);
  }%>
```

we can disable buffer of jsp page by taking buffer="none" i.e jsp page generated messages using out.write(-) ,out.println(-)/out.print(-) goes to response object directly with out taking the support of buffer In that situation disabling autoFlush mode is meaning less and throws "BadCombo" message.

```
<%@page buffer="none" autoFlush="false" %>
```

org.apache.jasper.JasperException: /test.jsp (line: [1], column: [2]) jsp.error.page.badCombo

```
<%@page buffer="none" autoFlush="true" %>
```

Does not throw any Exception.. though autoFlush is enabled by disabling buffer i.e jsp page directly writes the output messages to response object directly.

The above "buffer" "autoFlush" details will be reflected in JES calss in a statement that creates pageContext object..

```
pageContext = _jspxFactory.getPageContext(this, request, response,
                                          null, false, 10240, true);
```

session mode | buffer size | autoflush mode.

What is the diff b/w PrintWriter and Jsp Writer? (imp)

PrintWriter	JspWriter
(a) PrintWriter does not use Buffer while writing messages to destination	(a) JspWriter uses buffer while writing messages to destination(response obj)
(c) print(-),println(-) methods here do not raise IOException.	b) Every method throws IOException
(d) useful in servlet programming	It c) Is the type of "out" implicit obj in Jsp pages..

note:: JspWriter internally uses PritWriter to write messages to the destination response objs when jsp is not buffered.

note:: Displaying "null" values is not poosible in both write(-) method of PrintWriter and JspWriter objss..

What is the diff b/w page and PageContext?

page	pageContext
a) holds "this" nothing but reference of JES class obj	b) holds entire infomation about curren jsp page like req,res , error page, session mode, autoFlush side and etc..
b) useful to differentiate scriptlet local variables from declartion tag variables wrn <%@page isThreadSafe="false"%>	b) Using this object we can access other implicit objs and we can create 4 scopes of data..page , request, session, appliation)
c) "page"obj reference type is java.lang.Objct	c) "pageContext"obj reference type is javax.servlet.jsp.PageContext

isThreadSafe (default value is "true")

=Make JES class implementing special interface .. to make JSP/JES calss as Thread safe servlet/jsp.
=>That specila interface is "javax.servlet.SingleThreadModel (I) .. By seeing Interface implemetation the underlying server/container will be allowing only one thread at a time on to Jsp page or JES class object..

```
<%@page isThreadSafe="false"%>   (reverse meaning)
```

-JES class implements javax.servlet.SingleThreadModel(I) and makes container to take servelt multhreading. (Deprecated interface)

```
<%@page isThreadSafe="true"%>   (reverse meaning)
```

JES class does not implicit any special interfae..So we need using manually by synchronization concept.. : to handle multithreading

```
<% synchronized(session){
      ... // use session object
      ...
   }

   synchronized(application){
      ... // use application object
      ...
   }

} %>
```

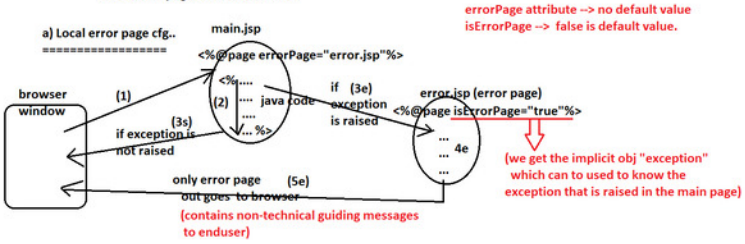
Error Pages cfg in jsp

=>The page executes only when exception is raised in other jsp pages is called error page
=> we can take either html or jsp page as the error page.. but jsp page is recomanded to take becoz u can use the implicit obj "exception" in that..
=>Error pages cfg is not given for exception handling .. It is given for displaying non-technial guiding messages to enduser for the exceptions raised in jsp page..

=>we can perform error pages cfg in two ways
a) Local error pages cfg (Specific to each jsp page)
=>Using errorPage,isErrorPage attributes of <%@page %> tag.
b) Global error pages cfg (common for all jsp pages of web application)
use <error-page> of web.xml file.

errorPage attribute --> no default value
isErrorPage --> false is default value.

a) Local error page cfg..



note:: The implicit object "exception" will be created .. only in the jsp page that is acting as error page (isErrorPage="true"%>

note:: In this approach the configured error page will execute for all the exceptions that are raised in main jsp pages.. but in every main jsp page we should cfg <%@page errorPage %> attribute.

main1.jsp

```
<%@ page errorPage="error.jsp"%>
<%
  int x=Integer.parseInt("a10");
%>
  value :: <%=x %>
```

error.jsp

```
<%@ page isErrorPage="true" %>
 <b>error.jsp</b>

 <br>
 <b><i> Internal problem -- Try Again</i></b>
 <hr>
 <%=exception.toString() %>
```

note:: we can take html file as error page.. but we can not use the implicit object "exception" in it.

---

b) Global Error Page cfg

=>To generate web.xml explicitly in Dynamic web project right click on the project --> JEE Tools ---> generate Deployment Descriptor sub

=>this is common the all jsp pages of web application...
=> This will not respond for the exceptions raised in servlet comps..

main1.jsp

```
<%
  int x=Integer.parseInt("a10");
%>
  value :: <%=x %>
```

error.jsp

```
<%@ page isErrorPage="true" %>
 <b>error.jsp</b>

 <br>
 <b><i> Internal problem -- Try Again</i></b>
 <hr>
 <%=exception.toString() %>
```

in web.xml

```
<error-page>
    <exception-type>java.lang.Exception</exception-type>
    <location>/error.jsp</location>
 </error-page>
```

we can cfg .. diff error pages for diff exceptions.. in web.xml file

```
<error-page>
   <exception-type>java.lang.NumberFormatException</exception-type>
   <location>/error.jsp</location>
</error-page>

 <error-page>
   <exception-type>java.lang.NullPointerException</exception-type>
   <location>/err.html</location>
</error-page>
```

---

if we write both specific exception type and common exception type error pages cfg in web.xml file then common exception type error page will execute for all exceptions..

```
<error-page>
    <exception-type>java.lang.Exception</exception-type>
    <location>/err1.html</location>
 </error-page>

 <error-page>
   <exception-type>java.lang.NumberFormatException</exception-type>
   <location>/error.jsp</location>
 </error-page>

  <error-page>
   <exception-type>java.lang.NullPointerException</exception-type>
   <location>/err.html</location>
 </error-page>
```

---

if we cfg both local error page and global error page for the same exception.. then which error page will execute when the exception is raised ?

Ans) Local error page executes

Error PAges cfg for http error codes

=>we can cfg diff error pages either as html pages or jsp pages for different http error codes..
These error pages will respond for the error codes raised for both servlet,jsp comps..

note:: we can cfg error pages based on http error codes like this..

```
<error-page>
    <error-code>404</error-code>
    <location>/404.jsp</location>
  </error-page>

 <error-page>
   <error-code>500</error-code>
   <location>/500.jsp</location>
 </error-page>
```

404.jsp

```
<b> 404.jsp</b> <br>
  <b> Wrong url problem</b>
```

500.jsp

```
<b> 500.jsp</b>
<b> internal problem..</b>
```

---

if exception is raised in servlet or jsp comp.. then the exception will be displayed on the browser havng error code 500 .. if cfg error pages for both error code and exception type then the exception type error page will execute..

```
<error-page>
   <exception-type>java.lang.Exception</exception-type>
   <location>/err.jsp</location>
 </error-page>

<error-page>
  <error-code>500</error-code>
  <location>/500.jsp</location>
</error-page>
```

<%@include %> /directive include /Static include

==========

=>This tag is given to include the code of dest web comp to the JES class of source jsp page

=>This tag performs code inclusion .. not the output inclusion..

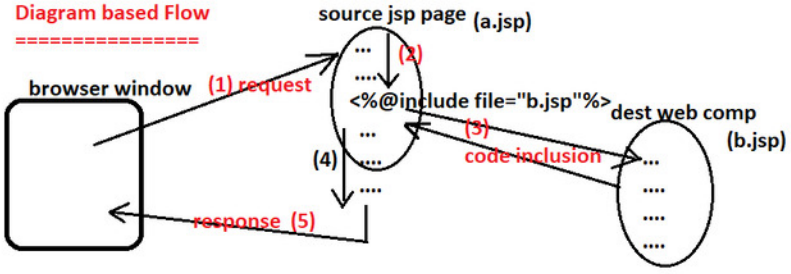
=> This tag does not use rd.include(-,-) internally ..

standard syntax :: <%@include file="...."%>

xml syntax :: <jsp:directive.include file="..."%>

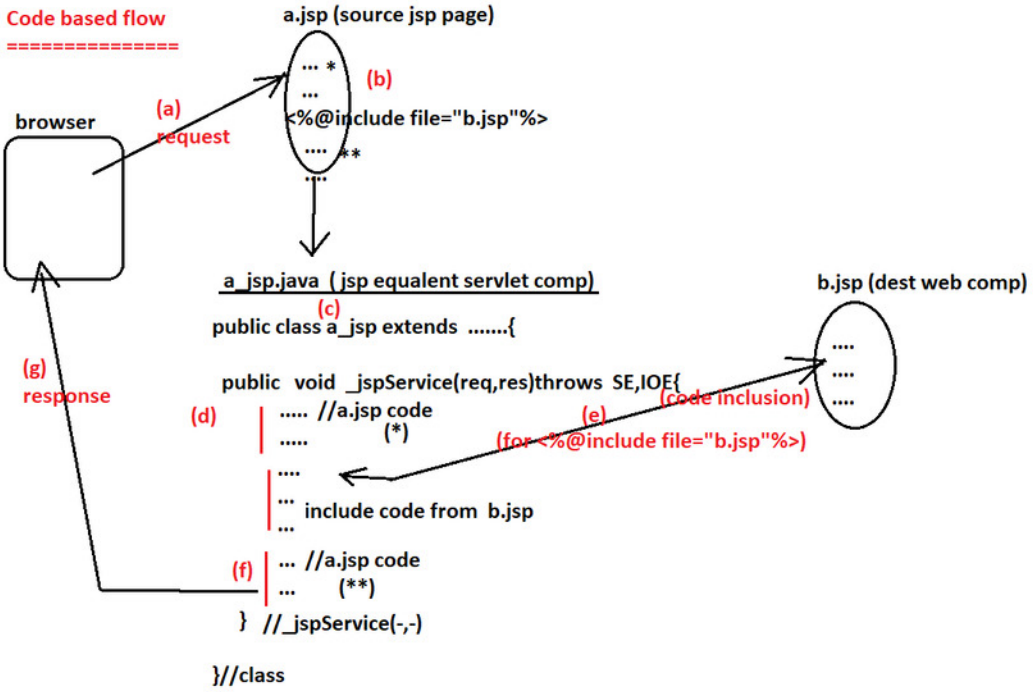
Diagram based Flow

==========



Code based flow

==========



=>In directive include , we can not take servlet comp as the destination comp becoz if the sevlet source code or byte code is included to the _jspService(-,-) of source jsp page then it becomes illegal code.

=> we can take html files ,jsp files as destion comps.. *if dest jsp page is having declaration tag s code ,they will be going to outside of _jspService(-,-) in source jsp's JES calss.*

=> This code inclusion is called static binding / compiletime binding.. becoz code the inclusion takes place at translation phase.

JspApp9- DirectiveInclude

|--->webcontent

|--->a.jsp,b.jsp

|--->WEB-INF

|--->web.xml

In directive the destionation html,jsp files will not executed...but their code will be included.. to JES source code of source jsp page

a.jsp code

==========

