List of server side web technologies

=>servlet from sunMs/oracle corp
=> jsp from sunMs/oracle corp
=> Asp.net from MicroSoft
=> Asp from MicroSoft
=> PHP from apache
=> COLD Fusion from adobe
=> Node JS + JAva Script

=>asp.net is extention of asp technology
=> jsp is extension of servlet technology

When Sun Ms is having Servlet as servlet side web technology why did they give jsp as another server side web technology?

Ans) In the intial days servlet launch , the programmers like servlet fetaure but did not move to Servlet by leaving asp technology in web application development becoz asp supports tag based programming and to work with servlet strong java knowledge is required and no supports for tags based programming. So Sun Ms had released JSP having support for tag based programming by inspiring from asp programming but Every Jsp programming is internally a translated servlet comp , So we can use all fetaures of servlet in jsp programming having the benifit of tag based programming.

ASP :: Active Server pages
JSP :: Java server pages..

Limitations of SErvlet

a) Strong java knowlege is required to work with servlets
b) Not suitable for non-java programmers
c) No Implicit objs support
d) Writing html code (presentation logic) in java code (b.logic) is error prone process.
e) It forces u to mixup both presentation logic (html code) and b.logic (java code)
pw.println("<h1 style='color:red;text-align:center'>hello </h1>");
f) Modifications done in sevlet comps' source code will reflect only after recompilation of servlet comp and reloading of the web application.
g) Mapping servlet comp with url pattern is mandatory.
h) Servlet technology is bit complex and time consuming
i) No support support for tag based programming..

req,res, servletConfig ,SerletContext and etc.. objs are called Container created object.. not the implicit objs becoz we need to write addtional code to access those objects
=>To access req,res objs we need to override service(-,-)/doXxx(-,-)
=> To access servletConfig obj we need to call getServletConfig()
=> To access servletContext obj we need to call getServletContext() and etc..
=>Implicit objs/reference variables are those objs/variables we can be used directly without writing any addtional code to access them like "this", "super" (implicit ref variables)
=> main(-) method args[] , catch(-) block exception class obj are JVM created objects ..but we can not call them as implicit objs becoz to access to them we need to place main(-) method and also catch(-) block

=>Jsp Latest version :: 2.3 (Internally uses servlet 4.0)
=> To Translate jsp prg/comp/file into equalent servlet comp we need jsp page compiler
=> Jsp Engine/jsp Container provides the " jsp page compiler" and also contains the env.. to execute Jsp equalent servlet comp with the support of servlet container/Servlet engine.

=>WebContainer = servlet container +jsp container

In Tomcat server
=> servlet container name :: CATALINA and jar file catalina.jar (avaliable in <Tomcat_home>\lib folder)
=> jsp container name :: JASPER and jar file jasper.jar
=> jsp page compiler name : Jspc (avalable in jasper.jar)

Jsp features

=>Supports tag based programming
=>Strong java knowledge is not required
=>suitable for both java and non-java programmers
=>Allows to seperate presentation logic (html code) from java code (b.logic)
=>The modifications in jsp page/file/comp will be reflected directly with out reloading the web application and without recompilation
=>we can use all features of servlet technology .becoz every jsp file/comp/page is internally a servlet comp
=> gives 9 implicit objs
=> No need of doing exception handling the JES(Jsp equalent servlet comp) will take care of it
=> gives built -in jsp tags , allows to develop custom jsp tags and also allows to use third party tags..
=> Easy to learn and easy use
=> Can be used independetly to devleop whole web application or can be used togather with other web technologies and etc..

Differences b/w serlvet and jsp

Servlet	jsp
a) To execute servlet comp we need servlet container	a) To execute jsp comp and its equalent servlet comp we need both Servletcontainer and jsp container
b) To work with servlet technlogy strong java knowledge is required	b) Not required
c) does not give implicit objs	c) gives 9 implicit objs
d) make the programmers to mixup both presentation logic (html code) and b.logic(java code)	d) Allows to seperate the logics
e) Not support for tag based programming	e) supports
f) Makes programmer to catch and handle the exceptions	f) Programmer need not to do exception handling becoz jsp equalent servlet comp takes care of it.
g) Servlet comp must be placed in private area and must be mapped with url pattern	g) jsp comp can be placed either in public area or in private area mapping jsp comp with url pattern is optional when it is placed in public area and mandatory when it is placed in private area.
h) suitable only for java programmers	h) suitable for both java and non-java programmers..

note:: In the initial days of jsp entire industry had used only jsp for developing java web applications..
Later they started using both servlet,jsp technologies gogather.

structure of jsp page

<filname>.jsp (jsp page/jsp file/jsp comp/jsp prg)

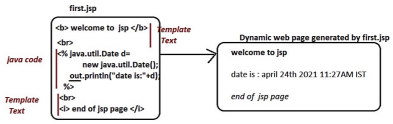
```
<   .....  html code
    .....
    ....  >

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

 .....
 .....  ordindary text
 .....

<% ....
     .....  java code
     ....  %>
 < ....  ordinary text
     .....
     .....  >
```

=>jsp supports free sytle programming

Template Text= html code+ordindary text
(gives static output)

<% .... %> :: scriptlet tag
(Contains java code and gives varying output)

=>Jsp page/file generates dynamic web page .. In that dynamic web page static output comes becoz of template text and dynamic output comes becoz of java code..



Procedure to develope first jsp page based web application

step1) create Deployment Directory strutire directly in <Tomcat_home>\webapps folder..

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

first.jsp
same as above

Since we are not taking any content private area in this "JspApp1" web application , So keeping WEB-INF,classes, web.xml files is optional in "JspApp1" folder.

step2) start the Tomcat server...

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

step3) Test the Web application..

url in browser address bar::
http://localhost:2525/JspApp1/first.jsp

Jsp container takes care of all these activities

=>For jsp page/file compilation will not be there , only translation to JES (jsp equalent servlet) will be there through page compiler (jspc)..
=> The Generated JES source code will be compiled dynamically and will be executed dynamically by Jsp container..

Q) Why we are not adding jsp-api.jar file to the classpath while developing jsp pages?

Ans) For jsp pages , there is no compilation , only translation to JES class is there.. So there is no need adding any jar file to CLASSPATH/BUILDPATH.

many

Q) How types of objects can be there in jsp pages?

We jsp page , we can have two types of objects

a) Implicit objs
-> Given by JES class automatically as part jsp translation to equalent servlet class
-> Total 9 implicit objs are there.. it is better to call them as implicit ref variable
a) request b) response c) page d)pageContext e)config (internally ServletConfig)
f) application(internall ServletCotnext) g)session h)exception i)out

b) Explicit objs
->Created by the programmers explicty in the scriptlet of jsp page/file
<% java.util.Date d=new java.util.Date(); %>
Explicit obj

What is the difference b/w HTML and JSP?

HTML	JSP
a) It is client side web technology	a) It is server side web technology
b) HTML is given by w3c	b) Jsp is given by Sun Ms (oracle corp)
c)html files Generate static web pages	c) jsp files generate dynamic web pages
d) html tags are not strictly typed tags	d) jsp tags are strictly typed
e) html tags and attributes are not case-sensitive	e) jsp tags and attributes are case-sensitive
f) Does not allow to add java code	f) Allows to add java code..
g) Can be used in any technology/domain based web application development	g) can be used only in java based web applicaiton development.
h) To execute html code we need HTML interpeter	h) To execute Jsp code we need Jsp container + servlet container

=>ServletContainer manages servlet comp life cycle by calling servlet life cycle methods for life cycle events.

1. Instaintiation event :: raises when Servlet container creates our servlet class obj and calls init(ServletConfig cg) as the life cycle method
2. RequestProcessing event :: raises when Servlet container keeps our servlet class obj ready for request processing and calls service(ServletRequest req, ServletResponse res)as the life cycle method
3. Destruction Event :: raises when servlet contianer is about to destroy our servlet class object and calls destroy() method as life cycle method.

=> JspContainer= ServletContainer togather manages jsp page/file/comp life cycle with respect to jsp equlent servlet comp.. and by calling jsp life cycle methods through servlet life cycle methods

=>Instantiation event :: raises when servlet container cretes JES(jsp equalent servlet) class object and calls jspInit() and _jspInit() as the life cycle methods through servlet life cycle method init(ServletConfig cg)

jspInit() :: To place programmer choice initialization logic though .jsp file/page like creating jdbc con object

_jspInit() :: comes in JES class automatically having Container specific initialization logic like activating EL Engine

note:: _xxx methods , _xxx classes are generally Container generated methods , classes I.e programmers will not create them they will be created in JES class automatically.

=> request processing event:: raises when ServletContainer keeps the JES class object ready for request processing by completing initialization. calls _jspService(req,res) method as life cycle method through Servlet life cycle method called service(req,res);

=>The request processing logic kept in jsp page automatically goes to _jspService(-,-) method.

Destruction event :: raises when ServletContainer becomes ready/about to destroy the JES class objs.. calls _jspDestroy() and jspDestroy() methos as jsp life methods through the servlet life cycle method destroy();

jspDestroy() :: To place programmer choice unitialization logic through .jsp file like closing jdbc connection
_jspDestroy() :: contanins container choice unitialization logic dynamically like Deactivating EL Engine..

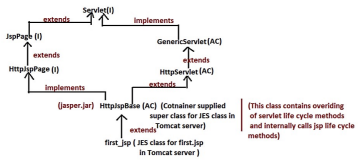
By default every JES class contain^5 _jspInit() , _jspService(-,-) and _jspDestroy() methods and jspDestroy() and jspInit() methods will come only when programmers places them in .jsp file.

=>In tomcat server the first.jsp of JspApp1 web application gets its equalent servlet source code and compiled code as first_jsp.java and first_jsp.class in the following location E:\Tomcat 9.x\work\Catalina\localhost\JspApp1\org\apache\jsp location pkg name

first_jsp.java
first_jsp.class

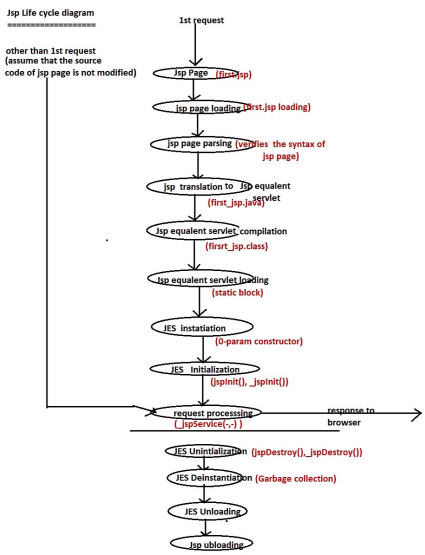
(HttpJspBase in Tomcat server)

=>Every JES class extends from Container spefic class that extends HttpServlet , So we can say every JES class is HttpServlet class.
=>The code placed as scriplet tag code and template text code goes to _jspService(-,-) method of JES class automatically that to as it is.



HttpJspBase.java

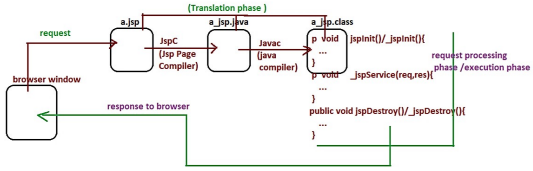
```
public abstract class HttpJspBase
extends HttpServlet
implements HttpJspPage {
  private static final long serialVersionUID = 1L;

  protected HttpJspBase() {
  }

  public final void init(ServletConfig config) throws ServletException {
    super.init(config);
    this.jspInit();
    this._jspInit();
  }

  public String getServletInfo() {
    return Localizer.getMessage("jsp.engine.info", "2.3");
  }

  public final void destroy() {
    this.jspDestroy();
    this._jspDestroy();
  }

  public final void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    this._jspService(request, response);
  }

  public void jspInit() {
  }

  public void _jspInit() {
  }

  public void jspDestroy() {
  }

  protected void _jspDestroy() {
  }

  public abstract void _jspService(HttpServletRequest var1, HttpServletResponse var2) throws ServletException, IOException;
}
```


Jsp Life cycle diagram
1st request
other than 1st request
(assume that the source
code of jsp page is not modified)
Jsp Page (first.jsp)
jsp page loading (first.jsp loading)
jsp page parsing (verifies the syntax of
jsp page)
jsp translation to JSP equalent
servlet
(first_jsp.java)
Jsp equalent servlet compilation
(firsrt_jsp.class)
Jsp equalent servlet loading
(static block)
JES instatiation
(0-param constructor)
JES Initialization
(jspInit(), _jspInit())
request processsing
(_jspService(-,-))
response to
browser
JES Unintialization (jspDestroy(),_jspDestroy())
JES Deinstantiation (Garbage collection)
JES Unloading
Jsp ubloading

Two phases of Jsp execuction

a) Translation phase
=> In this phase jsp page/jsp file will be translated into an equlent servlet comp source code and compiled code.

b) request processing phase/execution phase
=> Here the jsp life cycle methods will be executed by loading , instantiatiating and initializing JES class obj to process the request and to deliver the response to browser.



=>The request given to jsp page directly participates in request processing phase/execution phase if the source code jsp page is not modifiied compare to previous request and byte code (.class) code of JES class not deleted.. otherwise the request given to jsp page participates in both translation phase and request processing phase.

note: For every modification done in jsp page/jsp file soruce code.. the jsp page compiler recognizes that chanage by comparting with prerequest source code and makes the request to participate in both translation and request processing phases.. Due to this there is no need of reloading of web application and restart of server to recognize the chanages..

