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

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

Dynamic webpage of first.jsp

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

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

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.

step4) check the web application.

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

=>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)
==========

of

(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..

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

scriptlet

->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

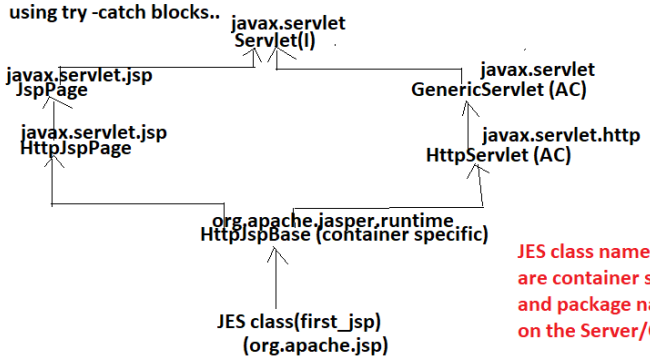
note: jspInit() , jspDestroy() methods will come in JES class only when they are defined by prgrammer in jsp file..

=>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

==========

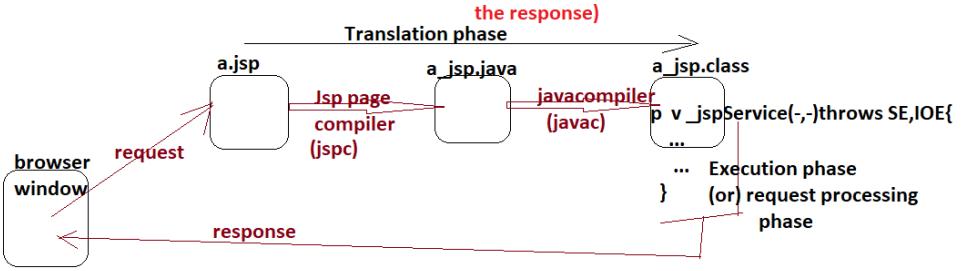
Container calls destroy() method on JES class obj ---> since not available in JES class ---> the destroy() method of JES super class will execute --> and that internally calls jspDestory() and _jspDestroy() methods ---> jspDestroy() of JES super class executes and _jspDestroy() method of JES class will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