```
<b> start of a.jsp</b>
<br>
<%@include file="b.jsp" %>
<br>
<b> end of a.jsp</b>
```

b.jsp code

=========

```
<br>
<b> from b.jsp</b>
<%=new java.util.Date() %>
<br>
```

request url :: http://localhost:3030/JspApp9-DirectiveInclude/a.jsp

note:: No JES class will be generated for b.jsp.. but its code will be include to the JES class code of a.jsp page.. (i.e code inclusion is taking place)

note:: In one source jsp page we can place multiple directive includes as needed.. this includes the content of multiple destination comps to the JES class of source jsp page..

note: if the dest jsp.html comp files in private area of the web applicaton.. them we need to pass their complete path in <%@include file="...."%>

Action tags
==========
=>These tags performs activties dyanmically at run time by taking the support of servlet,jsp apis..
=>these tags are having only xml syntax .. there is no standard syntax
=>For example <jsp:include> internally uses rd.include(-,-) and <jsp:forward> internally uses rd.foward(-,-) and etc...

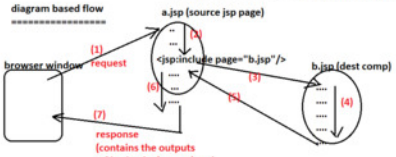
4 types of jsp Action tags
==========
a) Standard Action tags (given by SunMs as built-in tags of jsp)
b) JSTL Action tags (tags desgning givenSun Ms but implemetations are given by Servers)
c) Third party Action tags(given by third party like struts jsp tags, spring mvc jsp tags and etc..)
d) Custom Action tags (developed by the programmers)

List of jsp STandard Action tags
==========
<jsp:include> ,<jsp:forward> ,<jsp:useBean> ,<jsp:setProperty>,<jsp:getProperty>, <jsp:plugin> (old) , <jsp:fallback>(old) , <jsp:param> and etc..

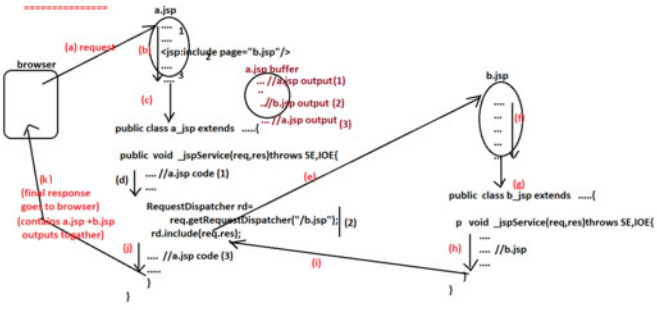
<jsp:include> /Action include/ dynamic include
==========
=>Performs output inclusion.. by internally using rd.include(-,-) method.
=> This output include takes place dynamically at runtime/execution phase .so it is called dynamic include or runtime include or dynamic binding..
=> syntax <jsp:include atttributes />

page --> to specifty destination comp details *(no default value)*
flush -> To specify wheahter source jsp page buffer should be flushed or not before including the output of dest web comp. *(true/false (default))*

diagram based flow
==========



Code based flow
==========



note:: here destination web comp code will not be included.. but its output will be included..

JspApp10-ActionInclude
|--->webcontent
|--->a.jsp
|--->b.jsp
|--->WEB-INF
|--->web.xml

Here JES classes for a.jsp (source page) and b.sjp (dest page) will be generated seperately..

a.jsp code

```
<b> start of a.jsp</b>
<br>
  <jsp:include page="b.jsp"/> <br>
  <br>
  <b> end of a.jsp</b>
```

b.jsp code

```
<br>
<b> from b.jsp</b>
<%=new java.util.Date() %>
<br>
```

note:: In one jsp page we can place any no.of directive includes and action includes..
note:: while working with <jsp:include> which internally calls rd.include(-,-) we should not commit response (like calling pw.close()) in the dest component..

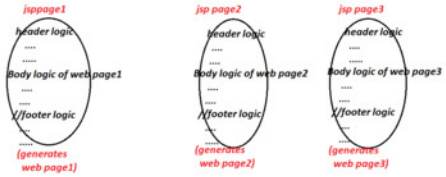
what is the diff b/w directive include and Action include?

Directive include	Action include
(a) Performs code inclusion at translation phase , So it is called static binding/ compile time binding	(a) performs output inclusion at runtime or executio phase so it is called dynamic binding or runtime binding
(b)Does not allow to take servlet comp as the dest comp	(b) Allows to take
(c) if the dest comp is jsp page.. then that jsp will not execute and does not generate JES class for it	(c) Dest jsp page executes and also generates the JES class..
(d) gives two syntaxes a) standard syn b) xml syn	(d) gives only xml syntax
(e) does not use rd.include(-,-) internally	(e) uses
(f) useful if the dest comp static web comp like html	(f) useful , if the dest comp dynamic web comp like servlet,jsp comps

eg:: <jsp:include page="b.jsp" flush="true"%>

To specify wheahter source jsp page buffer should be flushed or not before including the output of dest web comp. *(true/false (default))*

Problem::
=======



note:: Here header , footer logics are resuable logics.. becoz they are repeated in multiple jsp comps.. with no chanages (boilter plate code)

Solution: composite view design pattern
==========

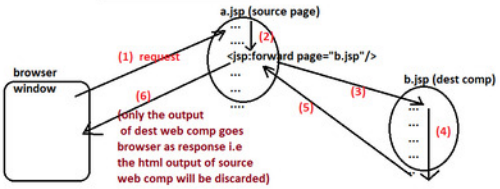
note:: the code that repeates across multiple parts of Project either with no chanages or with minor changes is called boiler plate code program.

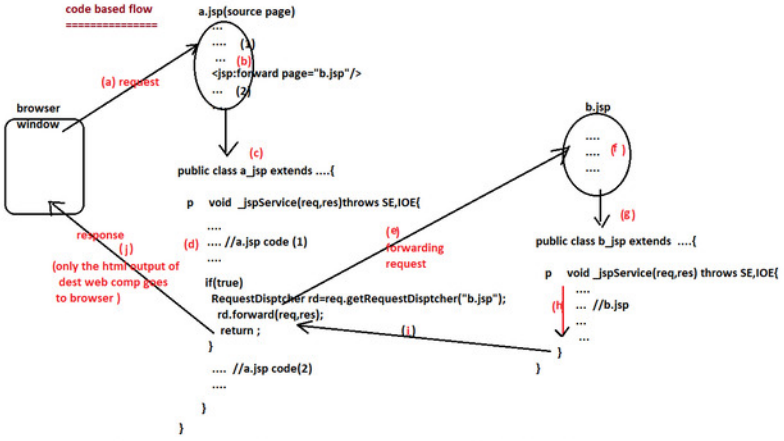
note:: here header,footer logics are reusable and there is no boiler plate code problem
note: here every web page comes having the outputs given by multiple web comps.. So it is composite view design pattern

Action Forward/ <jsp:forward>
==========

=> It internally uses rd.forward(-,-) and performs forwarding request operation..
=> the html output source jsp page will be discarded..and only the html output of dest web comp goes to browser as response..
=> There is no directive forward tag .. but we have action forward tag..

syntax:: <jsp:forward attributes />





=>Once the rd.forward(-,-) is executed.. the existing output from response obj will be dicarded.. and only the output collected from dest web comp will be stored in response obj and also commits output to response object i.e it does not allow to added further output to response object

note:: while working <jsp:include> ,<jsp:forward> tags...the source jsp page and dest web comp will use same req,res objs..becoz both are dealing with same request.

```
JspApp11-ActionFoward
  |--->webcontent
         |--->a.jsp
         |--->b.jsp
         |--->WEB-INF
                |--->web.xml
```

a.jsp
=====

```
<b> start of a.jsp</b>
<br>
  <jsp:forward page="b.jsp" /> <br>
<br>
<b> end of a.jsp</b>
```

b.jsp
====

```
<br>
<b> from b.jsp</b>
<%=new java.util.Date() %>
<br>
```

Why there is no directive forward tag ?

Ans) Directive tags perform their work by generating code in JES class.. if the generated code is added, then output discarding is not possible.. but forwarding operation needs output discarding. So directive forward is not given..

what is the diff b/w <jsp:include> and <jsp:forward>?

< jsp:forward>	<jsp:include>
(a) performs the forwarding mode of jsp communication	(a) performs including mode of jsp communication
(b)only the html output of dest comp goes to browser as response	(b)both source and dest web comp outputs togather goes to browser as response.
(c) statements placed after <jsp:forward> tag will not be executed..	(c) will be executed
(d) if we place multiple <jsp:forward> tags only first will be executed..	(d) multiple <jsp:include> tags will be executed togather..
(e) Use case:: Conditionally forwarding dest comps to execute special logics like forwarding to discount.jsp from bill.jsp only when billAmt>=50000	(e) usecase:: composite view design pattern implementation..
(f) internally uses rd.forward(-,-)	(g) internally uses rd.include(req,res)

what happens if we place <jsp:forward> and <jsp:include> tags in the same source jsp page?

Ans) Since <jsp:forward> tag not only discards orginal output of soruce jsp page ,it also discards.. the included output,So there will be no effect of <jsp:include> tag.

<jsp:param>
==========

=>must be used only as the sub tags for <jsp:forward> or <jsp:include> tags
=>Useful to pass data as the addtional request parameter values from source jsp page to dest web comp.
syntax: <jsp:param attributes />
|-->name, value

```
//a.jsp (source page)
<b> start of a.jsp</b>
  <br>
   <%
     float bAmt=300.0f+(300.0f * 0.03f);
   %>
   <jsp:forward page="b.jsp" >
       <jsp:param value="CRJ" name="bkName"/>
       <jsp:param value="<%=bAmt %>" name="billAmt"/>
   </jsp:forward>
     <br>
  <br>
  <b> end of a.jsp</b>
```

Internally uses request object to put addtional req param value..

b.jsp (dest page)