Various tags/elements in jsp programming

1) *scriptpting tags/elemements*
[These tags are given to place java code/script code in jsp page)
a) scriptlet <% .... %>
b) expresssion <%= ... %>
c) declaration <%! ... %>

2)jsp comments (To commnent jsp code/tags)
a) <%-- .... --%>
jsp tags/code to commnent

3) Directive tags (To give diretions jsp page compiler towards generating code in JES class)

a) page Directtive (<%@page attrbibutes %>)
=>To give global instructions to jsp page compiler like lagnuages, bufffer size,
pkgs to impoort and etc...
b) include directive (<%@include attributes %>)
=>To include the code of given dest web comp to the JSE class code of source jsp page
c) taglib directive (<%@taglib attributes)
=>To use one or more jsp tags of jsp tag librarries in the jsp page.
(it is library where set of jsp tags are avaiable like java package)

4) Action tags
=>These tags internally uses one or another api like servlet api to complete the tasks...
a)standard action tags
<jsp:useBean> ,<jsp:setProperty>, <jsp:getProperty>,<jsp:forward>, <jsp:include>
<jsp:param> ,<jsp:fallback> and etc...

b) Custom action tags
=>created by the programmers manually

c) JSTL tags (Jsp standard tag library tags)
=>tags designing is given by Sun Ms as part of jsp specification and
implemenented by Server vendors like Tomcat, glassFish and etc...
=> nearly 50+ tags are given to avoid or minimize java code in jsp pages..

d) Thirdparty Action tags
=>given by third party VEndors like
JSF ,spring MVC and etc..

Scripting tags -- scriptlet
==========

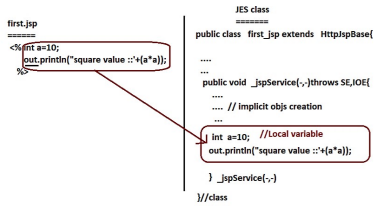
=>scriptlet is given to place java code in jsp page..
=> The code kept in scriptlet tag goes to _jspService(-,-) of JES class
(Jsq equalent servlet class)
=> generally scriptlet tag contains b.logic or request processing logic.

standard syntax::
<% .....
..... //java code
..... %>

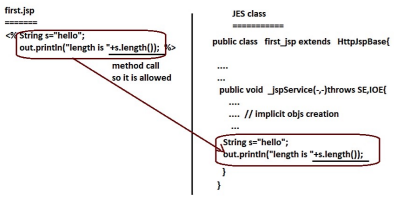
xml syntax:
<jsp:scriplet>
fixed prefix
..... //java code
.....
</jsp:scriplet>

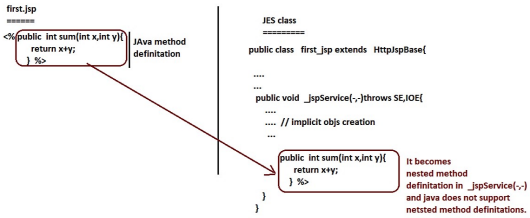
For all jsp supplied built-in tags <jsp:....> xml syntax pattern will be there .. where "jsp" is the fixed prefix.. indicating that tag is supplied as built-in jsp tag.

=>The variables in declared in scriptlet becomes local variables of _jspService(-,-) method.
=> The jsp implicit objects can be used in scriptlet ..becoz all implicit objs are local variables in _jspService(-,-) and the code placed in scriptlet also goes to _jspService(-,-) method.



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

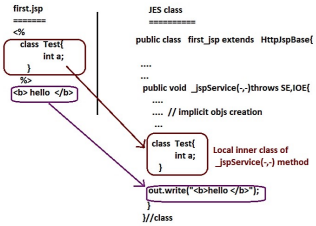




Can i place class definitation inside scriptlet?

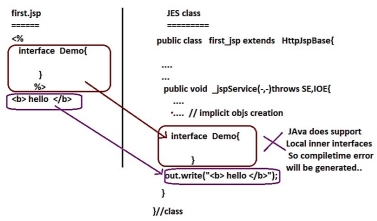
Ans) yes possible becoz it becomes Local inner class of _jspService(-,-) method.. but that class should be taken with out any public ,private and etc.. modifiers.

java supports 4 types of inner classes
a) Normal inner class
b) nested/static inner class
c) Local inner class (The inner class that is placed in definitation)
d) anonymous inner class



Can i place interface definitation inside the scriptlet?

Ans) Not possible.. As of now java does not support method level local inner interfaces.. So compilation error will come if we try to place.



What is the difference b/w out.write(-) and out.print(-) method?

out.write (-)	out.print(-)

(a) can not display null value on to the browser.. rather it thows NullPointerException
<% String s=null;
out.write(s); //throws NPE
%>
	(a) Can display null value on to the browser with out throwing any NullPoinnterException
	<% String s=null;
	out.print(s); //displays "null" on to the browser
	%>

b) GEnerally it is recomanded and also used to write template text (html code+ ordinary text output to browser where there is no chance of having null values) eg:: out.write("hello ");	b) Generally it is used to write the Java code generated outputs to browser.. So that it can even write "null" value to browser if needed.. eg: out.print("date and time "+new java.util.Date());
c) It is inherited method from java.io.Writer(AC) java.io.Writer(AC) ↑ extends javax.servlet.jsp.JspWriter(AC)	c) It is direct method of javax.servlet.jsp.JspWriter(AC) class.. note:: "out" implicit obj type in Jsp is "JspWriter(AC)".

=>can we define class inside another class definitation ? (yes)
=> can we define class inside another interface definitation ?(yes)
=> can we define interface inside another interface definitation ?(yes)
=> can we define interface inside another class definitation ?(yes)

Xml syntax based scritplet tag
==========

first.jsp
======
```
<jsp:scriptlet>
   int a=10;
   int b=20;
   int c=a+b;
   out.println("result ::"+c);
</jsp:scriptlet>
```

JES class
=========
```
public class first_jsp extends   HttpJspBase{
  ...
  ...
  p   v    _jspService(-,-)throws SE,IOE{
    ....
    .... //implicit objs creation
    ...
    int  a=10;
    int b=20;
    int  c=a+b;
    out.println("result ::"+c);
  }
}
```

In one jsp page /file , can place multiple scripting tags either having same xml syntax or same standard syntax or mix of both.

first.jsp
=======
```
<jsp:scriptlet>
   int a=10;
       int  b=20;
 </jsp:scriptlet>
<%
       int c=a+b;
       out.print("result is:"+c);
      %>
```

=>nested scriptlet are not allowed in jsp page.. if we place the jsp page compiler JSPC itself throws the error.

first.jsp
=======
```
<jsp:scriptlet>
   int a=10;
       int  b=20;
          <jsp:scriptlet>
              int d=30;
          </jsp:scriptlet>
</jsp:scriptlet>

<%
       int c=a+b;
       out.print("result is:"+c);
      %>
```

Jsp page compilation error will be generated.

=>While working with xml syntax of scripting tag, we must take care of "<" sysmbol problem

"<" ----- start of tag (jsp/xml/html)
"<" --- java conditional/relational operator

problem:

```
<jsp:scriptlet>
    int  a=20;
         int b=10;
         boolean result= a<b;
         out.println("result is :"+result);
</jsp:scriptlet>
```

Jsp page compiler raises error for "<" symbol of a <b by taking "<" symbol as begining of the sub tag..

Solution1:: (Use standard syntax :: Not a real solution)

first.jsp

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

solution2 :: (take body/content of <jsp:scriptlet> as the xml level CDATA]

=> does not apply any xml meaning and parsing for conversion i.e given content/body will be used as it is in jsp page translation.

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

Here "<" symbol in "a<b" will be taken as normal charater or java text i.e no xml tag meaning will be applied.

note:: since there is no value for ">" symbol with out "<" symbol in xml /html tags , so placing ">" symbol in the xml syntax based scriptlet is not a problem

first.jsp
=======
```
  <jsp:scriptlet>
      int  a=20;
           int b=10;
           boolean result= a>b;
           out.println("result is ::"+result);
   </jsp:scriptlet>
```
true

Expression tag - another scripting tag
==========

=>Any thing that results data generation after evaluation is called an expression.
=>In Java expression can be
  arithmetic operaton (eg: a+b)
  logical operation (eg: a<b)
  instantiation (object creation like eg: new java.util.Date())
  method call that returns result (eg: sum(10,20))
  and etc..

=>Jsp expression tag evaluates the given java expression and also writes the generated result/Data to browser by using out.print(-) method internally.

standard syntax:

```
<%=  ...      %>
```
Expression to evaluate

Xml syntax:

```
<jsp:expression>
   ... //expression to evaluate
</jsp:expression>
```

in

=>The code placed expression tag becomes the arg value of out.print(-) in _jspService(-,-) of JES class.

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

JES class
=========
```
public class first_jsp extends   HttpJspBase{
  ...
  ...
  p   v    _jspService(-,-)throws SE,IOE{
    ....
    .... //implicit objs creation
    ...
    int a=10;
    out.write("value::");
    out.print(a);
    out.write("<br>");
    out.write("square ::");
    out.print(a*a);
  } //method
}//class
```

note:: By using expression tag effectively , we can avoid the utilization of out.print(-) direcly in our jsp page.. and it is good pratice to avoid out.print(-) becoz it helps to seperate java code (b.logic) from html code/ordinary text (presentation logic)

first.jsp
=======
```
<% int a=10;
out.print(" value ::"+a);
out.print("<br> square ::"+a*a);
 %>
```
(Bad coding becoz we mixing up b.logic(java code) and presentation logic(html code))

first.jsp
```
<% int a=10;  %>
 value  :: <%=a  %> <br>
 square :: <%=a*a %>
```
(Good code becoz we are seperating java code (b.logic) from html code (presentation logic)

=>we can use jsp implicit objs in expression tags becoz both are going to come in _jspService(-,-) of JES class..

first.jsp

```
browser name :: <%=request.getHeader("user-agent") %>
<br>
request url  :: <%=request.getRequestURL() %>
<br>
web app name/context path  :: <%=application.getContextPath() %>
```

JES class
=========
```
public class first_jsp extends   HttpJspBase{
  ...
  ...
  p   v    _jspService(-,-)throws SE,IOE{
    ....
    .... //implicit objs creation
    ...
    out.write("browser name :: ");
    out.print(request.getHeader("user-agent") );
    out.write("<br>");
    out.write("request url  :: ");
    out.print(request.getRequestURL() );
    out.write("<br>");
    out.write("web app name/context path  :: ");
    out.print(application.getContextPath() );
    }
}//class
```

Expression tag

=>we can use expression tag to call a method and to display result on to the browser.. but method return type must be other than void

first.jsp

```
<% String s="hello"; %>

value :: <%=s %> <br>
length :: <%=s.length() %>
```

JES class

```
public class first_jsp extends HttpJspBase{

  p v  _jspService(-,-)throws SE,IOE{
     ....
     .... //implicit objs
     ....

    out.write("value::");
    out.print(s);
    out.write("<br>");
    out.write("length::");
    out.print(s.length());

  }
}
```

if u use expression tag to call java method whose return type is void then we get java compilation error

first.jsp

```
<%=System.gc() %>
```

raises error

becoz System.gc() method return type is void

```
An error occurred at line: [6] in the jsp file: [/first.jsp]
The method print(boolean) in the type JspWriter is not applicable for the arguments (void)
3:    value ::  <%=s %> <br>
4:    length ::  <%=s.length() %>
5:
6:    <%=System.gc() %>
```

We can use expression tag for instantiation and to display the default data of the object on to the browser.

first.jsp

```
Date and time :: <%=new java.util.Date() %>
```

JES

```
public class first_jsp extends HttpJspBase{

  p v  _jspService(-,-)throws SE,IOE{
     ....
     .... //implicit objs
     ....

    out.write("Date and time ::");
    out.print( new java.util.Date() );

    .....
    ....
  }
}
```

Xml syntax of expression tag

first.jsp

```
<jsp:scriptlet>
    int a=10;
</jsp:scriptlet>
  value :: <jsp:expression>a </jsp:expression> <br>
  square value :: <jsp:expression>a*a </jsp:expression>
```

=>There is no solution for "<" symbol problem while working with the xml syntax based expression tag.

first.jsp

```
<jsp:scriptlet>
    int a=10;
      int b=20;
</jsp:scriptlet>
  result is :: <jsp:expression>
                  a<b
               </jsp:expression>
```

✗ gives java compilation error.

```
result is :: <jsp:expression>
                <![CDATA[ a<b ]]>
             </jsp:expression>
```

✗ gives java compilation error

=>In One jsp page we can place multiple expression tags either having same xml syntax or standard syntax or mixed syntax but nested expression tags are not allowed..

=> In one jsp page we can use both standard syntax and xml syntax for the same or different scritpting tags.