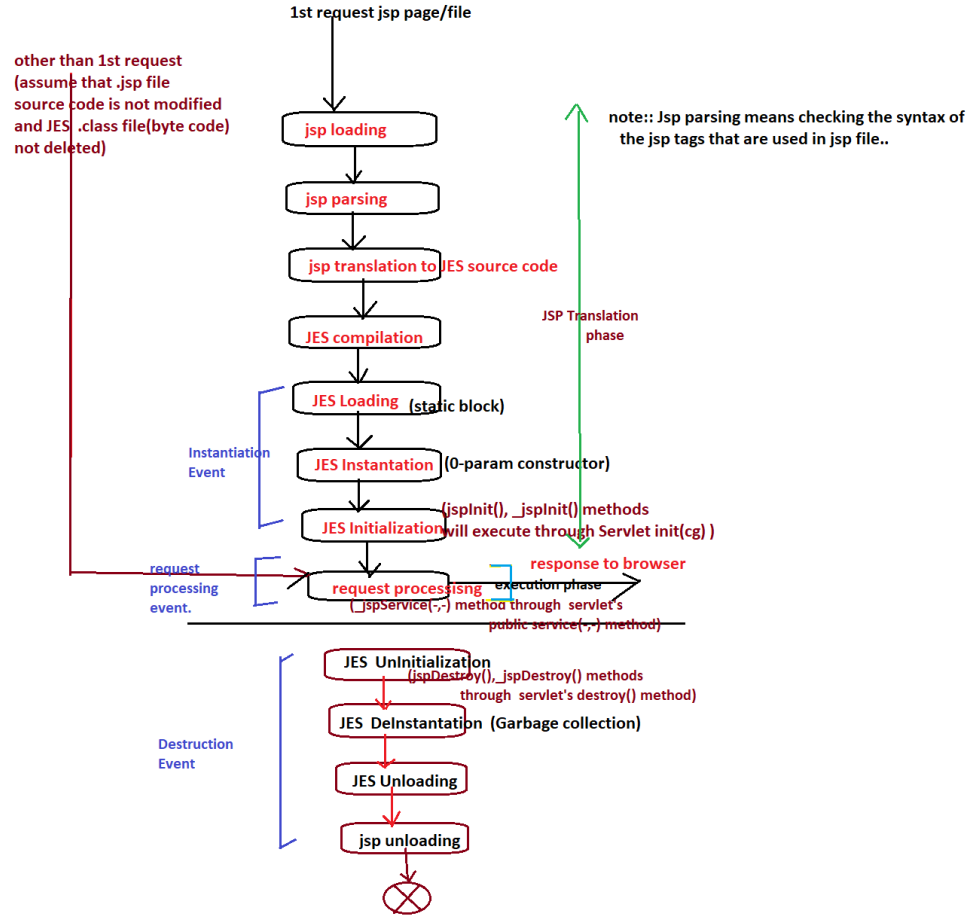


=>The request given to jsp page participates directly request processing phase/execution phase if the source code jsp page is not modified before the request and the byte code JES class is already availble otherwse the request given jsp page first participates translation and later particpated in request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

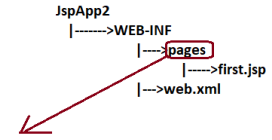
note:: if we just delete soruce file of JES class.. then also next request directly particpates in execution phase.

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

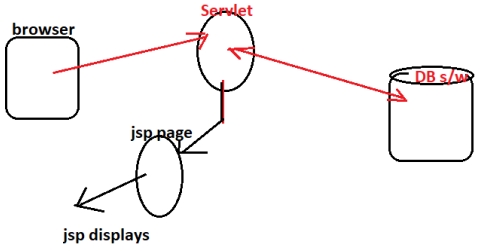
What is the advnatage of placing jsp page in private area?

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...

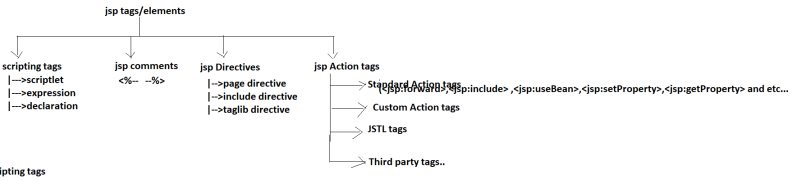
=> To get automatic <load-on-startup> advantages (we should cfg in web.xml)

=> 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 effectively, 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 Demonstrating 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 about using servlet life cycle methods directly.. with out using jsp life cycle methods...

think

```
<%!  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>
```

expresssion tag

template text

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...

---

jsp Scopes in 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: Test — Object type: Test

Object obj = new Test();
reference type: Object — Object type: Test

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

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....

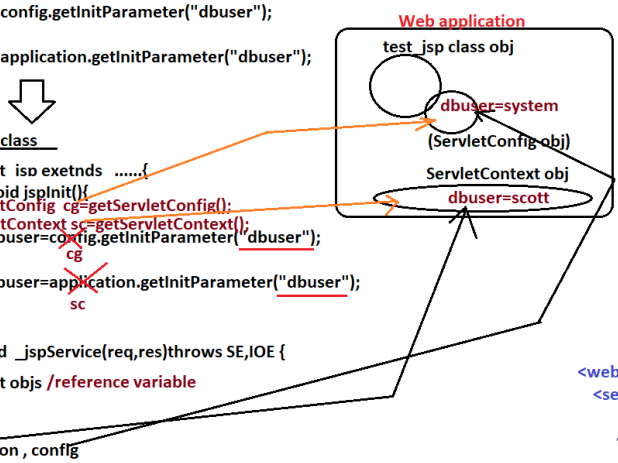
note:: In realtime, we use unchecked exceptions most of time to enjoy exception propagation/ passing in layered applications

---

test.jsp
=======

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

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



JES class

```
public class test isp exetnds .....{
      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 , config
        ...
        }
    }
```

cg
sc

=>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 ,ServletConfing, 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