```
<br>
<b> from b.jsp</b> <br>
<%=new java.util.Date() %> <br>
  book name is ::<%=request.getParameter("bkName") %>,
  book price ::<%=request.getParameter("billAmt") %>.
<br>
```

Jsp Communication
==========

It is all about taking request from browser to jsp and passing other dest web web cmps..

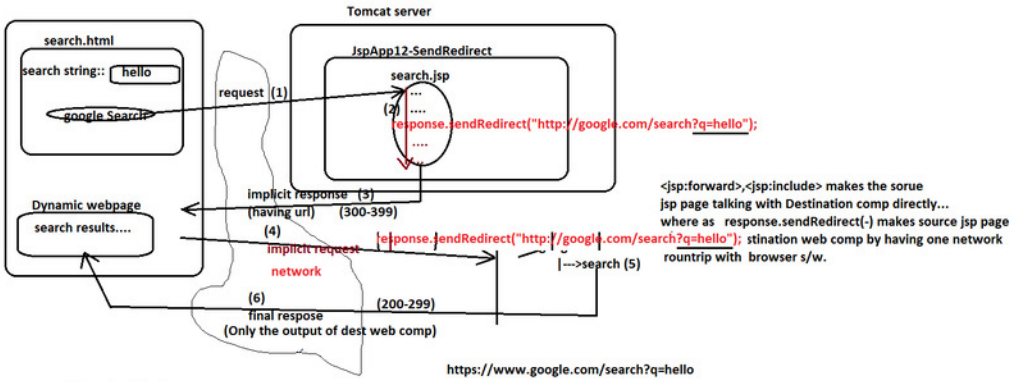
if source jsp page and and dest web comp are there in the same web application
==========
a) <jsp:include> (for Including response mode of jsp communication)
b) <jsp:forward> (for forwarding request mode of jsp communication)

=>source jsp page and dest web comp will use same req,res objs
=>dest web comp must there in jsp,html servlet which can be taken in java web application

if source jsp page and and dest web comp are there in the two different web applications of same server or diffent servers belonging same machine or different machines
==========
use sendRedirection
|--> 1. using hyperlinks (bad)
|--> 2. using response.sendRedirect(-) method (No tag for this) (good)

=>Here source jsp page and dest web comp will not use same request ,response objs..
=>Here the dest comp can be there in any location and in any technology like html, servlet,jsp ,php,asp.net and etc..

Example App on response.sendRedirect(-) method for sendRedirection concept
==========



<jsp:forward>,<jsp:include> makes the sorue jsp page talking with Destination comp directly... where as response.sendRedirect(-) makes source jsp page stination web comp by having one network rountrip with browser s/w.

JspApp12-SendRedirection
|---->webcontent
|-->search.html
|-->search.jsp
|--->WEB-INF
|--->web.xml

refer JspApp12-SendRedirection application

What is the diff b/w <jsp:forward> and response.sendRedirect(-)?

<jsp:forward>	response.sendRedirect(-)
(a) performs the forwaring request mode communication	(a) perform sendRedirection mode commnunication
(b) source jsp page directly interacts with dest web comp	(b) interacts by having network round trip with browser
(c) Source jsp page and dest web comp uses same req,res objects	(c) will not use
(d) source jsp page can pass data to dest comp either as request attributes or using <jsp:param> tags (as addtional request params)	(d) here data can be passed by appending queryString to request the url placed in response.sendRedirect(-) method response.sendRedirect("http://google.com/search?q=hello");
(e) while doing forwarding request operation the url in browser's address will not be changed	(e) while doing sendRirection operation the url in browser's address will be changed
(f) dest comp must be placed in the place where the source web comp is available	(f) dest web comp can be placed any where...
(g) dest web comp must be on of the following comps (a)servlet b) jsp ,c) html	(g)Dest web comp can be any web comp.. like servlet,jsp, html ,php, asp.net and etc..
	https://www.google.com/search?q=naresh%20it

How to pass data from source jsp page to destination web comp?

if the source jsp page and dest web comp are there in the same web application
==========

and using same req ,res objs (keeps in request)
==========
=>use request attributes or <jsp:param> style addtional request params..

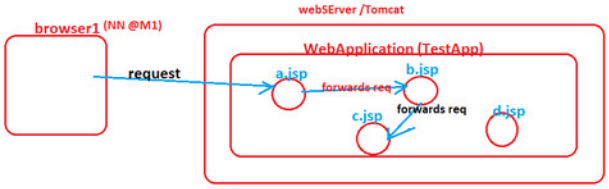
and using same browser of same client machine
==========
=>use session attribute (keeps in session scope)

and using same or differ browsers for different clients or same clients..
==========
=>application attributes (keeps in application scope)

jsp scopes
==========
page
request/response
session
appication.

if the source jsp page and dest web comp are there in not same web application
==========
Ans) append query to the String url that is required..to send addtional data alongwith in the query String..

response.sendRedirect("http://google.com/search?q=hello");



=>request attribute created in "a.jsp" is visible and accessible in b.jsp and c.jsp but not in d.jsp
=>sesson attribute created in "a.jsp" is visible and accessible in all other jsp pages..but they must get request from same browser for which session attribute is oginally created.
=>application attribute created in "a.jsp" is visible and accessible through out web application..irrespective of anu coditions.
=>page attribute created in "a.jsp" is visible and accessible only in the same jsp page.. (a.sjp)

=>Instead of using 3 different objects to create 4 scope attributes.. we can we use single "pageContext" object to create all the 4 scope of attributes.

note:: since pageContext obj holds multiple other implicit//all objects, So we can use on pageCotnext obj to access othe objects..to create diff scope attrbutes internally..

pageContext attributes
==========
=>Instead of taking 4 different objects like page,request,session,application to create 4 scopes of attributes we can use single pageContext object to create all the 4 scopes of attributes becoz 1 pageContext obj holds all the implicit objs of jsp ..

To create pageContext attributes
==========
pageContext.setAttribute("attr1","val1");
-->creates "attr1" attribute having page scope

pageContext.setAttribute("attr2","val2",pageContext.SESSION_SCOPE);
-->creates "attr2" attribute having session scope

the scopes are
pageContext.PAGE_SCOPE
pageContext.REQUEST_SCOPE
pageContext.SESSION_SCOPE
pageContext.APPLIATION_SCOPE

To modify pageContext attribute values
==========
pageContext.setAttribute("attr1","val11");
--> modifies the page scope pageContext attribute "attr1" value

pageContext.setAttribute("attr2","val22",pageContext.SESSION_SCOPE);
--> modifies the session scope pageContext attribute "attr2" value

note: attribute name must be string but value can be any object

To read pageContext attribute vlaue
==========
String value=(String) pageContext.getAttribute("attr1");
-->reads "attr1" attribute value from page scope

String value=(String) pageContext.getAttribute("attr2",pageContext.SESSION_SCOPE);
-->reads "attr2" attribute value from session scope

note:: attribute is logical variable name that value with scope.. it is not no way related xml/html tag attributes..

To find pageContext attribute value
==========
String val1=(String)pageContext.findAttribute("attr1");
String val2=(String)pageContext.findAttribute("attr2");
-->searches given attriubte in the multiple scopes in a following order ..where ever it finds it reads the attribute value... if same attribute is there in two scopes.. then lower scope gets priority.

page scope ---> request scope ----> sesssion scope -----> application scope ----> returns null
gets value gets value gets value gets value

note:: if we try to place simple value in any attribute then it will be converted into an wrapper automatically (auto boxing)
note:: if we try to read and hold the retrieved attribute value from any scope into simple data type variable..then the wrapper object (attribute value) will be converted into simple value using auto unboxing concept.

primitive/simple value ----> wrrapper obj (Auto boxing)
wrrapper obj ----------> primitive/simple (Auto unboxing)

What is the difference b/w getAttribute and findAttribute method?

=>getAttribute() method searches for given attribute only in the specified scope.. if not available then it will not search in other scopes..but it returns null

=>findAttribute() method searches for the given attribute in multiple scopes in a order.. that is shown above... if the attribute is not avaiable in all the scopes then it returns null .. if it is avaiable in specific scope then it collects and does not search in other scopes.. if attr is there in multiple scopes then it collects from specific lower scope.

To remove pageContext attribute
==========
refer JspApp12

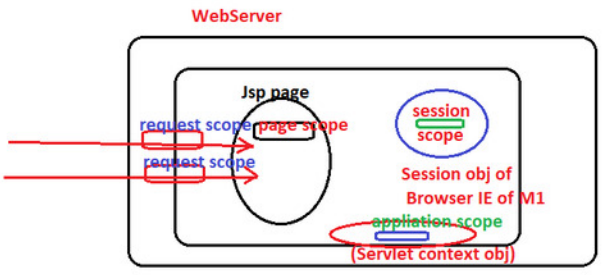
pageContext.removeAttribute("attr1");
-->Removes "attr1" attribute from page scope

pageContext.removeAttribute("attr2",PageContext.SESSION_SCOPE);
-->Removes "attr2" attribute from session scope

Can i work with pageContext attributs in servlet programming?
Ans) not possible becoz there is no pageContext object in servlet programming...

can i read pageContext attributes using direct page,request,session, application objs ?
Ans) yes .. but not recomanded...

note:: even reverse operation is also possible..



=>Java beans are helper classes that are useful to carry multiple values from layer to another layer or from one project to another project.

3 types of Java beans
==========
VO class :: Value Object class (To carry inputs or outputs)
DTO class :: DataTransfer Object class (to transfer data)
BO class/Entity class/Domain class :: To carry Persistable data for persistent data...

form page (html) ----------> Servlet ----DTO----->Service class

[servlet comp sends the recieved form data to Servlet comp as DTO class obj. for that Servlet comp should create DTO class obj manually]
(java code)

form page (html) ----DTO----> jsp ----DTO----> service class

[Jsp page can the following tags create DTO class obj and to set/read to/from DTO obj)

<jsp:useBean> :: To create or Locate Java bean class/java class obj
<jsp:setProperty> :: To call setter method and write data to bean property
<jsp:getProperty> :: To read and dispaly data from bean property.

<jsp:useBean>
==========
=>Useful to create or locate Java bean /class object to/from specified scope.
syntax:: <jsp:useBean attributes />

attributes
==========
id :: reference variable name (object name)
class :: fully qualified java bean class name.
scope :: scope to keep/locate java bean class obj
[page(default) /request /session/ application]
type :: referent type of Bean class object (super class name of bean class (or) interface implemented by the bean class)

eg1:: <jsp:useBean id="st" class="com.nt.dto.StudentDTO" scope="session"/>
=>Create StudentDTO class obj having name "st" and keeps in session scope if it is not already there in the scope .. otherwise it will collect from the existing scope..

JES class code
==========
//get object from scope
StudentDTO st=(StudentDTO)pageContext.getAttribute("st",pageContext.SESSION_SCOPE);
if(st==null){ //if not there
st=new StudentDTO(); //create object
pageContext.setAttribute("st",st,pageContext.SESSION_SCOPE); //keep in scope
}

=>StudentDTO is both reference, Object type

eg2: <jsp:useBean id="st1" class="com.nt.beans.StudentDTO" type="com.nt.beans.PersonDTO" scope="request"/>

PersonDTO is the super class of StudentDTO

code in JES class
==========
PersonDTO st1=(PersonDTO)pageContext.getAttribute("st1",pageContext.REQUEST_SCOPE);
if(st1==null){
st1=new StudentDTO();
pageContext.setAttribute("st1",st1,pageContext.REQUEST_SCOPE);
}

Here ref type is :: PersonDTO and object type is :: StudentDTO

attr name value scope

<jsp:setProperty>
==========
==>calls setXxx(-) and writes given data to java bean property ..
syntax :: <jsp:setProperty attributes />

attrbutes
==========
name :: give bean id (java bean/class obj name)
property :: bean property or xxx word setXxx(-) method
value :: value to assign (if needed convertion takes place)
param :: request param from whom data to be collected to assign to bean property
note:: use either value or param at a time

StudentDTO
|-->sno,sname,sadd
|-->getters and setters

eg1:: <jsp:setProperty name="st" property="sno" value="1001"/>
calls st.setSno(1001); to assign the value 1001 to the property "sno".

eg2: <jsp:setProperty name="st" property="sname" param="stname"/>
calls st.setSname(request.getParameter("stname")); to read "stname" req param value to assign to the bean property "sname" by calling st.setSname(-) method.

<jsp:getProperty>
==========
==> calls getXxx() method to read and display bean property values...
syntax:: <jsp:getProperty attributes />

attributes
==========
name :: bean id or object name
property :: the property name from whom we need to read and display the property value..

eg:: <jsp:getProperty name="st" property="sno"/>
calls out.print(st.getSno()); to read and display sno property value..

eg:: <jsp:getProperty name="st" property="sname"/>
calls out.print(st.getSname()); to read and display sname property value.

Example App
==========
JspApp14-useBean
|-->java resources
|--->src
|---->com.nt.beans
|----->StudentDTO.java
|--->webcontent
|--->set_values.jsp
|--->read_value.jsp
|--->WEB-INF
|-->web.xml

refer JspApp14-UseBean

To set request param value (form data) as java bean property values .. we can use param attribute of <jsp:setProperty> tag
==========