Declaration tag – Another scripting tag

=>The code placed in declaration tag goes to outside to the _jspService(-,-) method of JES class So we can use declaration tag to declare global variables, to define methods , to define inner classes and etc..

standard syntax:

```
<%!
   ....  //global variables decl,
   ....     method definitations,
   ...      inner class/interface definitation
 %>
```

xml syntax:

```
<jsp:declaration>
   ....  //global variables decl,
   ....     method definitations,
   ...      inner class/interface definitation
</jsp:declaration>
```

=>Variables declared in declaration tag becomes global variables of JES class

first.jsp

```
<%! int a=10; %>
value :: <%=a %> <br>
square :: <%=a*a %> <br>
```

JES

```
public class first_jsp extends HttpJspBase{

  int a=10;

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

note:: we can apply all possible modifiers on the global variables declaration that is happening in declaration tags i.e we can apply private ,public , static ,final and etc.. modifiers..

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

Q) How to declaration tag variable from scriptlet tag variable when both have got same name?

Ans) we differentiate variables either using "this" keyword or using "page" implicit obj..

note:: "this" keyword is recomanded.

first.jsp

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

To work with "page" implicit obj type casting with JES class name is mandatory .. but this JES class name changes server to server or containe to containe (So not recomanded to use)

The implicit object "page" points to "this" in JES class as show below

```
final java.lang.Object page = this;
```

=> if the super class ref variable is refering to sub class object . then to invoke sub class direct methods or direct variales on the reference variable we need to go for type casting.

=>We can declaration tag to to define a java method and we can call that method either using scriptlet tag (if the the return type is void) using expression tag (if the return type is other than void)

=>In one jsp page , we can place zero or more declaration tags..

=> In each declaration tag we can multiple global variables decl, methods declarations, inner classes/ interfaces definitations..

first.jsp

```
<%!
  public int sum(int x,int y){
          return x+y;
  }
  %>
<%!
      public void display(String name){
      System.out.println(name);
      }
%>

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

      <%
          display("raja");
      %>
```

JES class

```
public class first_jsp extends HttpJspBase{

  public int sum(int x,int y){
          return x+y;
  }

  public void display(String name){
   System.out.println(name);
   }

  p v  _jspService(-,-)throws SE,IOE{
     ....
     .... //implicit objs
     ....

    out.write("result::");
    out.print(sum(10,20));
    out.write("<br>");
    display("raja");

  }

}//class
```

Declaration tag

we can use declaration tag to place class,interface definitations becoz they become inner class , inner interface definitation in JES class.

first.jsp

```
<%! class Test{
      ...
    }
    interface Demo{
      ...
    }
  %>
```

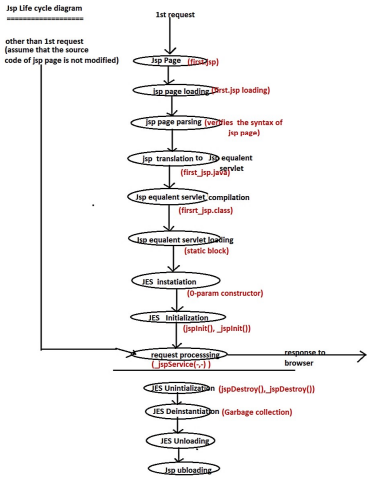
```
public final class first_jsp extends
org.apache.jasper.runtime.HttpJspBase

  class Test{
  }
  interface Demo{
  }
  public void _jspService(-,-)throws SE,IOE{
    ...
    ... //implicit objs

  }
```

becomes inner class of JES class

becomes inner interface of JES class.

=> To place programmer choice initialation logic use jspInit() method and to place programmer choice uninitialation logic use jspDestroy() method .. To place both these method definitations in our jsp page use declaration tag[s].

Jsp Life cycle diagram



---

first.jsp

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

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

        public void jspInit(){
            System.out.println("first_jsp:: jspInit()");
        }
        %>

  <%
      System.out.println(" first_jsp: _jspService(-,-) method");  %>
        <b> hello1 </b>

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

output on the server console for the first request

```
static block
first_jsp:: 0-param constructor
first_jsp:: jspInit()
first_jsp: _jspService(-,-) method
```

for other than 1st request with out modifiying source code of jsp page

```
first_jsp: _jspService(-,-) method
```

for other than 1st request by modlyfing soruce code of jsp page

```
first_jsp:: jspDestroy()
static block
first_jsp:: 0-param constructor
first_jsp:: jspInit()
first_jsp: _jspService(-,-) method
```

=>if modify the source of jsp page
before giving other than 1st request
then
a) JES class object is uninitialized (jspDestroy(), _jspDestroy())
b) JES class obj DeInstantiation
c) JES unloading
d) Jsp page loading
e) the given request will be taken as 1st request and all the operations of 1st request takes place (refer above diagram)

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

Ans) no , becoz all servlet life cycle methods are overriden in the super class of JES class (HttpJspBase in case of Tomcat server) as final methods .. So servlet life cycle methods placed in JES class become illegal (final methods of super class can not be overriden in sub classes)

first.jsp

```
<%!
    public void  init(ServletConfig cg){
      System.out.println(" init(cg) Servlet life cycle method");
    }

      %>
  <b> hello  </b>
```

JES class

```
public final class first_jsp extends org.apache.jasper.runtime.HttpJspBase{

  public void  init(ServletConfig cg){
    System.out.println(" init(cg) Servlet life cycle method");
  }
  ....
  ...
  public void  _jspService(-,-)throws  SE,IOE{
    ...
    ...
  }
}
```

gives java compilation error becoz indirectly we overriding final method of HttpJspBase class in its sub class (JES class)

HttpJspBase.java

```
public abstract class HttpJspBase
extends HttpServlet    implements HttpJspPage {

  public final void init(ServletConfig config) throws ServletException {
    super.init(config);
    this.jspInit();
    this._jspInit();
  }

  public final void destroy() {
    this.jspDestroy();
    this._jspDestroy();
  }
  public final void service(HttpServletRequest request,
                            HttpServletResponse response) throws ServletException, IOException {
    this._jspService(request, response);
  }
  ....
  ....
  ....

}
```

Q) Why servlet life cycle methods are overridden as final methods in HttpJspBase class?

Ans) To make jsp programmers to not to place any servlet life cycle methods in jsp page and to make them to work with jsp life cycle method directly .. this provision restriction is given.

Q) Can place _jspInit() ,_jpService(-,-) and _jspDestroy() method definitations in the declaration tags of jsp page?

Ans) No , becoz they becom duplicate methods in JES class.. as _jspxxx() are alreay there in JES class..
note:: Java supports method overloading but not duplicate methods.

first.jsp

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

      %>

      <b> hello  </b>
```

JES class

```
public final class first_jsp extends org.apache.jasper.runtime.HttpJspBase{

  public void  _jspInit(){
    System.out.println("_jspInit()");
  }
  ....
  ....
  public void _jspInit() {
  }

  public void  _jspDestroy() {
  }

  p  v  _jspService(-,-)throws SE,IOE{
    ....
    ....
    ...

  }
}
```

acts duplicate method for jsp page compiler generated other _jspInit() method ,so java compilation error will come

```
An error occurred at line: [3] in the jsp file: [/first.jsp]
Duplicate method _jspInit() in type first_jsp
1:
2: <%!
3:          public void  _jspInit(){
4:                  System.out.println("_jspInit()");
5:          }
6:
```

Can I jsp implicit objs directly in Declaration tags code?

Ans) No, we not use becoz jsp implicit objs are local variables to _jspService(-,-) and they are not visible in the delcaration tag code that goes to outside of _jspService(-,-) method.

first.jsp

```
<%!  public void display(){
        System.out.println("web name is"+application.getContextPath());
    }
  %>

    <%
        display();
            %>
```

JES class

```
public final class first_jsp extends org.apache.jasper.runtime.HttpJspBase{

  public void display(){
      System.out.println("web name is"+application.getContextPath());
  }
                                  not visible , So compile time error is generated

  p  v  _jspService(-,-)throws SE,IOE{
    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;

      ....
      ....
      ....
  }
}
```

```
An error occurred at line: [3] in the jsp file: [/first.jsp]
application cannot be resolved
1: <%!  public void display(){
2:          System.out.println("web name is"+application.getContextPath());
3:      }
4:    %>
5:
```

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

problem:
=========

first.jsp

```
<jsp:declaration>
   public String findBig(int a,int b){
       if(a<b)
                   return b+" is big";
                      else if(b<a)
                       return a+" is big";
               else
                   return   a+","+b+" are equal";
  }
</jsp:declaration>
 Big Value :: <%=findBig(10,10) %>
```

gives page compilation error

```
org.apache.jasper.JasperException: /first.jsp (line: [2], column: [17]) Unterminated [&lt;jsp:declaration&gt;] tag
```

solution:
=======

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

   Big Value :: <jsp:expression>findBig(10,20) </jsp:expression>
```

=>we can place multiple declaration tags in one jsp page having same xml syntax or same standard syntax or mix of both . but nested declaration tags are not allowed.

=>The first request given to jsp page takes bit extra time as request proessing time compare to other than 1st request becoz the first request has to participate in the following operations before processing the request where other than 1st request directly participates in request processing

->jsp loading
->jsp parsing
->jsp translation to JES soruce code
-> JES compilation
->JES loading
->JES instaitation
->JES intialization

=>To make first directly participating in request processisng ... we need to make container the performing the above listed operations before 1st request either during the deployment of web application or during the server startup.

Code
======

```
  JspApp1
     |------------------>WEB-INF
     |                     |--->web.xml
     |-->first.jsp
```

first.jsp

```
<%! public void jspInit(){
        System.out.println("JES initialization");
    }
            static{
        System.out.println("JES ::static bkock");
    }
            public first_jsp(){
        System.out.println("JES ::0-param constructor");
    }
            %>

<%
  System.out.println("JES :: _jspService(-,-) method");
%>

<b> hello </b>
```

web.xml
=======

```
<web-app>
  <servlet>
    <servlet-name>f</servlet-name>
        <jsp-file>/first.jsp</jsp-file>
        <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
      <servlet-name>f</servlet-name>
  <url-pattern>/furl </url-pattern>
  </servlet-mapping>
</web-app>
```

during deployment of web application
(or)
during server startup
==================================
jsp loading
jsp parsing
jsp translation to JES source code (first_jsp.java)
JES compilation (first_jsp.class)
JES loading (static block)
JES instantiation (0-param consturctor)
JES initialization (jspInit())

For the 1st request (http://localhost:2525/JspApp1/furl)

Request processing (_jspService(-,-))

For other than 1st request (assume no source code is modified)

Request processing (_jspService(-,-))

=>The cfgs done on jsp page in web.xml file like providing url pattern , enabling load-on-startup and etc.. will be applied on jsp page only when that jsp page is requested using url pattern given in web.xml.. (if u request same jsp page using file name that will be take seperate new request and seperate another object will be created for JES class)

note: once the jsp page is cfg in web.xml file having url pattern.. it recomanded to request that jsp page using url pattern..

---

note:: From Tomcat8.x onwords the jsp cfg in web.xml file having url pattern itself enables <load-on-startup> on jsp page .. So there is no need of specyfing <load-on-startup> in jsp cfg seperately,

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

```
<web-app>
 <servlet>
    <servlet-name>f</servlet-name>
        <jsp-file>/first.jsp</jsp-file>
  </servlet>

  <servlet-mapping>
      <servlet-name>f</servlet-name>
     <url-pattern>/furl </url-pattern>
  </servlet-mapping>
</web-app>
```

=>Though <load-on-startup> is not there it bahaves like enabling <load-on-startup> just becoz jsp page/file is cfg in web.xml file having url pattern.

=> if jsp page is there in public area then its cfg in web.xml file is optional.. if jsp page is placed in private area then its cfg in web.xml file in mandatory.

keeping jsp page in private area

=>outside WEB-INF area is called public area and it is visible to every one..

=> WEB-INF and its sub folders area is called private area and its is visible only underlying server/cotainer.. if any outside request i.e browser request wants to go to private area web comp then that web comp must take permission from underlying server/container by cfg its details in web.xml

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>/furl </url-pattern>
 </servlet-mapping>
</web-app>
```

request url

http://localhost:2525/JspApp2/first.jsp (error)
http://localhost:2525/JspApp2/WEB-INF/pages/first.jsp (error)
http://localhost:2525/JspApp2/furl (sucess)

first.jsp

```
<%int a=10; %>
  value :: <%=a %> <br>
  Square :: <%=a *a %>
```

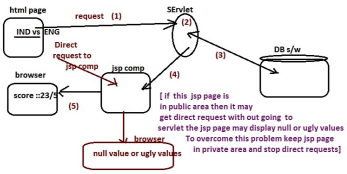
Advantages of keeping jsp page in private area

a) we can hide jsp file name being visible in address bar.. So indirectly we can hide the technology of web app or website from enduser/hackers/jackers.

b) Since web cfg private are jsp comps in web.xml file .. the automatic <load-on-startup> will be enabled.

c) Direct access to jsp page using its file name is not allowed .. so second obj creation for JES class will not created becoz we can not request private jsp page using its filename

d) if jsp page is getting buniess data from servlet comp to display .. then direct to request jsp page will display null or ugly values.. To stop this place jsp page in private area and stop direct requests.

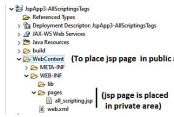


Procedure to develope Jsp based web application in eclipse IDE

step1) make sure that Tomcat is server cfg with eclipse IDE

step2) create Dynamic web Project in eclipse IDE

step3) Add Jsp page in private area web application having all the 3 scripting tags...



(To place jsp page in public area then it can be placed in "webContent") folder.