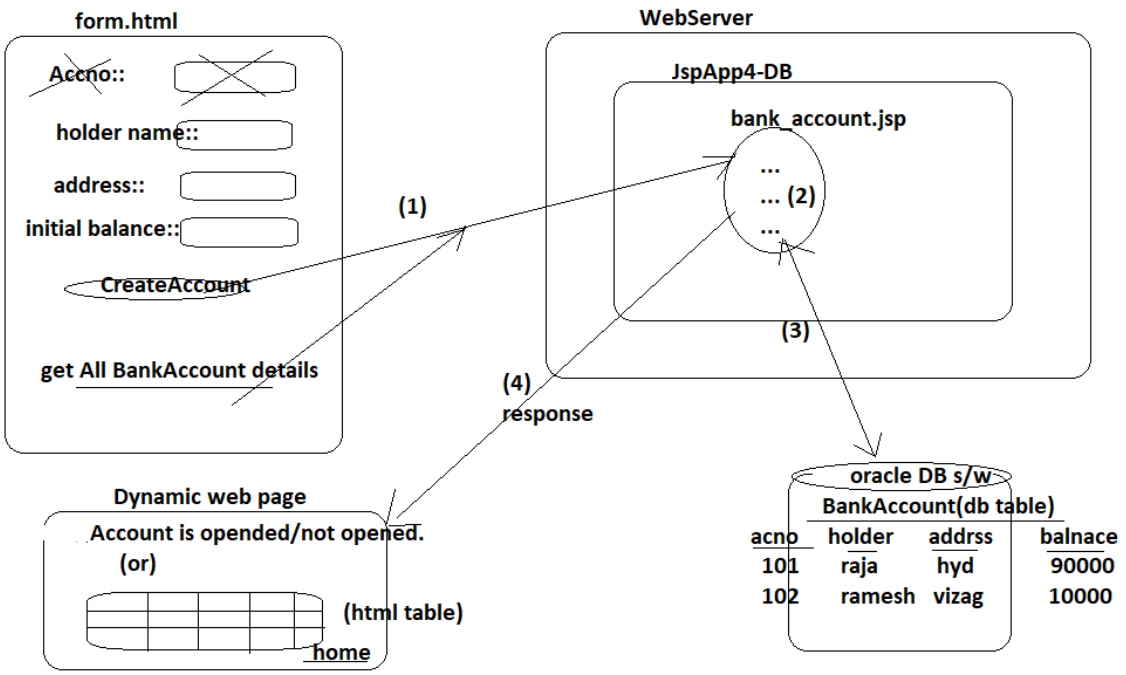
---

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
```

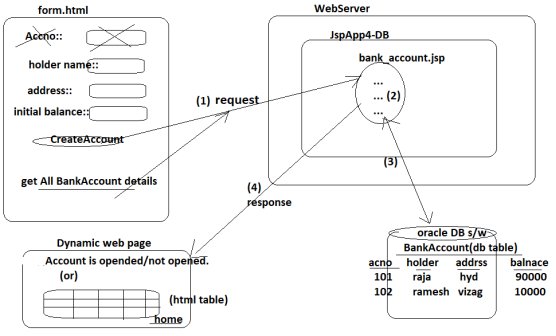
html to jsp to Db s/w

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,@p, 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 *(attribute 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	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 wh <%@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 to handle multithreading manually by synchronization concept..

using

```
<% 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)

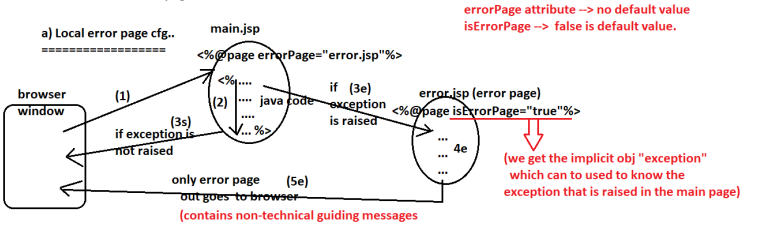
use <error-page> of web.xml file.

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..

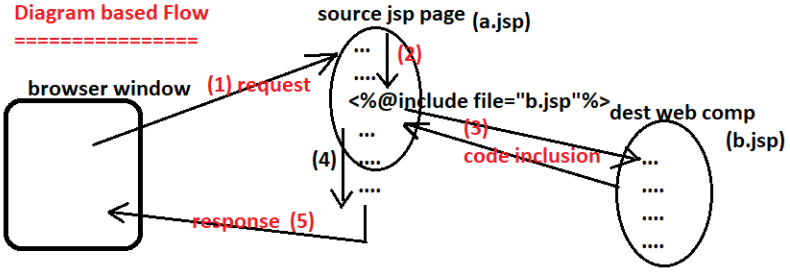
=> This tag does not use rd.include(-,-) internally ..