```
<!-- create or Locate Java bean class object -->
<jsp:useBean id="st" class="com.nt.dto.StudentDTO" scope="session"/>

<!-- To set request param values(form data) to bean properites -->
<jsp:setProperty property="sno" name="st" param="stno"/>
<jsp:setProperty property="sname" name="st" param="stname"/>
<jsp:setProperty property="sadd" name="st" param="stadd"/>
<jsp:setProperty property="avg" name="st" param="stavg"/>
```

URL from the browser ::
http://localhost:3030/JspApp14-UseBean/set_values.jsp?stno=101&stname=suresh&stadd=hyd&stavg=40.0

```
<!-- To set request param values(form data) to bean properites by matching
req param names with bean property names -->
<jsp:setProperty name="st" property="*"/>
```

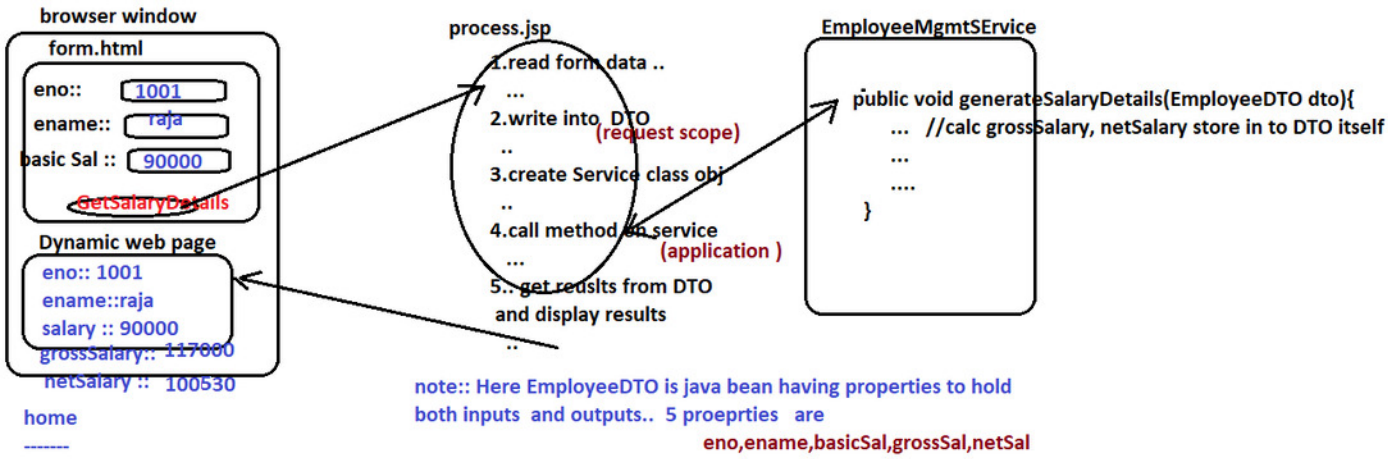
StudentDTO
|-->sno,sname,sadd,avg
|-->getters ,setters
|-->toString()

URL from the browser ::
http://localhost:3030/JspApp14-UseBean/set_values.jsp?sno=121&sname=mahi&sadd=delhi&avg=90.0

Use case on <jsp:useBean> ,<jsp:setProperty>,<jsp:getProperty> tags

==========

form page (html) ------> jsp ----------DTO----------->SErvice class



home

---

note:: Here EmployeeDTO is java bean having properties to hold both inputs and outputs.. 5 proeprties are eno,ename,basicSal,grossSal,netSal

note:: since multiple endusers may submit same form with diff values from multiple browsers and their tabs So it recomanded to take DTO class obj scope as request scope

note:: since there is not state for Service class ... it is better to create 1 object and use it for multiple times.. So prefer keeping Service class obj in application scope

```
JspApp15-UseBean-UseCase
    |----->java resoruces
            |--->src
                |--->com.nt.dto
                        |---->EmployeeDTO.java
                |--->com.nt.service
                        |---->EmployeeMgmtService.java
                        |--->EmployeeMgmtServiceImpl.java
    |---->webcontent
            |----->form.html
            |---->WEB-INF
                    |--->pages
                            |---->process.jsp
                    |--->web.xml
```

While develpoing java web application we need to place multiple logics like
a)request gathering logics (reading req params(form data), req headers(more info about browers),
Misc Info (req method, protocol, and etc..)
b) form validation logic
c) request processing logic /b.logic
d) presentation logic (user interface)
e) persistence logic (jdbc code....)
f) middleware services... (security ,auditing, TxMgmt and etc..)
and etc..

Different models of Java web application development

Model1

=> Here we take only servlets comps or only jsp comps as main web comps.. i.e if servlet comps are taken .. we do not jsp comps.. vice-versa..

=> It is more of jsp comps based web application Development..
=> So far we developing Model1 architecture based web applications..

=> Here multiple logics will be mixed up in main Servlet/jsp web comps..So we can say code becomes clumsy code..

Model2 (MVC)

=>Here we develop web applications as layered Applications.. by keeping different logics in different comps developed in different technologies.

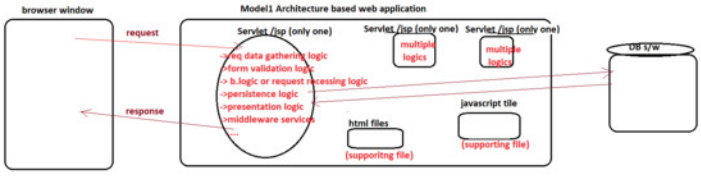
MVC (model2)
MVC1
MVC2

M-->Model Layer --> Data --> b.logic +persistence logic (like AO)
V-->View Layer --> presentation logics --> like Beautician
C-->Controller layer --> monitoring logics --> like Super viser..

=>Here code is not clumzy ..becoz we get clean seperation b/w logics here..

=>In java ,we have only MVC1,MVC2 architcture to develop web applications..
=> MVC3,MVC4,MVC5 ..and etc.. are not java architectures... they are versions of Asp.net mvc ..

Model1 Architecture
==========
=>In Servlet/jsp main web comps we place multiple logics...
=> if servlet comps are used as main web comps.. then jsp comps will not be used and vice-versa



Limitations of Model1 ARchitecture
==========
(a) clean seperation b/w logics is not possible becoz we are mixingup multiple logics
(b) The modifications done in one kind of logics will effect other logics
(c) Maintence and enhacement of the proejct becomes complex...
(d) parellel development is not possbile , So productivity is poor
(e) It is not industry standard architecture...

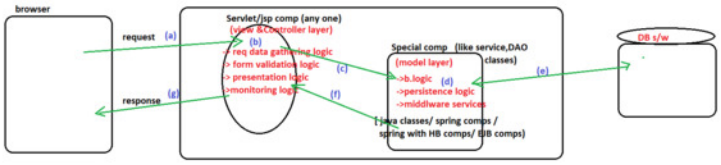
=>Use This architecture to develop small scale web applications like web application with 5 to 10 web pages.. which will not be maintained for long time..
(eg:: car model , movie promotional websistes)

advantages of Model1 architecture
==========
(a) knowledge on only Servlet comp or only jsp comp is sufficient to develope the Apps...

=>In MVC1 Architecture , we take seperate singlecomp having view, controller layers logics and we take seperate comps for model ayer.. (yesterday's App html --> jsp -->service class -DB app)

=>In MVC2 Architecture , we take seperate comps for view, seperrate comps for controller and we take seperate other comps for model layer..

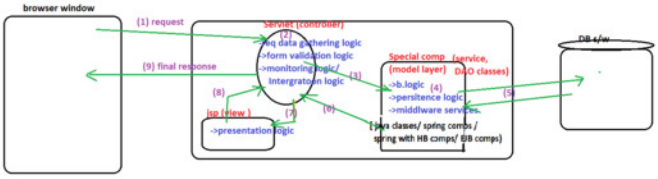
MVC1 ARchiteture
==========



note:: It gives seperation b/w logics where compare to model1 architecture.. If u want more seperation b/w logics we need to go for mvc2 architecture.. (we can also seperrate view comps)

=>this architecture is useful medium scale web applications like college , university , institute, company websites..
(10 to 20 web pages)

MVC2 Architecture
==========



Advantages
==========
=> clean seperation b/w logics , So code is not clumsy
=>the modifications done one layer logics does not effect other layer logics ..
=> The maintence and enhancement becomes easy
=>parellel development is possible.. , so productivity
=>It is indudstry defacto standard .
=> the f/w like spring ,HB gives more middleware services to use..

Disadvantages:
==========
=>knowledge on multiple technologies is required
=>for parellel development more programmer are required..

How parellel development is possbile?
TL divides the team into two parties

Party1->business-tier developers:: these developers use JEE technologies, f/w to develop model layer service, DAO classes logics

Party2->presentation-tier developers:: these developers use servlet,jsp , html, Css, JS other UI technologies to view, controller layer logics.

These two parties can work parellely either by staying in same company or different companies...

=>use this architecture to develop large scale complex websites.. (>20 webpages)
eg:: e-commerice sites, banking sites and etc...

MVC2 rules /priciples
==========
=> Every Layer is given to place /keep certain logics ..So place only those logics.. and do not place any addtional logics.
=> All operations /activities of the Application must take place under the monitoring or control of Controller Servlet..
=>There can be multiple view comps and there can be multiple model comps..but there should be only one ControllerServlet.
=> View Comps should not talk with model comps directly....and vice-versa.. They must interact with each other through controller Servlet..
and etc...

MVC = MVC2

Q) Can we change the role of various recomanded technologies in the development of MVC2 architecture based web application development.?
Ans) yes ,we can change .. but not recomanded....

Servlet as view comp
==========
=>The view comp presentation logic changes at regular intervals.. But modification of presentation logic or any logic in servlet comp maks the programmer to recompile the servlet comp and to reload the web application.. which is bit complex.. So prefer using jsp comps as the view comps becoz the modifications done in jsp comps will reflect with out recompilation and reloading activities..

jsp comp as controller
==========
=>Writing java code in jsp page is bad pratice.. but controller should have lot of java code to interact with service, dAO classes.. So taking jsp as controller with lot of java code is bad pratice.. So it is recomanded to take Servletcomp as controller

Java classes or Spring with HB or EJB comps as view , controller comps
==========
Ans) Not possbile... This though it self wrong ..becoz these are not web comps.. only we can web comps (like sevlet,jsp,html) as the view ,controller comps..

Another seperate servlet/jsp comps as model layer comps as Service ,daO classes
==========
Ans) Not recomanded to take ... becoz the b.logics or persistence logics placed in servlet/jsp comps
a) becomes Servlet/jsp technology specific logics
b) becomes http protocol specific logics
c) we can not access/invoke these logics from diff types of client Apps like mobile Apps , destkop Apps and etc....
conclusion:: place b.logic and persitence logic in ordinary java classes

Mini Project Discussions
==========

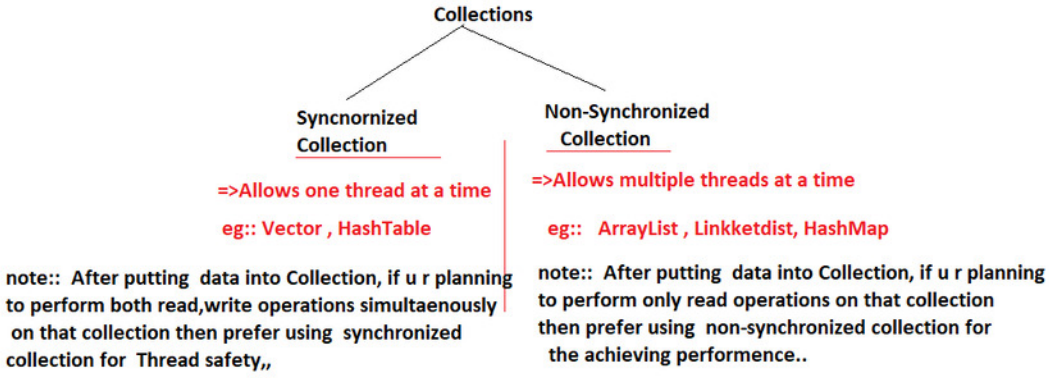
Problem1:
Writing jdbc code (persistence logic) in every layer not recomanded... We should place only in DAO classes..
So the RS (ResultSet obj) genereated in DAO class can not be sent to service to controller to jsp comps.. directly..

Problem2: We can send only Serializable objects over the network.. We can not send JDBC ResultSet obj from One Project to another project over the n/w becoz JDBC RS is not a Serilizable object..
(PayTM --->flipkart) (Gpay to Myntra) and etc..

Solutions
=========

Solution1:: copy RS object record to Collection and send the collection over the network
(note:: All collections are serializable objects by default)
(Good)

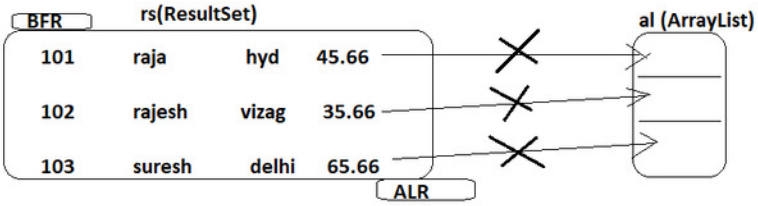
Solution2:: Use RowSets instead of ResultSets
(note:: RowSets are serializable objects by default -->Very few JDBC drivers support rowsets)



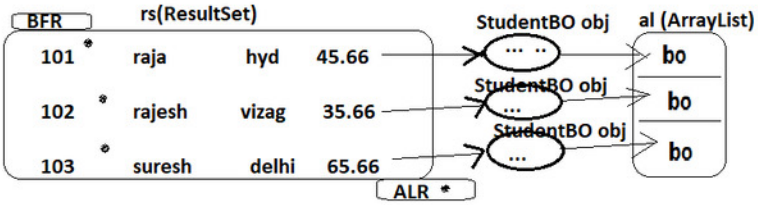
note:: After putting data into Collection, if u r planning to perform both read,write operations simultaenously on that collection then prefer using synchronized collection for Thread safety,,

note:: After putting data into Collection, if u r planning to perform only read operations on that collection then prefer using non-synchronized collection for the achieving performence..

While copying RS object records to List Collection (used to preserve the insertion order) we have a Problem that is each record of RS contains multiple values.. so they can not be copied as the each element of ArrayList becoz each element ArrayList can hold only one object..



We can solve this problem using java bean (BO) support .. copy each Record values to one object of BO class and add that object to the element of ArrayList.



Sample code to copy ResultSet object records to ArrayList collection as the object of BO class
==========