(jsp page is placed in private area)

all_scripting.jsp

```
<%!
      public   String generate(String user){

               //get System  date and  time
                 java.time.LocalDateTime ldt=java.time.LocalDateTime.now();
               // get current hour of the day
               int hour=ldt.getHour();
               //generate 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;
      }
  %>

<h1 style="color:red;text-align:center">  Welcome  To  JSP  -- Eclipse IDE</h1>
 <b> Date and  time ::</b> <%=new java.util.Date() %>

 <%
      String  name=request.getParameter("uname");
 %>

 <br>
  <b> Wish  Message is :: <%=generate(name) %></b>
```

Declaration tag

Template Text

(Template Text) (Expression tag)

scriptlet

(template text)

(template text) expression tag

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://xmlns.jcp.org/xml/ns/javaee"
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>JspApp3-AllScriptingsTags</display-name>

  <servlet>
    <servlet-name>all</servlet-name>
    <jsp-file>/WEB-INF/pages/all_scripting.jsp</jsp-file>
  </servlet>
  <servlet-mapping>
     <servlet-name>all</servlet-name>
     <url-pattern>/scripting</url-pattern>
  </servlet-mapping>

 </web-app>
```

step4) Run the Web application by deploying in Tomcat server..

Right click on Project ---> run as ---> select Tomcat server ....

request url :: http://localhost:2525/JspApp3-AllScriptingsTags/scripting ?uname=rajesh

note:: Eclipse IDE uses its own copy of Tomcat SErver .. So the JES classes for jsp pages will come in the Tomcat server... the JES class of the above all_scripting.jsp in Eclipse Tomcat server comes in the following location..

G:\Worskpaces\advjava\WTAJ114\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\work\Catalina\localhost\JspApp3-AllScriptingsTags\org\apache\jsp\WEB_002dINF\pages

Comments in Jsp pages
==========

note:: The commnets recognized by the compiler or interpeter will be replaced with white spaces .. Due to this the commented code does not participate in execution.

In jsp page , we can place 3 types of comments

a) Jsp commnets /Hidden Comments

<%-- ... --%>

=>Given to comment jsp tags of jsp page
=>recognized by Jsp page compiler (jspc) during jsp page compilation
(translating jsp into an equalent servlet comp)
=> These comments are visible in any phase of JSP execution i.e not visible in JES class source code , byte code and in the output content that goes to browser . So these comments are called "Hidden comments".

=>There is no xml syntax for jsp component.

b) Html Comments /output comments

<!-- .... -->

=>given to comment the html code of jsp page
=>Recognized by html interpeter of the browser s/w
=>These comments go to browser along with output code that goes to browser and we can see these comments using "view source " option of browser ..So these are called output comments.

c) Java comments / scripting comments

// --> single line comment
/* ... */ --> multi line comment
/** ... */ --> documentation comment

=> Given to comment java code (script code) of jsp page
=> Recognized by java compiler..
=>Since the java code placed in jsp page is called script code.. So these comments are called scripting comments.

=>In eclipse editor , we can use crtl+shift+C to enable or disable all types comments as singleline commnets.. if singleline code is java code then it generated java comment.. if it is jsp code then it geneates jsp comment and etc..

Q) can we commnent template text using jsp comments?
Ans)yes , but not recomanded.

Q) can we commnet jsp tags/jsp code using html comments?
Ans) yes , but not recomanded

all_scriptings.jsp

```
<%!
    public   String generate(String user){

//            int abc=100;

            //get System date and time
              java.time.LocalDateTime ldt=java.time.LocalDateTime.now();
            // get current hour of the day
            int hour=ldt.getHour();
            //generate 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;
      }
 %>

<!--  <h1 style="color:red;text-align:center">  Welcome   To  JSP  -- Eclipse IDE</h1> -->

   <%-- <b> Date and  time ::</b>--%> <!--  <%=new java.util.Date() %> -->

   <%                                  (not a good pratice)
        String  name=request.getParameter("uname");
   %>

    <br>
      <b> Wish  Message is ::
<%--        <%=generate(name) %> --%>

       </b>

   <b>hello </b>
```

=>To enable or disable multiline comments in jsp page according code type
use ctrl+shift+/ --> To enable comment
ctrl+shift+\ --> To disable comment
=>To code is having both template and text and jsp code then these short cut keys give us jsp commnets to comment the whole code.

Comment type	In JES class soruce code	Visibility In JES class byte code	(View Source) In output code goes to browser	browser document
jsp comment <%-- ... --%>	NO	NO	NO	NO
html comment <!-- .... -->	YES	YES	YES	NO
JAva commnent // /* ... */ /** ... */	YES	NO	NO	NO

Scopes in jsp progamming
==========

In Jsp comps , we can keep data in 4 scopes
a) page scope (Specific to current jsp page)
b) request scope (specific to current request)
c) session scope (specific to current browser s/w of a client machine]
d) applicationScope [Conditionless global visibility with in the web application specific to web application]

Test t=new Test();
"t" reference type is "Test" (left hand side)
"t" object type is "Test" (right hand side)
==========
Object obj=new Test();
"obj" reference type is "java.lang.Object" (left hand side)
"object" object type is "Test" (right hand side)

9 implicit objs /reference variables of jsp

implicit object	reference type	scope
out	javax.servlet.jsp.JspWriter (AC)	page
request	javax.servlet.http.HttpServletRequest(I)	request
response	javax.servlet.http.HttpServletResponse(I)	response/ requst
config	javax.servlet.ServletConfig(I)	page
application	javax.servlet.ServletContet(I)	application
session	javax.servlet.ServletHttpSession(I)	session
page	java.lang.Object (C)	page
pageContext	javax.servlet.jsp.tagext.PageContext(AC)	page
exception	javal.lang.Throwable(C)	page

note:: The above jsp implicit objs are not the objects of specified rereferenced types. They are the objects given by Servletcontainer/Jsp container impleementing or extending from the above given interfence type or class type or abstrct class type.

=> In jsp page we need to do exception Handling .. becoz JES class takes care of exception handling..

note:: The above statement is not correct with respect to declaration tags becoz the delcaration tag code goes outside to _JspService(-,-) where we need to handle exception manually..

```
<% Class.forName("oracle.jdbc.driver.OracleDriver"); %>
```

This code goes to _jspService(-,-) of JES class .. So no need of handling exception

JSE class
==========

```
p ublic void _jspService(-,-)throws SE,IOE{
 try{
   Class.forName("oracle.jdbc.driver.OracleDriver");
     out.write(" <b> JDBC driver Loaded </b>\r\n");
   } catch (java.lang.Throwable t) {
       ....
   }
 }
```

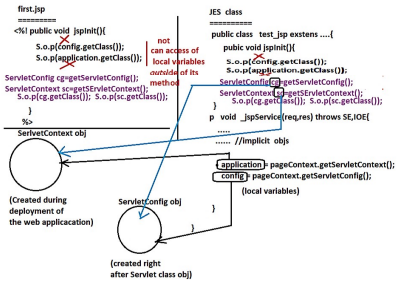
```
<%! Class.forName("oracle.jdbc.driver.OracleDriver"); %>
```

Thos code goes to JES class that to outside of _jspService(-,-) ..So we need handle exception manually as show below

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

=>In Declaration tag code, Jsp implicit objs /ref variables can not be used becoz they are local variables of _jspService(-,-) method and the declaration tag codes outside to _jspService(-,-) method.

=>ServletConfig object, ServletContext obj (application) are actually visible in all 3 life cycle methods of servlet comp becoz ServletConfig obj is created right after ServletClass obj creation and ServletContext object is created during the deployment of the web application..



=> orginally ServletContext ,ServletConfig objs are visible and accessible in all life cycle methods of servlet comp and jsp comp.. But In Jsp page we can not access those using "application","config" implicit ref variables in declaration code that goes outside to _jspService(-,-) method becoz "appliation","config" local variables of _jspService(-,-) method. To overcome this problem create direct reference variables to Servletcontext obj,ServletConfig objs as shown above.

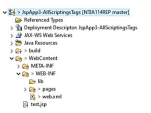
=> If webapplication is cfg having context params in web.xml file then they will be saved automatically in SevletContext obj and servlet /jsp comps can access them through ServletContext obj using sc.getInitParameter(-) methods get

=> If multiple servlet/jsp comps wants to technical data like JDBC properties from outside of servlet,jsp comps then use context parametes/global init parameters throuhg the support of ServletContext obj

=> If Servlet/jsp is cfg having init params in web.xml file then they will be saved automatically in SevletConfig obj of servlet/jsp comp and servlet /jsp comp can access them through ServletConfig obj using cg.getInitParameter(-) methods

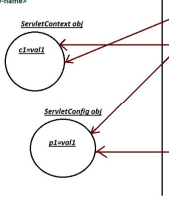
=> If specific servlet/jsp comp wants to get technical data like JDBC properties from outside of that servlet/jsp comp then use Init parametes throuhg the support of servlet.config obj

Example
=======



web.xml
=======

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
 http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
<display-name>JspApp3-AllScriptingsTags</display-name>

 <context-param>
   <param-name>c1</param-name>
   <param-value>val1</param-value>
 </context-param>

 <servlet>
  <servlet-name>t</servlet-name>
  <jsp-file>/test.jsp</jsp-file>
  <init-param>
    <param-name>p1</param-name>
    <param-value>val2</param-value>
  </init-param>

 </servlet>
 <servlet-mapping>
   <servlet-name>t</servlet-name>
   <url-pattern>/testurl</url-pattern>
 </servlet-mapping>

</web-app>
```



test.jsp

```
<%!
  public void jspInit(){
       //get access to ServletConfig,ServletContext objs
       ServletConfig cg=getServletConfig();
       ServletContext sc=getServletContext();
       //read context param values
       System.out.println("c1 context param value ::"+sc.getInitParameter("c1"));
      //read init param values
       System.out.println("p1 init param value ::"+cg.getInitParameter("p1"));
  }
  %>

<b>c1 context param value :: <%= application.getInitParameter("c1")%> </b> <br>
<b>p1 init param value :: <%= config.getInitParameter("p1")%> </b> <br>
```

request url :: http://localhost:2525/JspApp3-AllScriptingsTags/testurl

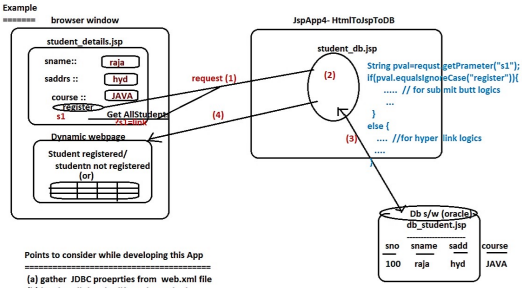
note:: The above process can not be applied to access other immplicit objs of jsp page becoz they are orginally ready only after instantiation and initialization of JES class object.

Html to jsp to Db s/w communication
==========

For html to jsp communication we can pass jsp file or url pattern either as action url in <form> tag or as href ulr in <a> tag.

=>For jsp to Db s/w communication place JDBC code in jsp comp

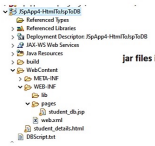
Example
=======



Points to consider while developing this App
==========

(a) gather JDBC proeprties from web.xml file
(b) involve all the jsp life cycle methods
(c) avoid out.println(-) methos by using expression tag effectively
(d) use all the 3 scripting tags
(e) differentiate logics from hyperlink and submit buttons
(f) keep jsp page in private area
and etc..

Refer this this Example App



jar files in BUIDPATH and in WEB-INF/lib folder
ojdbc6.jar

```
### sequence
CREATE SEQUENCE  "SYSTEM"."SNO_SEQ"  MINVALUE 1 MAXVALUE 1000 INCREMENT BY 1 START WITH 4 CACHE 20 NOORDER
NOCYCLE ;

### DB table
CREATE TABLE "SYSTEM"."JSP_STUDENT"
  (    "SNO" NUMBER(10,0) NOT NULL ENABLE,
       "SNAME" VARCHAR2(20 BYTE),
       "SADD" VARCHAR2(20 BYTE),
       "COURSE" VARCHAR2(20 BYTE),
       CONSTRAINT "JSP_STUDENT_PK" PRIMARY KEY ("SNO"));
  }
```

Directive tags in jsp page

=>These tags gives directions to jsp page compiler to generate Java code in JES class.

syn::

```
<%@ <tag name>  attributes  %>
```

3 Directive tags

a) page directive <%@page attributes %>

=>To provide global/common info to JES class from jsp page like buffer size, autoFlush mode, pkg imports , content type and etc..

b) Include directive <%@include attributes %>

=> To include given file content to JES class code of current jsp page

c) taglib directive <%@taglib attributes %>

=> To import and use Custom Jsp/ Third party jsp tag library tags in our jsp page

a)page Directive

=>To provide global/common info to JES class from jsp page like buffer size, autoFlush mode, pkg imports , content type and etc..

standard syn::

```
<%@page attributes %>
```

xml syntax::

```
<jsp:directive.page attributes />
```

=>The code of this jsp tag goes to different parts of JES class based on the attributes that we are using.

attributes

- info
- language
- import
- session
- contentType
- pageEncoding
- extends
- buffer
- autoFlush
- isELIgnored
- isThreadSafe

info

=>It is given to provide short description jsp page

=> No default value

first.jsp

```
<%@page info="this is report page"%>
```

=>JES class internally uses getServletInfo() method to return this given short description

JES class

```
public class first_jsp extends ...{

   public String getServletInfo(){
        return " this is report page";
    }

    ....
    ....
}
```

=>we can place this attribute for multiple times either in same page directive or different page directives of a jsp page having same value.. but placing with different values throw exception.

first.jsp

```
<%@ page info="test page" info="test page" %> //valid
```

first.jsp

```
<%@ page info="test page" info="test page1" %> //invalid
```

first.jsp

```
<%@ page info="test page" %>
<%@ page info="test page" %> //valid
```

first.jsp

```
<%@ page info="test page112" %>
<%@ page info="test page113" %> //invalid
```

language

=>Allows to specify programming language using which we can place script code in jsp page .. As of now we can place only "java" as the language. "java" is the default language and only one language that is supported as of now.

```
<%@ page language="java" %> //valid
<%@page language="c" %> //invalid
<%@ page language="c++" %> //invalid
```

=>This attribute does not place anything in JES class .. but makes jsp page compiler to check wheather correct /valid language is placed or not

=> we can repeat this attribute either in same or differnt page directive having same values (same as info)

import

=>gives to import java pakages to JES class..

=> No default value

=> Multiple packages can be imported either using single import attribute taking "," seperator b/w values or we use import attribute for multiple times either in same <%page%> tag or in different <%page %> tags.

first.jsp

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

first.jsp

```
<%@ page import="java.sql.*" import="java.util.*" %>  //valid
```

first.jsp

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

//valid

=>this attribute can be repeated either in one page directive tag or in mulitple page directive tags either having same values or different values.

contentType

=>Allows to specify response content type

=> internally uses res.setContentType(-) in _jspService(-,-) of JES class to give instruction to browser on response content type

=>default value :: text/html; charset=ISO-8859-1

allows to specify ISO or UTF charsets..

first.jsp

```
<%@page contentType="text/plain"%>
<b> welcome to jsp</b>
```

if we place content type explictly by calling response.cotentType(-) then does it overrides "contentType" attribute value of <@page >tag or not?