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

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

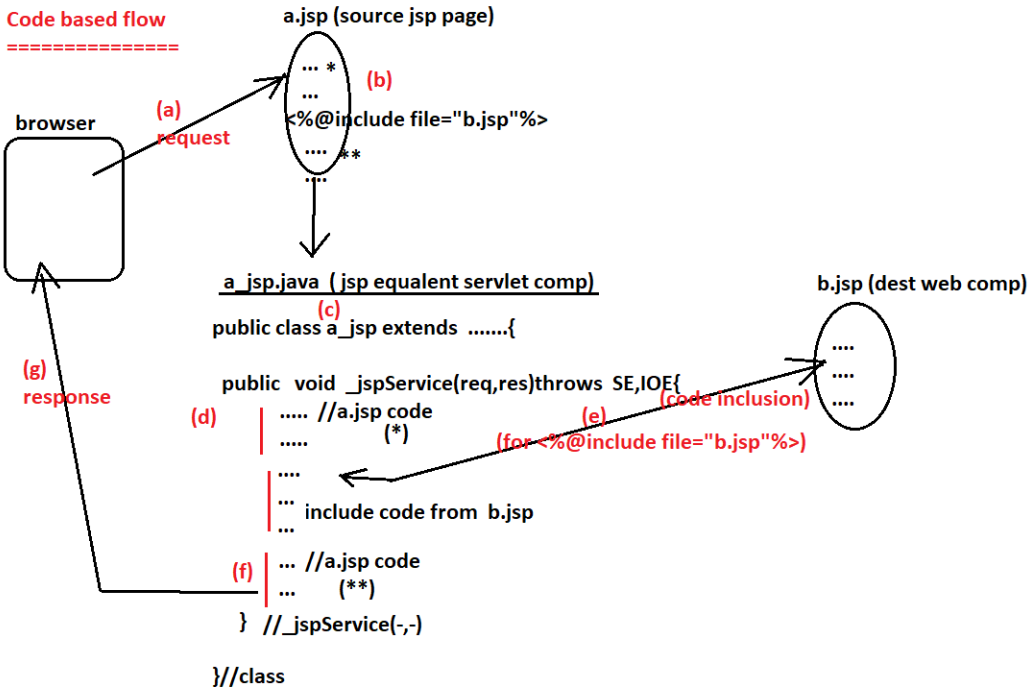
Diagram based Flow

==========



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 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)

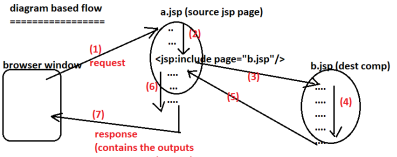
List of jsp STandard Action tags
==========
<jsp:include> ,<jsp:forward> ,<jsp:useBean> ,<jsp:setProperty>,<jsp:getProperty>, <jsp:plugin> (old) , <jsp:fallback>(old) , <jsp:param> and etc..

<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..

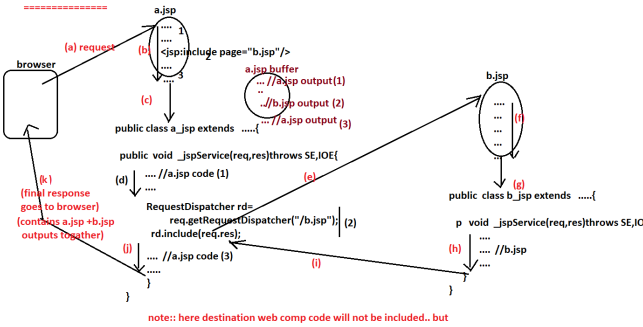
=> syntax <jsp:include atttributes />

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..

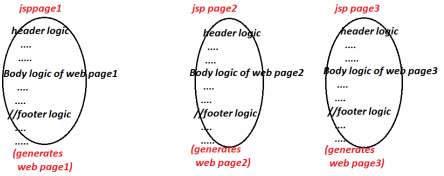
what is the diff b/w directive include and Action include?