```
Statement st=con.createStatement();
ResultSet rs=st.executeQuery("SELECT * FROM STUDENT");
List<StudentBO> list=new ArrayList();
 while(rs.next()){
     //copy each record of Rs to each object of BO
    StudentBO bo=new StudentBO();
     bo.setSno(rs.getInt(1));
     bo.setSname(rs.getString(2));
     bo.setSadd(rs.getString(3));
     bo.setAvg(rs.getFloat(4));
     // add each of bO class to List collection
        list.add(bo);
     }
```

```
public class StudentBO {
   private int sno;
   private String sname;
   private String sadd;
    prvate float avg;
      //setters && geters
     .....
      ....
   }
```

Where did u use Java beans in u r Project?

Where did u use Collections in u r Project?

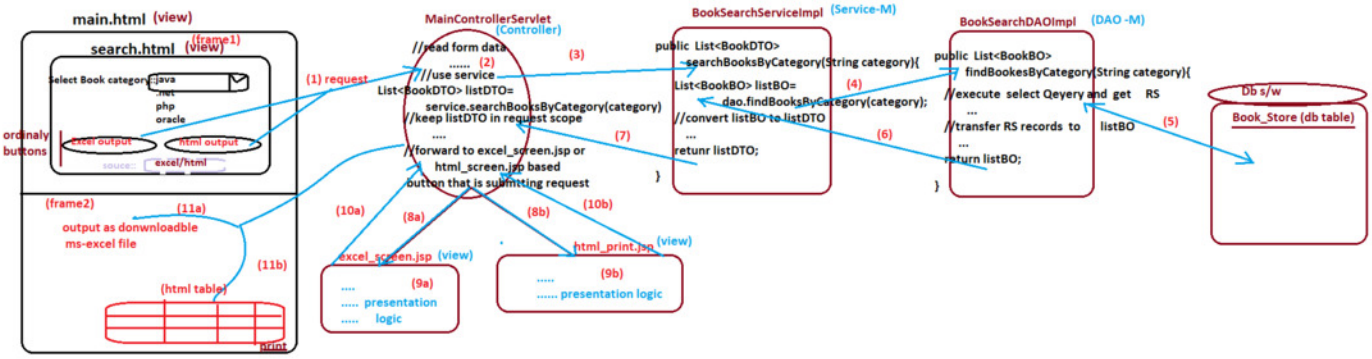
MVC2 Architecture based MiniProject

==========

html /jsp comps with js ----> as view comps

Servlet comp ---------> as Controller

Service class ,DAO class with java beans(BO,DTO)-----> model layer comps



(a) Develop based on mvc2 architecture

(b) pass bulk data (multiple book details) b/w layers as java bean class objs added to the ListCollection

(c) Here Java script is used

i)To submit the requests (based on ordinary buttons)

ii) for form validations

iii) To initiate printouts

note:: Appliication should use servermanaged jdbc con pool

(d) DAO , Service clases the propagates the exceptions.. and Controller Servlet comp catches and handle the exception

(e) same results(ListDTO) will be displayed in two diff formats.. 1. html table printout 2.downloable file as ms-excel file.

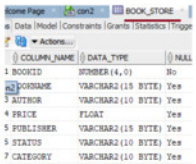
(f) rd.forward(req,res) will be used to pass control to mainControllerServlet to result jsp pages

(g) ControllerServlet passsing data to result pages.. as request scope attributes

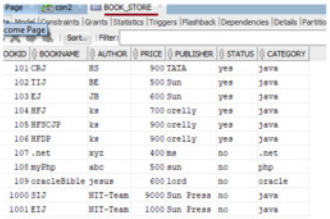
(h) On view comps based web pages..we are going to add headers,footers based by including other web comps output..

This called composite view dp.

and etc..



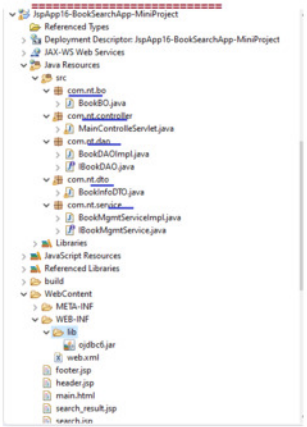
	COLUMN_NAME	DATA_TYPE	NULL
1	BOOKID	NUMBER(4,0)	No
2	BOOKNAME	VARCHAR2(15 BYTE)	Yes
3	AUTHOR	VARCHAR2(10 BYTE)	Yes
4	PRICE	FLOAT	Yes
5	PUBLISHER	VARCHAR2(15 BYTE)	Yes
5	STATUS	VARCHAR2(10 BYTE)	Yes
7	CATEGORY	VARCHAR2(10 BYTE)	Yes



OOKID	BOOKNAME	AUTHOR	PRICE	PUBLISHER	STATUS	CATEGORY
101	CRJ	HS	900	TATA	yes	java
102	TIJ	BE	500	Sun	yes	java
103	EJ	JB	600	Sun	yes	java
104	HFJ	ks	700	orelly	yes	java
105	HFSCJP	ks	900	orelly	yes	java
106	HFDP	ks	900	orelly	yes	java
107	.net	xyz	400	ms	no	.net
108	myPhp	abc	500	sun	no	php
109	oracleBible	jesus	600	lord	no	oracle
1000	SIJ	NIT-Team	9000	Sun Press	no	java
1001	EIJ	NIT-Team	1000	Sun Press	no	java

Deployment diectory structure

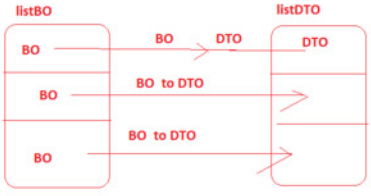
==========



from java 7 we have try with resource::

syntax:

```
try( resource obj){
  ...
  ... //use resource obj
  ...
} // here the resource obj will be closed automatically.
catch(Exception e){
  e.printStackTrace();
}
```



Jsp Custom TagLibrary Development (Custom jsp tags development)

=>Wrting java code (script code) in jsp page is bad pratice and it is having the following limitations
They are
(a) Jsp page/file code looks very clumzy.
(b) Not undestandable UI Developers.
(c) Kills the redability of jsp pages
(d) kills the reusability of java code placed in jsp page..

To overcome these problems.. we need jsp page as java code less(scriptless) jsp page by using the following cocepts
a) html tags
b) jsp built-in tags (Except scriptings)
c) EL
d) JSTL tags
e) Third party Tags (spring, themeleaf, sturts and etc.. supplied tags)
f) Custom tags

Custom tags

Terminlogies

a)Jsp taglibrary :: It is a library that contains bunch of jsp tags.. It is like java package
b)TagHandler class :: That java class that extends from javax.servlet.jsp.tagext.TagSupport class defining the functionality of jsp tag is called tag handler class.