Ans) yes overrides becoz explictly placed response.cotentType(-) method always come after "contentType" attribute reated response.setContentType(-) in JES class.

first.jsp

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

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

<b> welcome to jsp</b>
```

in _jspService(-,-) of JES class for first.jsp

```
try {
    response.setContentType("text/plain"); -->related to "contentType" attribute
    pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
    _jspx_page_context = pageContext;
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut();
    _jspx_out = out;

   response.setContentType("text/html");  ---> related to explicitly called response.setContentType(-)
   ...
   ....
   }
```

To place an render different speaking languages content in jsp page and on to browser we need to response content type along with charset=UTF-8 as shown below

first.jsp

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

<b> नमस्ते</b> <br>
<b>ನಮಸ್ಕೆ </b> <br>
<b>ఓ</b>
```

Get these things from Google translate.

Same code can be written as

first.jsp

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

   <b> नमस्ते</b> <br>
   <b>ನಮಸ್ಕೆ </b> <br>
   <b>ఓ</b> <br>
   <b>ઓ</b>
```

=>"pageEncoding" is to sepcify "encoding" charsets like ISO,UTF and etc..

=>default value of pageEncoding is "ISO-8859-1".

=>These attributes can not repeated with different values either in same or in different page directives

session

=>gives to specify wheather implicit obj "session" should be created or not ..

=> we need session object only when session tracking is enabled otherwise it is not required but implicit objs it will be created every time .. To disable that go for this "session" attribute.

=> default value is "true"

```
<%@page session="false" %>
      Does not create implicit obj "session"
<%@page session="true" %>
      creates the implicit obj "session"
```

=>This attribute can not repeated with different values

Page directive tag and its attributes

extends

=> allows to specify our choice class name as the super class of JES class.
=> This attribute is not recomanded to use becoz the super JES class should have multiple things in order Jsp and JES class execution smoothtly

(a) class should extends HttpServlet
(b) should override servlet life cycle methods calling jsp life cycle methods
(c) should place the following methods as null method definitations
jspInit() , _jspInit() , _jspService(-,-) , _jspDestroy(), jspDestroy() methods..

=> When do not use this attribute, the JES class gets a container supplied java classs as super class performing all the above operations..

first.jsp

eg: `<%@page extends="com.nt.base.MyTestBase" %>`

`<b> hello </b>`

JES class

```
public class first_jsp extends MyTestBase{
  ....
  ...
}
```

=>This atribute can not repeated with different values

isELIgnored

=>EL means Expression Language .. it can be used jsp page to avoid or minimize java code jsp page
=> EL not only evaluates the given expression and also displays the generated output on to the browser like expression tag.
=> EL allows to perform airthmetic and logical operations with out using java code.
syn :: ${<expression>}
eg: ${4+5} -->gives 9

=>JES internally use EL Engine to recognize and execute EL code .. To give instruction to JES class to recongnize or ignore EL code of jsp page we can use "isELIgnored" attribute.

first.jsp
<%@page isELIgnored="true" %>
${4+5} ---> gives ${4+5}

ELEnginge is not activated in JES

first.jsp
<%@page isELIgnored="false" %>
${4+5} ---> gives 9

ELEnginge is activated in JES class

Default value of isELIgnored is "false".

buffer & auotFlush

buffer/cache is a tempoary memory that holds data for temporary period.
to out
Servlet comp writes the output messages directly response object with having buffer support

Servlet comp

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

response obj

Jsp comp writes the output messages to response object having buffer support

jsp comp /JES class

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

buffer/cache

response obj

=>The processing writing buffer/cache content to its destination is called flushing
=>The buffer/cache content of jsp page will be flushed only when buffer/cache is filled up or jsp/JES class code execution is completed.

=>The default size of jsp buffer is 8kb ..and it can be controlled using "buffer" attribute of <%@page %>
=>The default value "autoFlush" attribute is "true" i.e once the buffer is filled up or jsp /JES class code execution is completed the buffer content will be flushed out to ressponse obj

first.jsp

```
<%@page buffer="5kb" autoFlush="true"%>

<b>hello </b>
```

all these detials relfects in JES class while creating pageContext obj becoz pageContext should hold multiple details about courrent jsp page..

```
pageContext = _jspxFactory.getPageContext(this, request, response,
                "error.jsp", false, 5120, false);
```

error page, session mode, buffer size, autoflush mode

=>This atribute can not repeated with different values

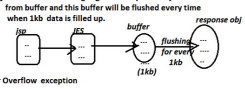
What happends if the jsp page tries to write more that buffen size data as output?

Ans) if autoFlush is enabled (autoFlush="true") then no problem.. jsp generated output will be written to response object through buffer for multiple times

first.jsp

```
<%@page buffer="1kb" autoFlush="true" %>
 <%
   for(int i=1;i<=50000;++i){
       out.write("raja--->"+i);
   }
 %>
```

[each time 1kb messages goes to response obj from buffer and this buffer will be flushed every time when 1kb data is filled up.

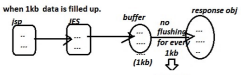


if autoFlush is disabled .. we will get IOExcepetion:: Jsp buffer Overflow exception

first.jsp

```
<%@page buffer="1kb" autoFlush="false" %>

 <%
   for(int i=1;i<=50000;++i){
       out.write("raja--->"+i);
   }
 %>
```

when 1kb data is filled up.



Root Cause
`java.io.IOException: Error: JSP Buffer overflow`

When autoFlush is disabled ..we can manually flush the buffer

first.jsp

```
<%@page buffer="1kb" autoFlush="false" %>

 <%
   for(int i=1;i<=50000;++i){
       if(i%50==0)
           out.flush(); //manuall flushing
       out.write("raja--->"+i);
   }
 %>
```

=>Default value for "buffer" is 8kb
=>Default value for "autoFlush" is true
=>These attrbutes can not repeated with different values

=>we can disable buffer by taking buffer="none"

first.jsp

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

Since no buffer the jsp comp can write the output to response obj directly

```
 <%
   for(int i=1;i<=50000;++i){
       out.write("raja--->"+i);
   }
 %>
```

=>When buffer is "none" the enabling or disabling of autoFlush mode is meaningless ...but error will come only autoFlush="false" .. not for autoFlush="true" becoz "true" is the default value.

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

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

What is the difference PrintWriter and JspWriter?

PrintWriter	*JspWriter*
(a) given by IO streams java.io pkg	*(a) given by jsp api in javax.servlet.jsp pkg*
(b) Useful in servlet programming to write outputs directly to response obj	*(b) useful in jsp programming to write the outputs to response object through Buffer (it is "out" object type)*
(c) Does not support buffering	*(c) Supports buffering*
(d) Will be used directly	*(d) JspWriter internally uses PrintWriter if the jsp page is taken with out buffering i.e buffer="none" in <%@page %>*
(e) It is concreate class extends from java.io.Writer (C) class	*(e) It is an abstract class extending from java.io.Writer (C)*
(f) pirnt(-) methods of this does not throw IOException	*(f) pirnt(-) methods of this class throw IOException*

Buffering
The initial JspWriter object is associated with the PrintWriter object of the ServletResponse in a way that depends on whether the page is or is not buffered. If the page is not buffered, output written to this JspWriter object will be written through to the PrintWriter directly, which will be created if necessary by invoking the getWriter() method on the response object. But if the page is buffered, the PrintWriter object will not be created until the buffer is flushed and operations like setContentType() are legal. Since this flexibility simplifies programming substantially, buffering is the default for JSP pages.

What is the difference b/w page and PageContext objs?

What is the difference b/w page and PageContext?

Ans) Both are implict objects where page refers to "this" nothing bit JES class ref and it can be used to differentiate global varible from local variable when both have same name ..

```
// JES class
-----------
final java.lang.Object page = this;
```

=>pageContext obj holds multiple details current jsp page like req,res , this, error page name, session mode, bufffer size, autoFlush mode and etc..

=> All implicit can be accessed through pageContext object..

```
JES class
application = pageContext.getServletContext();
config = pageContext.getServletConfig();
session = pageContext.getSession();
out = pageContext.getOut();
....
.....
```

=> since pageContext holds multiple implict object jsp .. using one pageContext object we can create multiple scopes of attributes like "pageScope" , "requestScope", "sessionScope" and " applicaitonScope"

[In servlet programmign we use req,session, ServletContext obj seperately to create request scope, session scope, application scope attributes respectively.. where as in jsp programming we can creates all 4 scopes are attibutes (discussed above using same pageContext obj).

```
In JES class
==========
pageContext = _jspxFactory.getPageContext(this, request, response,
                "error.jsp", true, 8192, true);
error page ⇩  session mode   ⇩ buffer size   ⇩ autoFlush mode.
```

---

isThreadSafe

=> Specfies wheather JES class object /code should become thread safe or not by impmenting javax.servlet.SingleThreadModel()).

```
<%@isThreadSafe="false"%>
    =>Makes JES class as thread safe by implementing javax.servlet.SingleThreadModel()
<%@isThreadSafe="true"%>
    =>Makes the JES class not to implement javax.servlet.SingleThreadModel()).. So we should place
      synchronized blocks explicitly in scriptlets to achieve thread safety..
```

first.jsp

```
<%  synchronized(session){
        ....
        .... //session realated logics
        ....
    }
    synchronized(application){
        ....
        .... //application realated logics
        ....
    }
%>
```

=>default value "true" and this attribute can be repeated with different values.

---

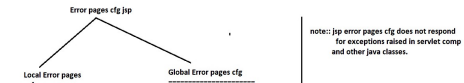
import points on page Directive tag

a) page directive and its attributes are case-sensitive
b) invalid attributes are not allowed
c) except "import" no other attribute can have multiple values seperated with "," symbol
c) except "import" no other attribute can be repeated having different values either sname <%@page %> tag or in multiple <%@page %> tags..

Error Page cfg in Jsp

=>The page that executes only when the exception is raised in other pages of jsp is called error page.
=>we can take either html page or jsp page as error page .. but jsp page is recomanded becoz the implicit obj "exception" is visible the jsp page that is acting as error page..
=> Using "exception" obj we can get the exception related details in the error page..
=> The implicit obj "exception" will be not created for normal jsp pages.. It will be created only in error pages.

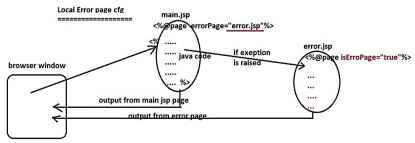
=>Error cfgs does not mean exception handling becoz exception handling taken care by _jspService(..) of JES class automatically.. Error pages cfg is all about executing certain html/jsp page only when the exception is raised in other pages..



=>use errorPage,isErrorPage attributes of <%@page %>

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

note:: jsp error pages cfg does not respond for exceptions raised in servlet comp and other java classes.



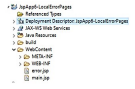
note:: error pages are useful to display exception related message non-technical guding messages to enduser like instead of "NullPointerException","ClassNotFoundException" and etc.. on the browser as techinal ugly messages.. it is recomanded to display " internal problem-Try again" kind of non-technical guiding messages to enduser

<%@page %> attribues are

errorPage :: To sepcify the error page name(jsp or html) , no default value

isErrorPage ::(true/false) default vlaue is "false" specifies wheahter current jsp page should act as error page or not

Example App



main.jsp

```
<%@page errorPage="error.jsp" %>

<b> from main.jsp </b>
value =<%=Integer.parseInt("a10") %>
```

error.jsp

```
<%@ page isErrorPage="true" %>
<b> <i> Internal Problem try again</i></b> <br>
<a href="main.jsp">home</a>
<br>
<%=exception.getMessage() %>
```

Example on Global error pages

main.jsp

```
<b> from main.jsp </b>
value =<%=Integer.parseInt("a10") %>
```

main1.jsp

```
<%@page import="java.util.*" %>
<b> from main1.jsp </b> <br>

 <%  Date d=null;
     d=new Date();%>
 current year :: <%=d.getYear() %>
```

error.jsp

```
<%@ page isErrorPage="true" %>
<b> <i> Internal Problem try again</i></b> <br>
<a href="main.jsp">home</a>
<br>
<%=exception.getMessage() %>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>JspApp6-LocalErrorPages</display-name>
  <error-page>
     <exception-type>java.lang.Exception</exception-type>
     <location>/error.jsp</location>
  </error-page>
</web-app>
```

For any exception raised in any jsp page the error.jsp executes as the error page

if needed, we can cfg different errror pages for diffrent exceptions

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="http://xmlns.jcp.org/xml/ns/javaee"
 xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
 id="WebApp_ID" version="4.0">

 <display-name>JspApp6-LocalErrorPages</display-name>
 <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>
</web-app>
```

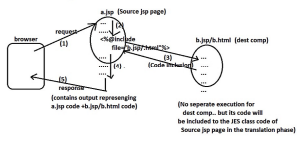
What happens if cfg two diffent error pages (1 for Local error page cfg , 2 for global error page cfg) for the exception ?

Ans) when exception is raised the local error page executes..

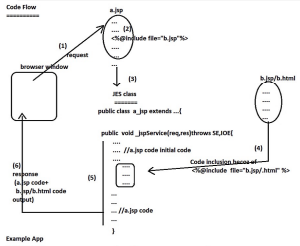
Diretive include /static include/ Translationphase include

=>This tag takes given dest comp code and includes in JES class code of source jsp page
=> This tag does not use rd.include(..,) internally
=> This code inclusion takes place during translation phase of source jsp page . So it is called Translation phase include or static include.
standard syntax:: <%@include file="<dest comp"%>
xml syntax:: <jsp:directive.include file="<dest comp"%>

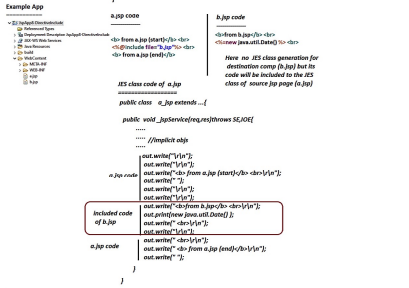
note: This tag allows to take jsp/html comp as the destination comp.. but not the servlet comp.