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"%>

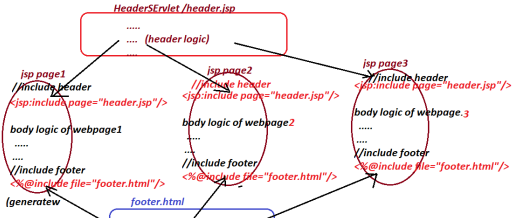
To specify wheahter source jsp page buffer should be flushed or not before including the output of dest web comp. *(true/false (default))*

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
==========



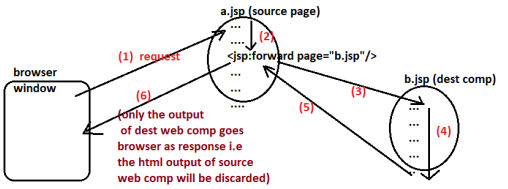
note:: the code that repeates across multiple parts of Project either with no chanages or with minor changes is called boiler plate code program.

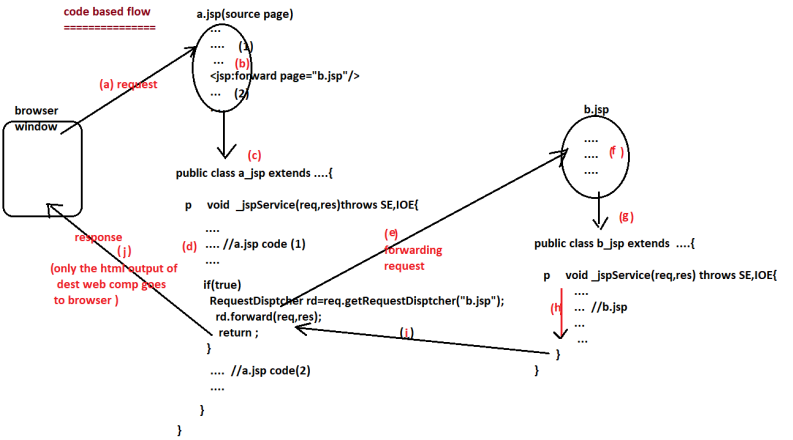
note:: here header,footer logics are reusable and there is no boiler plate code problem
note: here every web page comes having the outputs given by multiple web comps.. So it is composite view design pattern

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..

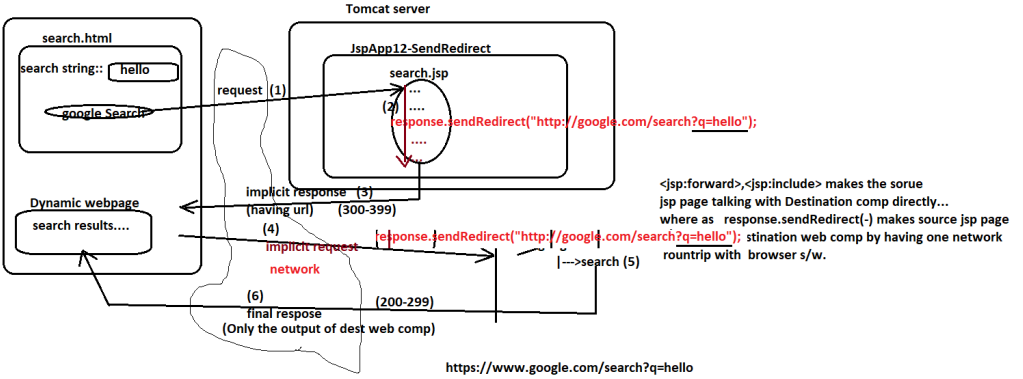
if source jsp page and and dest web comp are there in the same web application
==========
a) <jsp:include> (for Including response mode of jsp communication)
b) <jsp:forward> (for forwarding request mode of jsp communication)

=>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..

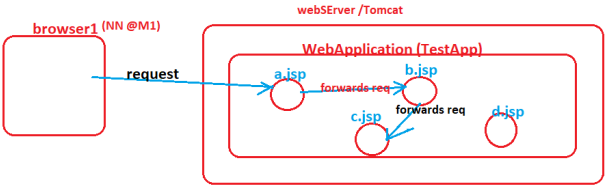
and using same browser of same client machine
==========
=>use session attribute (keeps in session scope)

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..

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)

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

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..