```
public class ABCTag extends TagSupport{

   public int doStartTag(){  //executes automatically for open <ABC> tag
     ... //logic for opne <ABC> tag
     ...
   }

   public int doEndTag(){  //executes automatically for closing </ABC> tag
     ... //logic for closing </ABC> tag
     ...
   }
}
```

doStartTag() and doEndTag() methods are callaback methods becoz they will be executed automatically for open and closing jsp tags..

c) tld file (Tag Library Descriptor file)
|--> It is a xml file that contains various details about the jsp tags of jsp taglibrary
tag name , tag handler class name, attribute names and etc..

e) taglib uri :: every jsp taglibrary is identified with its taglib uri/url..and import jsp taglibray in jsp page by specifying its taglib uri using the support <%@taglib %> (directive tag)

Procedure to develop and use Custom Jsp taglibrary in our jsp page

step1) Design jsp taglibrary

```
NITTaglibrary
    |-----> <ABC> tag
    |-----> <XYZ> tag
```

step2) Develop Handler tag handler classes for Custom jsp tags in WEB-INF/classe folder

ABCTag.java (In WEB-INF/classes folder)

```
package com.nt.tags;
public class ABCTag extends TagSupport{

   public int doStartTag(){  //executes automatically for open <ABC> tag
     ... //logic for opne <ABC> tag
     ...
   }

   public int doEndTag(){  //executes automatically for closing </ABC> tag
     ... //logic for closing </ABC> tag
     ...
   }
}
```

doStartTag() and doEndTag() methods are callaback methods becoz they will be executed automatically for open and closing jsp tags..

XyzTag.java (In WEB-INF/classes folder)

```
package com.nt.tags;
public class Xyz:Tag extends TagSupport{

   public int doStartTag(){  //executes automatically for open <Xyz> tag
     ... //logic for opne <Xyz> tag
     ...
   }

   public int doEndTag(){  //executes automatically for closing </Xyz> tag
     ... //logic for closing </Xyz> tag
     ...
   }
}
```

(f) (h) (h)

doStartTag() and doEndTag() methods are callaback methods becoz they will be executed automatically for open and closing jsp tags..

step3) Develop TLD file having deteails about the jsp tags.

WEB-INF/nit.tld

```
<ABC> ----------> com.nt.tags.ABCTag class
<XYZ> ----------> com.nt.tags.XyzTag class
```

(e)

Orginally , this info will be there in the form of xml content

step1 to step3 :: complets the development of custom jsp taglibrary.

step4) cfg jsp library in web.xml file and generate tag lib uri

web.xml

```
<web-app>
  <jsp-config>
    <taglib>
      <taglib-uri>http://nareshit.com/tags/nit</taglib-uri>
      <taglib-location>/WEB-INF/nit.tld </taglib-location>
    <taglib>
  </jsp-config>
</webapp>
```

(d)

Jsp taglibrary cfg ... having taglib uri..

step5) import jsp taglibrary to the jsp page.. and use tags in that jsp page..

test.jsp

```
<%@taglib uri="http://nareshit.com/tags/nit" prefix="n" %>  (importing the jsp taglib uri)
                                                    (c)
  <n:ABC/>
(b)
  <n:XYZ >
    (a)
```

utilizing the recods..

What is the use of prefix?
=>if working multipl third party jsp taglibraies.. there is possility of having two tags with same name and different functionalaties.. So to differentiate one tag from another tag .take the support of prefixes.

=>The callback methods of tag handler givess instruction to jsp container to specify the next operation that it has to perform

javax.servlet.jsp.tagext.Tag	
public static final int	EVAL_BODY_INCLUDE 1
public static final int	EVAL_PAGE 6
public static final int	SKIP_BODY 0
public static final int	SKIP_PAGE 5

=> Every jsp tag handler class gets pageContext object as the inheritted Project. (as protected member vaiable of TagSupport class) and tag hanlder class can use that pageContext object to access to other jsp obejcts like out,response,equest,out, config, servlet context, and etc..

Jsp Custom TagLibrary Development (Custom jsp tags development)

=>Wring java code (script code) in jsp page is bad pratice and it is having the following limitations
They are
(a) Jsp page/file code looks very clumzy.
(b) Not undestandable UI Developers.
(c) Kills the redability of jsp pages
(d) kills the reusability of java code placed in jsp page..

To overcome these problems.. we need jsp page as java code less(scriptless) jsp page by using the following cocepts
a) html tags
b) jsp built-in tags (Except scriptings)
c) EL
d) JSTL tags
e) Third party Tags (spring, themeleaf, sturts and etc.. supplied tags)
f) Custom tags

Custom tags

Terminlogies

a)jsp taglibrary :: It is a library that contains bunch of jsp tags.. It is like java package
b)TagHandler class :: That java class that extends from javax.servlet.jsp.tagext.TagSupport class defining the functionality of jsp tag is called tag handler class.

```
public class ABCTag extends TagSupport{

   public int doStartTag(){  //executes automatically for open <ABC> tag
     ... //logic for opne <ABC> tag
     ...
    }
    public int doEndTag(){  //executes automatically for closing </ABC> tag
     ... //logic for closing </ABC> tag
     ...
    }
}
```

doStartTag() and doEndTag() methods are callaback methods becoz they will be executed automatically for open and closing jsp tags..

c) tld file (Tag Library Descriptor file)
|--> It is a xml file that contains various details about the jsp tags of jsp taglibrary tag name , tag handler class name, attribute names and etc..

e) taglib uri :: every jsp taglibrary is identified with its taglib uri/url..and import jsp taglibray in jsp page by specifying its taglib uri using the support <%@taglib %> (directive tag)

Procedure to develop and use Custom Jsp taglibrary in our jsp page

step1) Design jsp taglibrary

```
NITTaglibrary
    |----> <ABC> tag
    |----> <XYZ> tag
```

step2) Develop Handler tag handler classes for Custom jsp tags in WEB-INF/classe folder

ABCTag.java (In WEB-INF/classes folder)

```
package com.nt.tags;
public class ABCTag extends TagSupport{

   public int doStartTag(){  //executes automatically for open <ABC> tag
     ... //logic for opne <ABC> tag
     ...
    }
    public int doEndTag(){  //executes automatically for closing </ABC> tag
     ... //logic for closing </ABC> tag
     ...
    }
}
```

doStartTag() and doEndTag() methods are callaback methods becoz they will be executed automatically for open and closing jsp tags..

XyzTag.java (In WEB-INF/classes folder)

```
package com.nt.tags;
public class XyzTag extends TagSupport{

   public int doStartTag(){  //executes automatically for open <Xyz> tag
     ... //logic for opne <Xyz> tag
     ...
    }
    public int doEndTag(){  //executes automatically for closing </Xyz> tag
     ... //logic for closing </Xyz> tag
     ...
    }
}
```

(f) (h) (h)

doStartTag() and doEndTag() methods are callaback methods becoz they will be executed automatically for open and closing jsp tags..

step3) Develop TLD file having deteails about the jsp tags.

WEB-INF/nit.tld

```
<ABC> ----------> com.nt.tags.ABCTag class
<XYZ> ----------> com.nt.tags.XyzTag class
```

(e) Orginally , this info will be there in the form of xml content

step1 to step3 :: complets the development of custom jsp taglibrary.

step4) cfg jsp library in web.xml file and generate tag lib uri

web.xml

```
<web-app>
  <jsp-config>
    <taglib>
      <taglib-uri>http://nareshit.com/tags/nit</taglib-uri>
      <taglib-location>/WEB-INF/nit.tld </taglib-location>
    <taglib>
  </jsp-config>
</webapp>
```

(d) Jsp taglibrary cfg ... having taglib uri..

step5) import jsp taglibrary to the jsp page.. and use tags in that jsp page..

test.jsp

```
<%@taglib uri="http://nareshit.com/tags/nit" prefix="n" %>  (importing the jsp taglib url)
                                                              (c)
<n:ABC/>
<n:XYZ >
```

(b) (a) utilizing the recods..

What is the use of prefix?
=>if working multipl third party jsp taglibraies.. there is possility of having two tags with same name and different functionalaties.. So to differentiate one tag from another tag .take the support of prefixes.

=>The callback methods of tag handler givess instruction to jsp container to specify the next operation that it has to perform

javax.servlet.jsp.tagext.Tag

```
public static final int EVAL_BODY_INCLUDE 1
public static final int EVAL_PAGE 6
public static final int SKIP_BODY 0
public static final int SKIP_PAGE 5
```

SKIP_BODY :: Tag is an empty.. so do not process body
EVAL_PAGE :: say evaluate the remainig jsp page
SKIP_PAGE :: stop executing jsp page
EVAL_BODY_INCLUDE :: include the body..

=> Every jsp tag handler class gets pageContext object as the inheritted , object (as protected member valable of TagSupport class) and tag hanlder class can use that pageContext object to access to other jsp obejcts like out,response,equest,out, config, servlet context, and etc..

Example Application

(a) <nit:label --> prints WElcome to nit
<nit:label msg="naresh it"> --> prints naresh it

An empty tag (tag with out body) with optional parm (msg) "msg" default value is " Welcome to nit "

(b) <nit:prime --> should print prime number b/w 1 to 10
<nit:prime n=30> --> should print prime number b/w 1 to 30

An empty tag with optional param "n" "n" default value is "10"

(c)

```
<nit:display font="arial" size="50">
      hello
</nit:display>
```

--> prints "hello" on the browser having font "arial" and size 50px

```
<nit:display font="verdana" >
      hello
</nit:display>
```

--> prints "hello" on browser having font "verdana" and size 20px (default size 20)

Tag with body having 1 mandatory attribute (font) and optional attribute size

JspCustTaglibApp

```
|---->java resoruces
      |-->src
          |--->com.nt.tags
               |--->LabelTag.java
               |--->PrimeTag.java
               |--->DisplayTag.java
|----->webcontent
      |-----> test.jsp
      |-----> WEB-INF
               |-----> nit.tld      refer JspApp17-CustTaglibApp ..
               |-->web.xml
               |-->lib
                   |-->jsp-api.jar
                   |-->servlet-api.jar
```

jar files in build path
servlet-api.jar,jsp-api.jar

Instead of writing taglib uri for jsp library in web.xml file of every web application.. we can write only for 1 time in tld file itself using <uri> tag as shown below..

```
<uri>https://nareshit.com/custom/tags</uri>
```

note: if write the taglib uri in both palces (in tld file and also in web.xml file) then both will be taken.. i.e we can import that jsp taglibrary using any taglib uri..

in web.xml

```
<jsp-config>
  <taglib>
    <taglib-uri>http://nit.com/tags</taglib-uri>
    <taglib-location>/WEB-INF/nit.tld</taglib-location>
  </taglib>
</jsp-config>
```

note:: in new versions of html ... the <font> tagdeprecated and given , tags..
=> tag is useful to apply styles on 1 line of text
=> tag is useful to apply styles on multiple lines of text.

EL --Expression Language (one more technique of reducing java code from jsp page)
==========
=> It is expression tag.. i.e no only evaluates expression .. also writes the generated messages to browser
note:: Performing airthmetic and logical operations with out using java code ..really not possible..To make it possible we got EL

EL is composed with 3 parts
(a) EL Operators
(b) EL Implicit objs
(c) EL functions

To enable or disable EL
==========
<%@page isELIgnored="true"%> --> Ignores EL
eg:: ${4+5} -->gives ${4+5}

<%@page isELIgnored="false"%> --> recognizes EL
eg: ${4+5} --> gives 9

EL Operators
==========
=> same as java operators but some operators are even having english words
< ---> lt
> -- > gt
<= --> le
>= --> ge
!= --> ne
== --> eq
and etc..

Arithmetic: +, - (binary), *, / and div, % and mod, - (unary)
Logical: and, &&, or, ||, not, !
Relational: ==, eq, !=, ne, <, lt, >, gt, <=, ge, >=, le. Comparisons can be made against other values, or against boolean, string, integer, or floating point literals.
Empty: The empty operator is a prefix operation that can be used to determine whether a value is null or empty.
Conditional: A ? B : C. Evaluate B or C, depending on the result of the evaluation of A.
The precedence of operators highest to lowest, left to right is as follows:
[] .
() (used to change the precedence of operators)
- (unary) not ! empty
* / div % mod
+ - (binary)
< > <= >= lt gt le ge
== != eq ne
&& and
|| or
? :

EL Implicit objects
==========
=> EL gives another 10 implicit objs on the top of reqular 9 implicit objs of jsp page..
=>these addtional objects are also useful to make jsp page as java code less jsp page..
=> pageScope , requestScope, sessionScope,applicationScope, header,headerValues, param,paramValues, initParam,cookie

xxxScope
==========
=> To read display data from differet scopes..
=>useCase:: In MVC2 architecture Controller SErvlet passes data to jsp pages in scope to read and display them with out writing java code we can use requestScope object.

syn :: ${xxxScope.<attribute>} --> Internally uses pageContext.getAttribute(-,-) method.. | searches in specificed scope
${<attribute>} --> internally uses pageContext.findAttribute(-) mehtod | seaches in multiple scopes (lower to higher :: page,request,session,application)

In SErvlet comp
==========