Code Flow



Example App



---

taglib directive / <%@taglib attribues %>
Will be discussed while working JSTL tags and Custom Tags development.

Jsp Action tags
uses and
=> These tags internally one or other technology api completes the task..
are
=>These action tags that having standard syntax .. only xml syntax is available

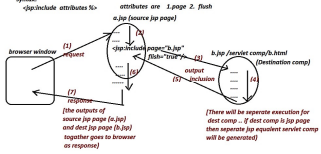
Different types of jsp action tags
=>standard action tags (built-in with jsp technology)
=> JSTL tags (specification is given Sun Ms and functionalities for tags is given by server vendors like tomcat ,weblogic and etc...)
=>Third party tags (given by struts, spring MVC and etc... frameworks)
=>Custom tags (developed by programmers)

The standard action tags are
<jsp:include>, <jsp:forward>, <jsp:useBean>, <jsp:setProperty>, <jsp:getProperty>, <jsp:param>
<jsp:plugin>, <jsp:fallback> and etc..

Action include / dynamic include /Execution phase include /<jsp:include>

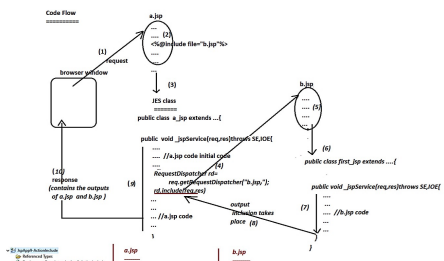
=>This tag internally uses rd.include(..,) and includes the dest comp output to source jsp page comp output (Here code inclusion will not happen.. output inclusion takes place)
=>Allows to take servlet comp or jsp comp or html file as the destination comp.
=>Destination comp will be executed seperately and its output will be included to the source jsp output dynamically at runtime ...so it is also called as runtime binding.

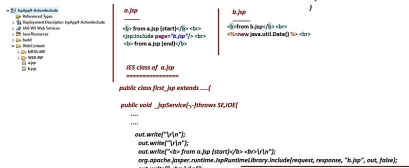
syntax:
<jsp:include attribues %> attributes are 1.page 2. flush



attriubes ::
page :: dest comp info like
file name /url pattern for jsp/html comps
url pattern for servlet comp
flush :: false(default)/true (specifies wheather the jsp buffer of source jsp should flush or not its already available output of jsp buffer to response object before including the dest comp output to buffer)

Code based flow



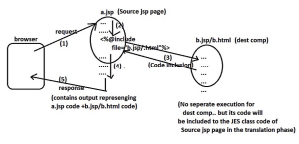


(Internally uses rd.include(..,))

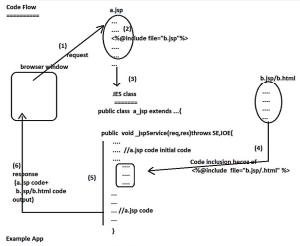
Diretive include /static include/ Translationphase include

=>This tag takes given dest comp code and includes in JES class code of source jsp page
=> This tag does not use rd.include(..,) internally
=> This code inclusion takes place during translation phase of source jsp page . So it is called Translation phase include or static include.
standard syntax:: <%@include file="<dest comp"%>
xml syntax:: <jsp:directive.include file="<dest comp"%>

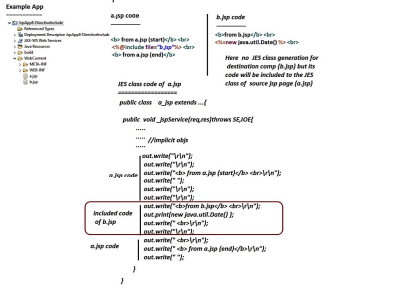
note: This tag allows to take jsp/html comp as the destination comp.. but not the servlet comp.



Code Flow



Example App



---

taglib directive / <%@taglib attribues %>
Will be discussed while working JSTL tags and Custom Tags development.

Jsp Action tags
uses and
=> These tags internally one or other technology api completes the task..
are
=>These action tags that having standard syntax .. only xml syntax is available

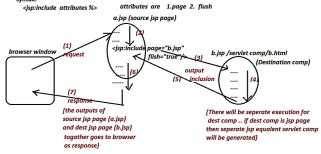
Different types of jsp action tags
=>standard action tags (built-in with jsp technology)
=> JSTL tags (specification is given Sun Ms and functionalities for tags is given by server vendors like tomcat ,weblogic and etc...)
=>Third party tags (given by struts, spring MVC and etc... frameworks)
=>Custom tags (developed by programmers)

The standard action tags are
<jsp:include>, <jsp:forward>, <jsp:useBean>, <jsp:setProperty>, <jsp:getProperty>, <jsp:param>
<jsp:plugin>, <jsp:fallback> and etc..

Action include / dynamic include /Execution phase include /<jsp:include>

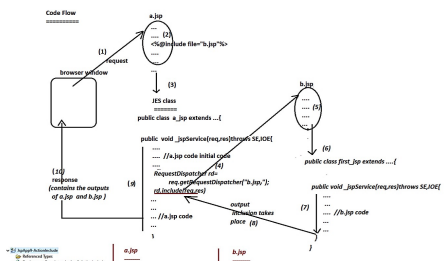
=>This tag internally uses rd.include(..,) and includes the dest comp output to source jsp page comp output (Here code inclusion will not happen.. output inclusion takes place)
=>Allows to take servlet comp or jsp comp or html file as the destination comp.
=>Destination comp will be executed seperately and its output will be included to the source jsp output dynamically at runtime ...so it is also called as runtime binding.

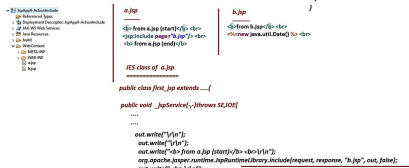
syntax:
<jsp:include attribues %> attributes are 1.page 2. flush



attriubes ::
page :: dest comp info like
file name /url pattern for jsp/html comps
url pattern for servlet comp
flush :: false(default)/true (specifies wheather the jsp buffer of source jsp should flush or not its already available output of jsp buffer to response object before including the dest comp output to buffer)

Code based flow



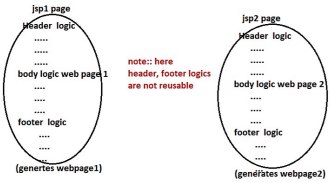


(Internally uses rd.include(..,))

What is the difference between Directive include and Action Include?

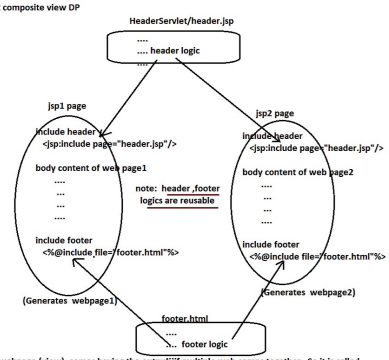
Directive include	Action include
1) Perform code inclusion at translation phase So it is called static inclusion or static binding or translation phase inclusion or compiletime binding	1) Performs output inclusion at runtime or at execution pahse So it is called dynamic inclusion or runtime inclusion or execution phase inclusion or runtime binding.
2) This tag having boht standard syntax and xml syntx	2) This tag having only xml syntax
3) Does not rd.include(-,-) internally	3) uses rd.unclude(-,-) internally
4) Does not allow servlet comp as the dest comp all only jsp,html comps as the destination comp	4) Allows servlet,jsp, html comps as the destination comps
5) No seperate execution for dest comp	5) executes destination comp
(6) if the destination comp jsp page.. then JES class will be geneated for that jsp page	(6) No JES class will be generated
(7) useful if the dest comp is static web comp like html file	(7) useful if the dest comp is dynamic web comp like jsp,servlet comp

Problem::



Every webpage basically contains 3parts minimum header section , body section and footer section.

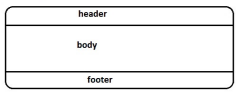
Solution: implement composite view DP



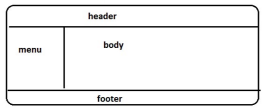
HEre every webpage (view) comes having the output of multiple web comps togather , So it is called as Composite view .

Popular layouts of web application development

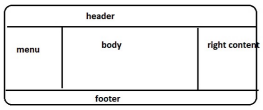
1.Classic Layout



Two colum layout

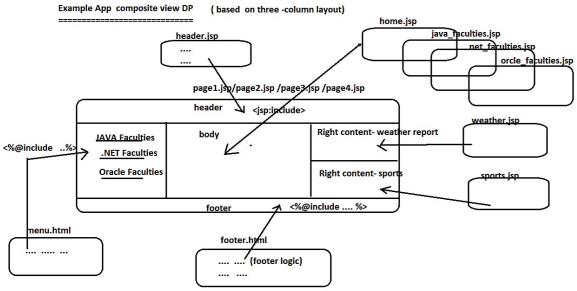


Three colum layout

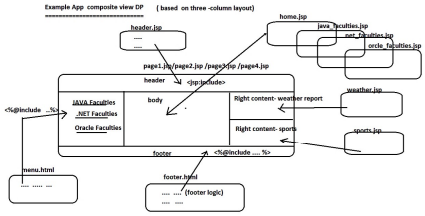


note:: we can place both action include and directive include in one jsp page for multiple times then both code inclusion and output inclusion takes place.

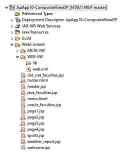
Example App composite view DP (based on three -column layout)



Example App composite view DP (based on three -column layout)



refr sample App on composite view

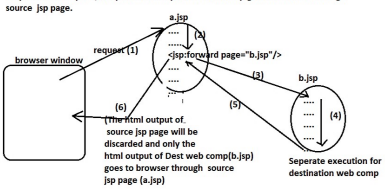


Action Forward /<jsp:forward>

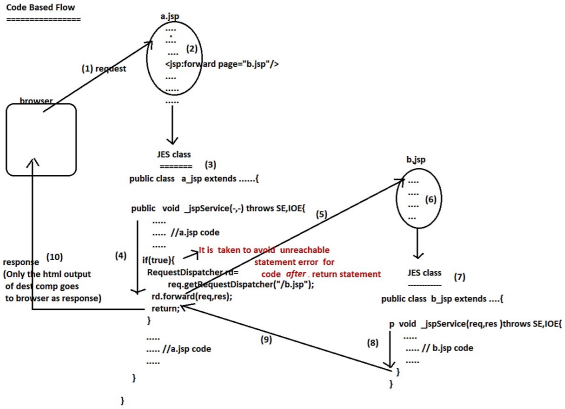
=> This tag internally uses rd.forward(-,-) and performs forwarding requset mode communication with Dest comp (another jsp page or servlet comp or html file)

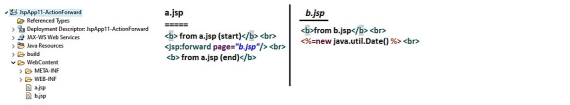
=> We have only Action forward tag (<jsp:forward>tag) and we do not have directive forward tag where as we are having both Directive include and Action include tags

=> Html outputs /template text outputs generated by the source jsp comp will be discarded only the html outputs /template text outputs of dest comp goes to browser through source jsp page.



Code Based Flow





Jsp Communication

=> Making the request given to the source jsp page to Dest web comp and it can be done in 3 ways

a) forwarding request (using <jsp:foward> tag)
b) including response (using <jsp:include> tag)
c) send Redirection using res.sendRedirect(-)
(No tag for this ..so we need to write *java* code directly)

(Here Source jsp page and dest web comp use the same req, res objs and more over the source jsp page and dest web comp must be there in the same web application or in two different web application of same server)

Here source jsp and dest web comps do not use same req,res objs and source jsp and dest web comps can be there in same web application or in two different web applications of same server or diffent servers belonging to same machine or different machines.

=>if source jsp page and dest web comps are using same req ,res objs with the support of <jsp:forward> ,<jsp:include>tags the source page can pass addtional data to dest web comp as addtional request params using <jsp:param> tags (should always be used as the sub tags of <jsp:forward> ,<jsp:include> tags)

Example code

a.jsp

```
<b> from a.jsp (start)</b> <br>
<%
  float xeroxPrice= 300 * 1.25f;
%>
<jsp:forward page="b.jsp">
  <jsp:param value="raja" name="name"/>
  <jsp:param value="30" name="age"/>
  <jsp:param value="<%=xeroxPrice%>" name="amt"/>
</jsp:forward>
<br>
<b> from a.jsp (end)</b>
```

creates the addtional req params with given names and values

b.jsp

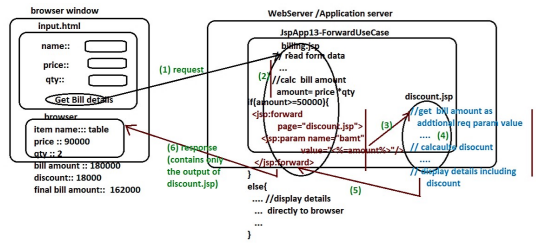
```
<b>from b.jsp</b> <br>
<%=new java.util.Date() %> <br>
<b> additonal req param values</b>
  name req param value :: <%=request.getParameter("name") %> <br>
  age req param value :: <%=Integer.parseInt(request.getParameter("age")) %> <br>
  amount for xerox :: <%=Float.parseFloat(request.getParameter("amt")) %> <br>
```

Q) Why there is not Directive Forward tag?

Ans) Directive tags perform their activity at translation phase by including the code of given dest web comp.. In this behaviour discarding output of source page is not possible.. So there is not Directive forward tag.