```
@WebServlet("/testurl")
public class TestServlet extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
   HttpSession ses=null;
   ServletContext sc=null;
   RequestDispatcher rd=null;
            //create diff scope attributes
            //req attribute
     req.setAttribute("attr1","val1");
    //ses attribute
            ses=req.getSession(true);
            ses.setAttribute("attr2","val2");
            //application attribute
            sc=getServletContext();
            sc.setAttribute("attr3","val3");

            //forward request to jsp page
            rd=req.getRequestDispatcher("/el_test.jsp");
            rd.forward(req, res);
    }

    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    doGet(req,res);
    }

}
```

in el_test.jsp

```
<%@page isELIgnored="false"%>
request scope attribute value :: ${requestScope.attr1} <br>
session scope attribute value :: ${sessionScope.attr2} <br>
application scope attribute value :: ${applicationScope.attr3} <br>
<br><br>
request scope attribute value :: ${attr1} <br>
session scope attribute value :: ${attr2} <br>
application scope attribute value :: ${attr3} <br>
```

---

param, paramValues
==========
=>Given to read and display req param values

"param" for reading single value of req param
"paramValues" for reading multiple values of each req param

```
uname req param value :: ${param.uname } <br>
addrs req param multipe values :: ${paramValues.addrs[0]}, ${paramValues.addrs[1]}
```

url :: http://localhost:3030/JspApp18-EL/el_test.jsp?uname=raja&addrs=hyd&addrs=vizag

header, headerValues
==========
=>given to read and display req header values.. like user-agent, accept , accept-language and etc..

```
user-agent req header value :: ${header['user-agent']} <br>
accept req header values :: ${headerValues.accept[0]}, ${headerValues.accept[1]} <br>
```

---

cookie
==========
=>To read and display cookie value create in various web comps

```
cookie name :: ${cookie.JSESSIONID.name } <br>
cookie value :: ${cookie.JSESSIONID.value } <br>
```

output ::
cookie name :: JSESSIONID
cookie value :: C638BFE536124E272361883A68D184C0

---

initParam
==========
=>Given to read and display Context param (global init param) values..

in web.xml
==========

```
<context-param>
  <param-name>dbuser</param-name>
  <param-value>system</param-value>
 </context-param>
 <context-param>
  <param-name>dbpwd</param-name>
  <param-value>manager</param-value>
 </context-param>
```

in jsp page

```
dbuser init param value :: ${initParam.dbuser } <br>
dbpwd init param value :: ${initParam.dbpwd } <br>
```

EL Functions
==========
=>EL functions are like Jsp custom tags which are given to execute java code.. when function called..This also help either to minimize or to avoid java code from jsp page..

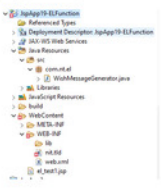
step1) create java class having static method based functionality/logic
=====
WishMessageGenerator.java

```
package com.nt.el;
public class WishMessageGenerator {
      public static String generateMessage(String user) {
            return "GoodMorning ::"+user;
      }
}
```

must be static method

step2) develop tld file configuring el function and generating taglib uri
WEB-INF/nit.tld
==========

```
<taglib xmlns="http://java.sun.com/xml/ns/j2ee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd" version="2.1">
        <tlib-version>2.0</tlib-version>
        <uri>http://nit.com/el</uri>

        <function>
          <name>sayHello</name>
          <function-class>com.nt.el.WishMessageGenerator</function-class>
          <function-signature>java.lang.String generateMessage(java.lang.String)</function-signature>
        </function>

        </taglib>
```



step3) import taglib in jsp page and call the el function..

```
<%@ page isELIgnored="false" %>
<%@taglib uri="http://nit.com/el" prefix="nit" %>

message is :: ${nit:sayHello('raja') }
```

JSTL (Jsp STandad Tag Library)
==========

|--> Gives bunch of jsp taglibraries having jsp tags to make jsp pages as java code less jsp pages..

|--> EL takes care airthmetic and logical oerpations by avoiding java code.... where as JSTL takes care remaining activities like basic control flow activities like if, else, switch case , loops, formatting, xml processing ,DB interaction , string manipulation and etc.. with out any java code..

|--> JSTL +EL is good combination in jsp pages either to avoid java code or to minimize java code..

JSTL is having 5 tag libraries

a) Core ---> tld file :: c.tld --> taglib uri :: http://java.sun.com/jsp/jstl/core -->For core operations (variable, control flow operations)
b) sql --------> tld file :: sql.tld ---> taglib uri :: http://java.sun.com/jsp/jstl/sql --> for DB interaction
c) Formatting --> tld file :: fmt.tld ---> taglib uri :: http://java.sun.com/jsp/jstl/fmt ---> For enabling I18n (Internationlization)
d) xml -------> tld file :: x.tld ----> taglib uri :: http://java.sun.com/jsp/jstl/x ----------> for xml processing
d) functions --------> tld file ::functions.tld ----> taglib uri :: http://java.sun.com/jsp/jstl/functions ----------> for String manipulation

=> JSTL Taglibraries and its tags are designed by Sun Ms and implemetations are given by Server Vendors in the form of jar files having tag handler classes..
In Tomcat sever (upto 8.x) jar files are :: jstl.jar , standard.jar
In Tomcat sevrer (9.x) jar files are :: taglibs-standard-impl-1.2.5.jar, taglibs-standard-spec-1.2.5.jar
(<Tomcat_home>\webapps\examples\WEB-INF\lib folder)

JSTL Core Taglibrary
==========

=>gives basic tags of programming like variable decl, scopes, control flow and others...
http://java.sun.com/jsp/jstl/core (taglib uri) recomanded prefix :: c

Tags	Description
c:out	It display the result of an expression, similar to the way <%=...%> tag work.
c:import	It Retrives relative or an absolute URL and display the contents to either a String in 'var',a Reader in 'varReader' or the page.
c:set	It sets the result of an expression under evaluation in a 'scope' variable.
c:remove	It is used for removing the specified scoped variable from a particular scope.
c:catch	It is used for Catches any Throwable exceptions that occurs in the body.
c:if	It is conditional tag used for testing the condition and display the body content only if the expression evaluates is true.
c:choose, c:when, c:otherwise	It is the simple conditional tag that includes its body content if the evaluated condition is true.
c:forEach	It is the basic iteration tag. It repeats the nested body content for fixed number of times or over collection.
c:forTokens	It iterates over tokens which is separated by the supplied delimeters.
c:param	It adds a parameter in a containing 'import' tag's URL.
c:redirect	It redirects the browser to a new URL and supports the context-relative URLs.
c:url	It creates a URL with optional query parameters.

<c:out>
=======
=> To evaluate the expression and to write to the browser
=> can also write static content to browser

=>The JSTL tags related tld file, tag halder clases are ready made in the jar files supplied by server vendors

<c:set>
======
To define variable with data having scope

<c:remove>
==========
to remove variable from the specified scope

JspApp20-JSTLCore
|--->webcontent
|--->core1.jsp
|--->WEB-INF
|--->lib
|-->*.jar(2)
|-->web.xml

Bullpath
jar files in ClassPATH and in WEB-INF/lib folder
==========
taglibs-standard-impl-1.2.5.jar, taglibs-standard-spec-1.2.5.jar

core1.jsp

```
<%@ page isELignored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!-- create data having request scope internall uses pageContext.setAttribute(,,,) -->
<c:set var="msg" value="welcome to jstl" scope="request"/>
<!-- Display data internally uses pageContext.findAttribte() -->
value: <c:out value="${msg}" />
<!-- Remove data from scopes -->
<c:remove var="msg"/>
<!-- Display data internally uses pageContext.findAttribte() -->
<br> value is:<c:out value="${msg}" />
```

<c:if>
=====
note: we can EL , JSp implicit objs in JSTL

=> allows to perform tradtional if condtion...
=>there is no <c:else> tag...
=>To bring the if-- else if -- else if -- else we need to use <c:choose> ,<c:when> and <c:otherwise> tags

core2.jsp

```
<%@ page isELignored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<c:set var="msg" value="welcome to JSTL" scope="request"/>
<c:if test="${!empty param.uname }">
  <c:out value="${msg}"/> Mr/Miss/Mrs. <c:out value="${param.uname}"/> <br>
  ${msg} Mr/Miss/Mrs. ${param.uname } <br>
</c:if>
```

EL Implicit object

url :http://localhost:3030/JspApp20-JSTLCore/core2.jsp?uname=raja

---

<c:choose> ,<c:when>,<c:otherwise>
==========

=> To bring the effect of if... else if..else tags
=>To even the effect of swtich--case..

```
<%@ page isELignored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<c:choose>
  <c:when test="${param.p >0 }">
    ${param.p } is positive
  </c:when>
  <c:when test="${param.p <0 }">
    ${param.p } is negetive
  </c:when>
  <c:otherwise>
    ${param.p } is Zero
  </c:otherwise>
</c:choose>
```

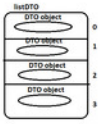
<c:forEach>
=> We can use this tag to behave like tradtional for loop or enhanced for loop

```
<%@ page isELignored="false" import="java.util.*"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!-- Traditional for loop -->
<table border="2" align="center" bgcolor="cyan">
<c:forEach var="i" begin="2" end="10" step="2">
  <tr>
    <td>2 * ${i}= </td> <td>${2*i} </td>
  </tr>
</c:forEach>
</table>
<br><br>
<jsp:scriptlet>
  String studNames[]={"raja","rani","suresh","mahesh"};
  request.setAttribute("namesArray",studNames);
</jsp:scriptlet>
<!-- Enhaced for loop -->
<c:forEach var="name" items="${namesArray}">
    ${name } <br>
</c:forEach>

<jsp:scriptlet>
 <![CDATA[
  List<Long> phonesList=List.of(9999999L,8888888L,7777777777L); //java9
  request.setAttribute("phonesInfo",phonesList);
  ]]>
</jsp:scriptlet>

<!-- Enhaced for loop -->
<c:forEach var="ph" items="${phonesInfo}">
    ${ph} <br>
</c:forEach>
```

With respect to Mini Project html_print.jsp code with JSTL Core tags
==========

```
<br><br>
<c:choose>
  <c:when test="${booksInfo ne null && !empty booksInfo}">
    <center> <b> Books belonging to :::: ${param.category} </b> </center><br>
    <table border="1" align="center" bgcolor="cyan">
      <tr>
        <th>SerialNo </th> <th>BookId </th> <th>BookName </th><th>Author </th><th>Publisher </th><th>Price </th> <th>status </th> <th>category </th>
      </tr>
      <c:forEach var="dto" items="${booksInfo}">
        <tr>
            <td>${dto.serialNo}</td>
            <td>${dto.bookId} </td>
            <td>${dto.bookName}</td>
            <td>${dto.author} </td>
            <td>${dto.publisher}</td>
            <td>${dto.price}</td>
            <td>${dto.status}</td>
            <td>${dto.category} </td>
          </tr>
      </c:forEach>
    </table>
  </c:when>
  <c:otherwise>
      <h1 style="color:red;text-align:center">No Books found to display </h1>
  </c:otherwise>
</c:choose>
    <script language="JavaScript" src="js/search.js"/>
    <a href="JavaScript:doPrint()">print</a>
    <br>
    <br>
    <%@include file="footer.html" %>
```

listDTO



---

c:forTokens
==========
=>Given to split the string multiple parts based on the given delimeter..

```
<%@ page isELIgnored="false" import="java.util.*"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>
 <c:set var="msg" value="hello , how are u " scope="request"/>
<c:forTokens var="tkn" items="${msg}" delims=" ">
    ${tkn} <br>
</c:forTokens>
```

Internally uses split() method
of String clss.

<c:url> , <c:redirect>
==========
<c:url>:: to define url having scope and name
<c:redirect>:: To perform sendRedirection

```
<%@ page isELIgnored="false" import="java.util.*"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>
 <c:url var="googleUrl"  value="https://google.com/search" scope="request" />
  <c:redirect url="${googleUrl}"/>
```

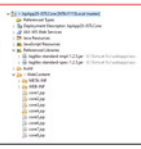
<c:import>
==========
=> To import one jsp/html file code in another jsp page/file..
=> It is like directive include..

ex7.jsp
=======

```
<%@ page isELIgnored="false" import="java.util.*"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>

  <b> from ex7.jsp(start)</b>
    <c:import url="xyzred.jsp"/>
  <b>from ex7.jsp(end)</b>
```

<c:catch>

=>Given for catching and handling exceptions..



```
<%@ page isELIgnored="false" import="java.util.*"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>
<c:catch var="e">
  <jsp:scriptlet>
    int x=Integer.parseInt("10");
  </jsp:scriptlet>
   x value :: <jsp:expression>x</jsp:expression>
</c:catch>
<c:if test="${e ne null}">
   <b> the raised exception is :: ${e}</b>
</c:if>
```

JSTL SQL TagLibrary

=>These tags are given DB connectivity to to manipulate DB Data...
=>Useful only model1 architecture based web application development.. not in mvc1_mvc2
becoz writing persistence logic in jsp pages is bad according to mvc1_mvc2 architectures .

taglib uri is ::http://java.sun.com/jsp/jstl/sql

1	<sql:setDataSource />	(to establish the connection)
	Creates a simple DataSource suitable only for prototyping	
2	<sql:query />	(select queries)
	Executes the SQL query defined in its body or through the sql attribute	
3	<sql:update />	(non-select queries)
	Executes the SQL update defined in its body or through the sql attribute	
4	<sql:param />	
	Sets a parameter in an SQL statement to the specified value	
5	<sql:dateParam />	
	Sets a parameter in an SQL statement to the specified java.util.Date value	
6	<sql:transaction />	
	Provides nested database action elements with a shared Connection, set up to execute all statements as one transaction.	

sql1.jsp
=======

```
<%@ page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/sql"  prefix="sql"%>

<!-- establish the connection -->
<sql:setDataSource  var="ds" driver="oracle.jdbc.driver.OracleDriver"
                    url="jdbc:oracle:thin:@localhost:1521:xe"
                    user="system"
                    password="manager"/>
<!-- Execute the SQL query and create RS -->
<sql:query var="rs"  sql="SELECT EMPNO,ENAME,JOB,SAL FROM EMP" dataSource="${ds}"/>

<b>All records</b> <br>
<c:forEach  var="row"  items="${rs.rows}">
  ${row.EMPNO } , ${row.ENAME } , ${row.JOB } , ${row.SAL } <br>
</c:forEach>
```

---

sql2.jsp
=======

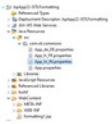
```
<%@page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/sql"  prefix="sql"%>

<!-- establish the connection -->
<sql:setDataSource  var="ds" driver="oracle.jdbc.driver.OracleDriver"
                    url="jdbc:oracle:thin:@localhost:1521:xe"
                    user="system"
                    password="manager"/>

<!-- execiute non-select Query -->
<sql:update  dataSource="${ds}" var="count" sql="UPDATE EMP SET SAL=SAL+? WHERE JOB=?">
  <sql:param>1000</sql:param>
  <sql:param value="CLERK"/>
</sql:update>

no.of records that are effected :: ${count}
```

JSTL Formatting Tag Library

=> These are given enable i18n(Internationalization) on jsp pages
=>Useful to display date values , numbers , currency values, labels and etc.. according the
current /choosen Locale

taglib uri ::http://java.sun.com/jsp/jstl/fmt

fmt:parseNumber	It is used to Parses the string representation of a currency, percentage or number.
fmt:timeZone	It specifies a parsing action nested in its body or the time zone for any time formatting.
fmt:formatNumber	It is used to format the numerical value with specific format or precision.
fmt:parseDate	It parses the string representation of a time and date.
fmt:bundle	It is used for creating the ResourceBundle objects which will be used by their tag body.
fmt:setTimeZone	It stores the time zone inside a time zone configuration variable.
fmt:setBundle	It loads the resource bundle and stores it in a bundle configuration variable or the named scoped variable.
fmt:message	It display an internationalized message.
fmt:formatDate	It formats the time and/or date using the supplied pattern and styles



formatting1.jsp

```
<%@ page isELIgnored="false" contentType="text/html; charset=UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<fmt:setLocale value="hi"/>
<jsp:useBean id="dt" class="java.util.Date"/>
<fmt:formatDate value="${dt}" var="fdt" type="both" dateStyle="FULL"/>
formatted date :: ${fdt} <br>
<hr>
<fmt:formatNumber  value="1234567890" var="fno" type="currency" />
formatted number :: ${fno }
<hr>
<fmt:formatNumber  value="1234567.55" var="fno1" type="number" />
formatted number :: ${fno1 }
<hr>
<fmt:formatNumber value="0.3" var="fno2" type="percent" />
formatted number :: ${fno2 }
<hr>
 <fmt:setBundle basename="com/ntj/commons/App"/>
 <fmt:message key="wish.welcome" var="fwelcome"/>
 <fmt:message key="wish.bye" var="fbye"/>
   wellcome message :: ${fwelcome } <br>
   bye message :: ${fbye}
```

App.properties
# base file for english
wish.welcome=Good morning
wish.bye=See you Later

App_fr_FR.properties
# file for fr_FR (french)
wish.welcome=Bonjour
wish.bye=à plus tard

App_de_DE.properties
# file for de_DE (german)
wish.welcome=guten Morgen
wish.bye=bis später

App_hi_IN.properties
# file for hi_IN (german)
wish.welcome=\u0938\u0941\u092D \u092A\u094D\u0930\u092D\u093E\u0924
wish.bye=\u092C\u093E\u0926 \u092E\u0947\u0902 \u092E\u093F\u0932\u0924\u0947\u0902\u0938\u0947

note:: basefile name should reflefect in all file names
note:: All files should have same keys and different vlaues
(values can be collected from google translator)
note: we must enable UTF-8 as charset in contentType
note: All files must have .properties extension

note:: if no matching file is found. then base file will execute automatically

---

JSTL Functions Taglibrary
==========
-> Gives bunch of jsp EL functions for String manipulation in this taglibrary
-> taglib uri :: http://java.sun.com/jsp/jstl/functions

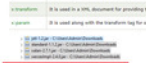
fn:contains()	It is used to test if an input string containing the specified substring in a program.
fn:containsIgnoreCase()	It is used to test if an input string contains the specified substring as a case insensitive way.
fn:endsWith()	It is used to test if an input string ends with the specified suffix.
fn:escapeXml()	It escapes the characters that would be interpreted as XML markup.
fn:indexOf()	It returns an index within a string of first occurrence of a specified substring.
fn:trim()	It removes the blank spaces from both the ends of a string.
fn:startsWith()	It is used for checking whether the given string is started with a particular string value.
fn:split()	It splits the string into an array of substrings.
fn:toLowerCase()	It converts all the characters of a string to lower case.
fn:toUpperCase()	It converts all the characters of a string to upper case.
fn:substring()	It returns the subset of a string according to the given start and end position.
fn:substringAfter()	It returns the subset of string after a specific substring.
fn:substringBefore()	It returns the subset of string before a specific substring.
fn:length()	It returns the number of characters inside a string, or the number of items in a collection.
fn:replace()	It replaces all the occurrence of a string with another string sequence.

functions1.jsp
==========

```
<%@ page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>

  <c:set var="msg" value="welcome nit"/>
  uppercase :: ${fn:toUpperCase(msg) }<br>
  lowercase :: ${fn:toLowerCase(msg) }<br>
  length :: ${fn:length(msg) }<br>
  substring :: ${fn:substring(msg,0,5) } <br>
```

JSTL Xml taglibrary
==========
=> Given to manipuate Xml content.. that collected from files or collected from response xml
=> while working with Ajax/jquery/Angularjs/angular there is possibility of getting xml response..
So , to format such response in our jsp page use xml tag library
=>taglib uri :: http://java.sun.com/jsp/jstl/xml

x:out	Similar to <%= ... > tag, but for XPath expressions.
x:parse	It is used for parse the XML data specified either in the tag body or an attribute.
x:set	It is used to sets a variable to the value of an XPath expression.
x:choose	It is a conditional tag that establish a context for mutually exclusive conditional operations.
x:when	It is a subtag of that will include its body if the condition evaluated be 'true'
x:otherwise	It is subtag of that follows tags and runs only if all the prior conditions evaluated be 'false'.
x:if	It is used for evaluating the test XPath expression and if it is true, it will processes its body content.
x:transform	It is used in a XML document for providing the XSL(Extensible Stylesheet Language) transformation.
x:param	It is used along with the transform tag for setting the parameter in the XSLT style sheet.



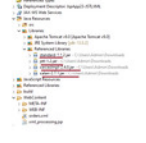
{Download from internet}
{mvnrepository.com}

orders.xml
=========

```
<?xml version="2.0" encoding="UTF-8"?>
<orders>
  <order>
    <name>table</name>
    <price>4000</price>
  </order>
  <order>
    <name>chair</name>
    <price>400</price>
  </order>
  <order>
    <name>sofa</name>
    <price>3000</price>
  </order>
  <order>
     <name>tv</name>
     <price>40000</price>
  </order>
</orders>
```

xml_processing.jsp

```
<%@ page isELIgnored="false"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core"  prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/xml"  prefix="x"%>

<!-- Load xml -->
<c:import url="orders.xml" var="file"/>

<!-- parse xml file -->
<x:parse var="doc" doc="${file}"/>

<!-- Display all orders -->
<x:forEach var="ord" select="$doc/orders/order">
   <x:out select="$ord/name"/>    
   <x:out select="$ord/price"/> <br>
</x:forEach>

<hr>
<!-- Display all orders whose price>3000 -->
<x:forEach var="ord" select="$doc/orders/order">
   <x:if select="$ord/price>=3000">
      <x:out select="$ord/name"/>    
      <x:out select="$ord/price"/> <br>
   </x:if>
</x:forEach>
```



Popularly used JSTL Libraries are :: Core ,Formatting, functions