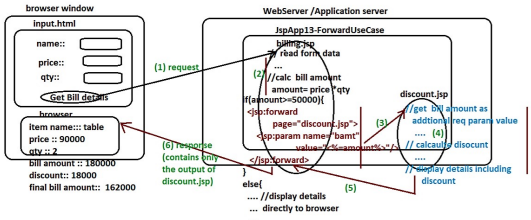
Use case /example App on <jsp:foward> and <jsp:param> tags

=>It is always recomanded to keep <jsp:forward> tag in source jsp page as a conditional statement so the html output of source jsp page will be discarded only when that condition is satisfied.



Use case /example App on <jsp:foward> and <jsp:param> tags

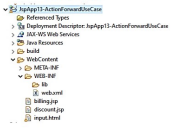
=>it is always recomanded to keep <jsp:forward> tag in source jsp page as a conditional statement so the html output of source jsp page will be discarded only when that condition is satisfied.



=>we can use <jsp:param> tag only as the sub tag of <jsp:forward> or <jsp:include>tags.

=>It is always recomanded to place <jsp:forward> tag in source jsp page as conditional statement.

Example App::



What is the diffrence b/w <jsp:include> and <jsp:forward>?

<jsp:include>	<jsp:forward>
(a) performs including response of communication	(a) performs forwarding request mode of communication
(b) All the statements in source jsp page that are placed before and after <jsp:include> tag executes	(b) the statements placed in source jsp page that are after <jsp:forward> tag does not executes
(c) No hmtl /template text ouput will be discarded.. More over the html/template text outputs of both soruce and dest pages togather goes to browser as response	(c) The html/templat text of source jsp page will be discarded and only the html/template text of output dest web comp goes to browser as response through source jsp page
(d) Internally uses rd.include(-,-) method	(d) Internally uses rd.forward(-,-) method.
(e) Useful in composite view DP implementation	(e) Useful in forwarding request from source jsp page to Dest web comp on condition basis.

What happens if we place multiple <jsp:forward> tags in source jsp page?

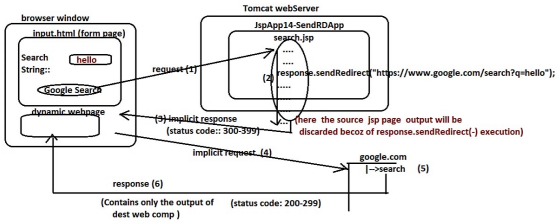
Ans) only first <jsp:forward> executes and remaining will not be executed becoz "return" statement that comes in after first rd.forward(-,-) in the JES class source jsp page.

What happens if we place multiple <jsp:include > tags in source jsp page?

Ans) The html/template text outputs given by mulitple destination comps will be included to the output of source jsp page

Working with response.sendRedirect(-) in jsp page

note: <jsp:forward> ,<jsp:include > tags allows only html,jsp,servlet comps as dest web comps ..more over they must be there in same web application where source jsp page is placed. So to take Dest comp in our choice tehnology (asp.net, php, js and etc..) and to place dest comp as Local /Remote comp we need to use "response.sendRedirect(-) method " (performs sendRedirection mode of communication)



example app

input.html

```
<h1 style="color:red;text-align:center"> Send Redirection </h1>
<form action="search.jsp"  method="POST">
  <table border="0" align="center" bgcolor="cyan" >
    <tr>
      <td>search String:: </td>
      <td><input type="text" name="ss"> </td>
    </tr>

    <tr>
      <td colspan="2"><input type="submit" value="Google search"> </td>
    </tr>


  </table>
</form>
```

search.jsp

```
<%
 //read form data
 String ss=request.getParameter("ss");
 //perform sendRedirction with Google
 System.out.println("before");
 response.sendRedirect("https://www.google.com/search?q="+ss);
 System.out.println("after");
 %>
```

response.sendRedirect(-) usecases

=> To redirect one company web requests to another company website if one company acquires another company
eg:: request given to sun.com will be redirected oracle.com

=> To use remote website services in current website irrepsetive of the location and technology of remote website.
=>Local website using Google search services
=>Local website using Google Map services

What happens if place multiple response.sendRedirect(-) methods in single source jsp page?

Ans)The second response.sendRedirect(-) throws Exception

```
Root Cause
java.lang.IllegalStateException: Cannot call sendRedirect() after the response has been committed
```

What is happens if we place response.sendRedirect(-) and <jsp:forward> tags in a source jsp page?

Ans) if <jsp:forward> is placed after response.sendRedirect(-) method then the

```
Root Cause
java.lang.IllegalStateException: Cannot call sendRedirect() after the response has been committed
```

will be raised ..

if <jsp:forward> is placed before response.sendRedirect(-) method then the the statements after <jsp:forward> tag will not be executed becoz return statement after rd.forward(-,-) for <jsp:forward> tag in JES class..

(overall we can response.sendRedirect(-) method call statement will not be executed)

What is happens if we place response.sendRedirect(-) and <jsp:include> tags in a source jsp page?

of

Ans) The effect <jsp:include> will not be there.. i.e only the effect of response.sendRedirect(-) method executes becoz response. sendRedirect(-) discards both direct and included html/template text outputs of source jsp page.

What happens if we place both <jsp:forward> and <jsp:include>tags in a source jsp page ?

Ans) The effect of <jsp:include> will not there.. becoz if <jsp:foward> is placed before <jsp:include>tag then <jsp:include> will not executed beoz the "return" statement that comes for <jsp:forward> tag if <jsp:forward> tag is placed after <jsp:include> tag then also no effect of <jsp:include> takes beocz <jsp:forward> discards both direct output and included output of source jsp page.

Q) What is the difference between <jsp:forward> and response.sendRedirect(-) mehod call.

<jsp:forward>	response.sendRedirect(-)
a)performs the forwaring request mode of jsp communication	a)performs the sendRedirection mode of jsp communication
b)Soruce jsp page directly interacts with Dest comp	b) source jsp page interacts with dest comp after having a network round trip with browser
c) Source jsp page and dest web comp uses same req,res objs	c) will not use same req, res objs
d) source jsp page and and dest web comp must be there in same web application	d) source jsp page and and dest web comp can be there in same web application or in two different web applications of same or different servers belonging same of differnt machines
e) To pass addtional data from source jsp page to dest web comp we can <jsp:param> tag or request attributes	e) To pass addtional data from source jsp page to dest web comp we append queryString to url if response.sendRedirct(-) method
f) Dest web comp can be html/jsp/servlet comp	f) Dest web comp can be html/jsp/servlet comp /php/asp.net/js comp and etc..
g) source jsp page can forward the request to Dest web cmp in both "GET","POST" modes	g) source jsp page redirects the request to dest web comp always in "GET" mode.
(h) while forwarding request operation ,the url in browser address bar will not be changed	(h) while performing sendRediction . ,the url in browser address bar will be changed.
i) internally uses rd.forward(-,-) and it is having tag <jsp:forward> tag called	i)It is java statements code with no jsp tag..
j) Useful for exception management	j) Useful for redirecting one websiste requests to another website .

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

Ans)

if source jsp page and dest web comp are using same req,res objs

a) request attribiutes (scope is request)
b) session attribiutes (scope is session)
c) application attribiutes (scope is application)
d) page attribiutes (scope is page scope- with in jsp page)

=>So for we have used different objects like request, response.page and etc.. to create attributes on your own having different scope ,where as in jsp page we can use single object "pageContext" to create all the 4 scopes of attributes.

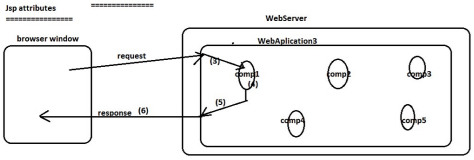
not

if source jsp page and dest web comp are using same req,res objs

(a) append query string to the url of response.sendRect(-) method

problem code::

Jsp attributes



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

Ans)

if source jsp page and dest web comp are using same req,res objs

a) request attributes (scope is request)
b) session attributes (scope is session)
c) application attributes (scope is application)
d) page attributes (scope is page scope- with in jsp page)

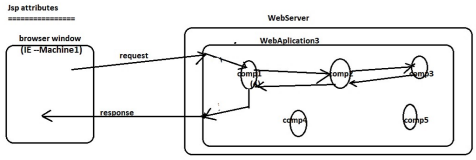
=>So far we have used different objects like request, response.page and etc.. to create attributes on your own having different scope ,where as in jsp page we can use single object "pageContext" to create all the 4 scopes of attributes.

not

if source jsp page and dest web comp are using same req,res objs

(a) append query string to the url of response.sendRect(-) method

Jsp attributes



=>The request attribute created in comp1 is visible and accesisble in comp2 and comp3 .. but not in comp4 and comp5 (request scope --> visible through out request)

=>The page attribute created in comp1 is visible and accessible in comp1 but not in other comps (page scope --> specific to each jsp page)

=>The session attribute created in comp1 by getting request from IE browser s/w of Machine1 is visible and accessible in all web comps of webapplication only when they get request from same browser IE of machine1 (session scope --> global within the web application but specific to a browser s/w of client machine)

=>The application attribute created in comp1 by getting request from any browser s/w of any machine is visible and accesisble in all web comps of webapplication irrespective of any conditions. (application scope -->global with in the web application for no conditions)

Using 1 pageContext obj we can create and read all the 4 scopes of attributes becoz pageContext holds all implicit objs of jsp page internally

To create pageContext attribute

```
pageContext.setAttribute("attr1","val1",PageContext.SESSION_SCOPE);
                         attr name   attr value        attr scope
```

->creates attr1 having the value "val1" with Session scope

all possible scopes are

```
pageContext.SESSION_SCOPE
pageContext.REQUEST_SCOPE
pageContext.PAGE_SCOPE   (default)
pageContext.APPLICATION_SCOPE
```

```
pageContext.setAttribute ("attr2","val2");
```

->creates attr2 having the value "val2" with default page scope

To modify pageContext attribute values

```
pageContext.setAttribute("attr1","val1111",pageContext.SESSION_SCOPE);
```

=>modifies "attr1" attribute vlaue having scope session scope

```
pageContext.setAttribute("attr2","val2222");
```

=>modifies "attr2" attribute vlaue having scope pageScope

To read pageContext attribute value

```
String value=(String)pageContext.getAttribute("attr1",pageContext.SESSION_SCOPE);
```

==>reads and displays "attr1" value from Session scope

```
String value1=(String)pageContext.getAttribute("attr2");
```

==>reads and displays "attr2" value from page scope

```
String value2=(String)pageContext.getAttribute("attr2",pageContext.SESSION_SCOPE);
```

=>Gives null value becoz "attr2" is not there in session scope

---

To find and read pageContext attribute values

=>This method searches and gets given attribute value in following order

pageScope --x--> request scope ---x--> session scope --------x-------> application scope

```
String val1=(String) pageContext.findAttribute("attr1");
String val2=(String) pageContext.findAttribute("attr2");
```

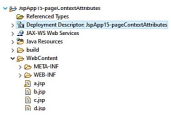
=>Searches and gives attr1,attr2 values in multiple scopes in a order pageScope--->requestScope -->session scope --> application Scope

What is difference b/w invoking getAttribute(-) and invoking findAttribute(-,-) methods on page Context object.

=> getAttribute(-) searches and gets the given attribute value from the specified scope
=> findAttribute(-) searches and gets the given attribute value from the different scopes page --->request ---> session ---> application.

To reomove pageContext attributes

```
pageContext.removeAttribute("attr1");
pageContext.removeAttribute("dean"))
and etc..;
```




```
<!-- a.jsp -->
<!-- a.jsp -->
<b> from a.jsp</b>

<%
//create PageContext attributes having pageScope
   pageContext.setAttribute("attr1","val1", pageContext.PAGE_SCOPE);
pageContext.setAttribute("attr2","val2", pageContext.REQUEST_SCOPE);
pageContext.setAttribute("attr3","val3", pageContext.SESSION_SCOPE);
pageContext.setAttribute("attr4","val4", pageContext.APPLICATION_SCOPE);

%>
<jsp:forward page="b.jsp" />
```

```
<!-- b.jsp -->
<!-- Read and display the attribute vlaues -->
<b> from b.jsp</b>
  attr1 (pageScope)  value :: <%=pageContext.findAttribute("attr1") %>
  attr2 (request)  value :: <%=pageContext.findAttribute("attr2") %>
  attr3 (sessionScope)  value :: <%=pageContext.findAttribute("attr3") %>
  attr3 (application scope)  value :: <%=pageContext.findAttribute("attr4") %>


<jsp:forward page="c.jsp" />
```

The rquest attribute values having request scope i.e they are through out request
The session attribute values having session scope i.e they are specific to browser
The appliation attributes having global scope are visible to all web comps with out worrying about they request,response objs and from browser they are getting reequest

```
<!-- d.jsp -->
<!-- Read and display the attribute vlaues -->
<b> from d.jsp</b>
  attr1 (pageScope)  value :: <%=pageContext.findAttribute("attr1") %> <br>
  attr2 (request)  value :: <%=pageContext.findAttribute("attr2") %> <br>
  attr3 (sessionScope)  value :: <%=pageContext.findAttribute("attr3") %>  <br>
  attr4 (application scope)  value :: <%=pageContext.findAttribute("attr4") %> <br>
```

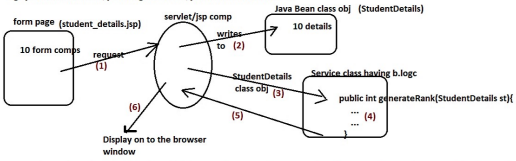
Execution :: Give 1 st request a.jsp and other requests c.jsp and d.jsp

JAva Bean class
==========
=> Java bean is a java class that is developed by following standards
=> It is always used as helper class to pass multiple values between the classes of same project or
between classes of diffrent projects.. It is like office boy in the bank

Java Bean vedios ::
=>https://www.youtube.com/watch?v=lo2TciOcF2Y
=>https://www.youtube.com/watch?v=GOvak6awDrI

standards are
==========
->class must be public class
-> Recomanded to implement Serializable interface
-> private and NON-static properties (member variables)
-> should have 0-param constructor directly or indirectly
> every proeprty should have seperate setter method and seperate getter

=>The interface that makes underlying JVM /server /Container/ Framework to provide special
run time capabilities to the implemetation class objects is called marker interface.
eg: java.io.Serializable , java.lang.Cloneable, java.rmi.Remote and etc..



In real practices , the multiple values will be passed from one class to another class or one java project to another java project in the form of Java ben class object.

=>In the above discussions,
->we need servlet to Java Bean interaction and jsp to java bean interaction to bind the recived
form data to Java bean class object

=>For servet to java bean interaction , we need to go for manual java code based logics
StudentDetails st=new StudentDetails();
st.setSno(011); st.setSname("raja"); st.setSadd("hyd");
....
...

=> For jsp to Java bean interaction, we can use jsp tags, they are
<jsp:useBean> ,<jsp:setProperty> ,<jsp:getProperty>

<jsp:useBean>
=============
=>Given to create or Locate java bean class object from /to specified scope.

syntax:: <jsp:useBean attributes/>

attributes
==========
a) id :: bean id internally java bean calss object name
b) class :: fully qualified java bean class name
c) scope :: bean class obj scope
page(default) | request | session | application
d) type :: To specify seperate reference type for java bean class obj

```
package com.nt.beans;
public class StudentDetails{
  //bean properties
  private int sno;
  private String sname;
  private String sadd;
  private String location;

 //setters && getters
    ....
    ....
    ....
}
```

eg1:: <jsp:useBean id="st" class="com.nt.beans.StudentDetails" scope="session"/>
=>creates StudentDetails class obj having name "st" and places that obj in session scope.
if object is already available with the name "st" in session scope then it locates and
returns the object.

In JES class
==========

```
StudentDetails st=null;
  //Locate bean class obj from session scope
  st=pageContext.getAttribute("st",pageContext.SESSION_SCOPE);
  if(st==null){
       //create bean class obj and keep in session scope
     st=new StudentDetails();
     session.setAttribute("st",st, pageContext.SESSION_SCOPE);
   }
```

eg2:: <jsp:useBean id="st1" class="com.nt.beans.StudentDetails" type="com.nt.beans.PersonDetails" scope="request"/>
Creates/Locates StudentDetails obj having "PersonDetails" as the ref type
and "StudentDetails" as the object type and keeps or locates in /from request scope

PersonDetails (C)
^
|
extends
|
StudentDetails (C)

JES class code
==========

```
PersonDetails st1=(PersonDetails)pageContext.getAttribute("st1",pageContext.REQUEST_SCOPE);
 if(st1==null){
    st1=new StudentDetails();
    request.setAttribute("st1",st1,pageContext.REQUEST_SCOPE);
 }
```

"st1" reference type is:: PersonDetails
"st1" object type is:: StudentDetails

<jsp:setProperty>
==========
=>Internally calls setXxx(-) methods for the specified property to assign given data to bean property.

syntax :: <jsp:setProperty attributes />

a) name :: bean id / bean class obj name given in "id"attribute of <jsp:useBean>
b) property :: bean property name (xxx word of setXxx(-) method)
c) value :: The value to be assigned
d) param :: Makes the request param value(form data) as bean property value
note:: we can either "value" or "param" attribute to set data to bean properties.

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

<jsp:setProperty name="st" property="sname" param="name" />
calls set.setSname(request.getParameter("name")) to assign "name" request param value
to "sname" property.

<jsp:property>
==========
=>calls getter methods on bean clas obj to read and use bean property values

syn: <jsp:getProperty attributes />

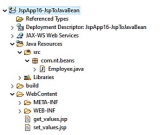
attributes :: a) name :: bean id can be collected <jsp:useBean> tag's id attribute
b) property :: the bean property name

<jsp:getProperty name="st" property="sname"/>
calls "st.getSname()" method will be called internally.

mvc vedios::
https://www.youtube.com/watch?v=s07-b1tdtkU
https://www.youtube.com/watch?v=HeABAGNLjPw&t=18s
https://www.youtube.com/watch?v=_kKEjRUqVqs
https://www.youtube.com/watch?v=s07-b1tdtkU
https://www.youtube.com/watch?v=s07-b1tdtkU
https://www.youtube.com/watch?v=TiTJ9V1PHR4

JAva Bean vedios ::
https://www.youtube.com/watch?v=lo2TclOcF2Y
https://www.youtube.com/watch?v=GOvak6awDrI&t=161s

Example App jsp to JAva Bean interaction



set_values.jsp

```
<!-- create or Locate java bean class obj -->
<jsp:useBean id="emp" class="com.nt.beans.Employee" scope="session"/>

<!-- set values to bean properties  -->
<jsp:setProperty name="emp" property="eno" value="10001"/>
<jsp:setProperty name="emp" property="ename" value="rajesh"/>
<jsp:setProperty name="emp" property="eadd" value="hyd"/>
<jsp:setProperty name="emp" property="esalary" value="90000"/>

<b>Values are set to bean properties</b>
```

get_values.jsp

```
<!-- create or Locate Java bean class obj -->
<jsp:useBean id="emp" class="com.nt.beans.Employee" scope="session"/>

<!-- Read and display java bean class obj properties data -->
<b>eno :: <jsp:getProperty name="emp" property="eno"/></b> <br>
<b>ename :: <jsp:getProperty name="emp" property="ename"/></b> <br>
<b>eadd :: <jsp:getProperty name="emp" property="eadd"/></b> <br>
<b>esalary :: <jsp:getProperty name="emp" property="esalary"/></b> <br>


<br><br>
<b> Java bean class object data is retrieved and displayed</b>
```

urls from same browser s/w

http://localhost:2525/JspApp16-JspToJavaBean/set_values.jsp
http://localhost:2525/JspApp16-JspToJavaBean/get_values.jsp

---

setting request param values (form data) to Java bean class object property values

set_values.jsp

```
<!-- create or Locate java bean class obj -->
<jsp:useBean id="emp" class="com.nt.beans.Employee" scope="session"/>


 <!-- To set request param values as java bean property values we can use "param" attribute in place of "value" -->
<jsp:setProperty name="emp" property="eno" param="empno"/>
<jsp:setProperty name="emp" property="ename" param="empname" />
<jsp:setProperty name="emp" property="eadd" param="empadd"/>
<jsp:setProperty name="emp" property="esalary" param="empsalary"/>
                          bean property          request param name
<b>Values are set to bean properties</b>  name
```

we can use either param or value attribute at a time

request url ::

http://localhost:2525/JspApp16-JspToJavaBean/set_values.jsp?empno=1001&empname=raja&empadd=hyd&empsalary=90000
http://localhost:2525/JspApp16-JspToJavaBean/get_values.jsp request params

---

note:: To set request param values as java bean property values we can use property="*" if the bean property names are matching with req param names

set_values.jsp

```
<!-- create or Locate java bean class obj -->
<jsp:useBean id="emp" class="com.nt.beans.Employee" scope="session"/>

<!-- To set request param values as java bean property values we can use property="*"
                  if the bean property names are matching with req param names-->
<jsp:setProperty name="emp" property="*"/>

<b>Values are set to bean properties</b>
```

request urls::
http://localhost:2525/JspApp16-JspToJavaBean/set_values.jsp?eno=1111&ename=rajesh&eadd=vizag&esalary=78999
http://localhost:2525/JspApp16-JspToJavaBean/get_values.jsp

Should match with bean property names..

Different models/architectures of Java web application devleopment

Model1

=>Here only servlet comps or only jsp comps will be used as the main comps i.e if servlet comps are used as the main web comps then jsp comps will not be used and vice-versa,
(or)
=>Model1 architecture more of jsp comps based web application development

=> So far we are developing web applicaitons either having only servlet comps as main comps or only jsp comps as main comps. i.e we haved developed model1 architecture based web applications sofar.

=> Here every main web comp contains multiple logics like presentation logic,b.logic. persistence logic and etc...i.e we are mixing up multiple logics and there is no clean seperation b/w logics.

Model2 (MVC)

MVC1 | MVC2

MVC :: MODEL-VIEW-CONTROLLER

M --->Model Layer -->represents data ..
-->contains b.logic +persistence logic (like accounts officer)

V --->View Layer --->contains presentation logic (like beaitician)

C-->Controller layer ---> contains Integration/Monitoring logic (like superviser)

=>Here we develop web application as the layered web application by keeping multiple logics in multiple layers using differen technologies

=>Clean seperation b/w logics is possiable becoz of multiple layers.

=>these are industry standard architectures becoz they allow to devleop web application as layered App. (especially MVC2)

What is the difference b/w MVC1 and MVC2?

what are the MVC2 Priciples?

Employee.java (java bean)

```
package com.nt.beans;

public class Employee {
        //bean properies
        private int eno;
        private String ename;
        private  String eadd;
        public  float  esalary;

        public Employee() {
                System.out.println("Employee:: 0-param constructor");
        }

        //getters && setters  (alt+shit+S, r)
        public int getEno() {
                System.out.println("Employee.getEno()");
                return eno;
        }
        public void setEno(int eno) {
                System.out.println("Employee.setEno()");
                this.eno = eno;
        }
        public String getEname() {
                System.out.println("Employee.getEname()");
                return ename;
        }
        public void setEname(String ename) {
                System.out.println("Employee.setEname()");
                this.ename = ename;
        }
        public String getEadd() {
                System.out.println("Employee.getEadd()");
                return eadd;
        }
        public void setEadd(String eadd) {
                System.out.println("Employee.setEadd()");
                this.eadd = eadd;
        }
        public float getEsalary() {
                System.out.println("Employee.getEsalary()");
                return esalary;
        }
        public void setEsalary(float esalary) {
                System.out.println("Employee.setEsalary()");
                this.esalary = esalary;
        }
          @Override  // alt+shift+s, s
          public String toString() {
                return "Employee [eno=" + eno + ", ename=" + ename + ", eadd=" + eadd + ", esalary="
        + esalary + "]";
          }


}
```

MVC1 Architectire Vs MVC2 Architecture

==========

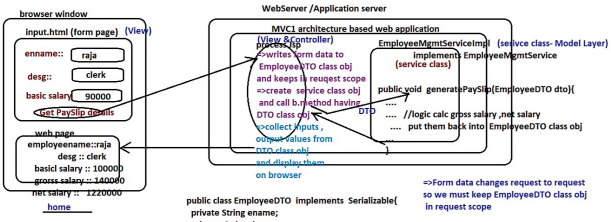
=>In MVC1 architecture single comp will ben taken as view and controller comp and seperate comps as model comps

=>In MVC2 architecture , we take seperate comps for views and seperate comp controlelr and alos other sperarate comps as model comp

=>Use Model1 architecture for developing small scale web Apps (<10 web pages)
=>Use MVC1 architecture for developing medium scale web Apps (>=10 and<20 web pages)
=>Use MVC2 architecture for developing large web Apps (>=20 web pages)

Example app MVC1 architecutre

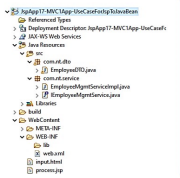
==========




```
public class EmployeeDTO  implements  Serializable{
 private String ename;
 private  String desg;
 private double basicSalary,grossSalary,netSalary;
  //setters && getters
   ......
   .....
}
```

=>Form data changes request to request so we must keep EmployeeDTO class obj in request scope

=>Since the service class is taken with out side .. It is better to create single obj for Service class by keeping in application scope.



EmployeeDTO.java

```
package com.nt.dto;

import java.io.Serializable;

public class EmployeeDTO implements Serializable {
	private String ename;
	private String desg;
	private double basicSalary;
	private  double  grossSalary;
	private  double  netSalary;

	public EmployeeDTO() {
		System.out.println("EmployeeDTO:: 0-param construtor");
	}

	//setters & getters
	public String getEname() {
		return ename;
	}
	public void setEname(String ename) {
		this.ename = ename;
	}
	public String getDesg() {
		return desg;
	}
	public void setDesg(String desg) {
		this.desg = desg;
	}
	public double getBasicSalary() {
		return basicSalary;
	}
	public void setBasicSalary(double basicSalary) {
		this.basicSalary = basicSalary;
	}
	public double getGrossSalary() {
		return grossSalary;
	}
	public void setGrossSalary(double grossSalary) {
		this.grossSalary = grossSalary;
	}
	public double getNetSalary() {
		return netSalary;
	}
	public void setNetSalary(double netSalary) {
		this.netSalary = netSalary;
	}

}
```

IEmployeeMgmtService.java

```
package com.nt.service;

import com.nt.dto.EmployeeDTO;

public interface IEmployeeMgmtService {
	public void  generatePayslipDetails(EmployeeDTO dto) ;
}
```

EmployeeMgmtServiceImpl.java

```
package com.nt.service;

import com.nt.dto.EmployeeDTO;

public class EmployeeMgmtServiceImpl implements IEmployeeMgmtService {

	public EmployeeMgmtServiceImpl() {
		System.out.println("EmployeeMgmtServiceImpl:: 0-param constructor");
	}

	@Override
	public void generatePayslipDetails(EmployeeDTO dto) {
		//calculate gross Salary and netSalary
		double  grossSalary=dto.getBasicSalary()+(dto.getBasicSalary()*0.4);
		double  netSalary=grossSalary- (grossSalary*0.2);
		//set gross and netSalary details to DTO class obj
		dto.setGrossSalary(grossSalary); dto.setNetSalary(netSalary);
	}

}
```

process.jsp

```
<%@page import="com.nt.dto.EmployeeDTO"%>
<%@ page isELignored="false" %>

<!-- read form data and store into EmployeeDTO class object  -->
<jsp:useBean id="empDTO" class="com.nt.dto.EmployeeDTO" scope="request"/>
<jsp:setProperty name="empDTO"  property="*" />

<!--  create service class object -->
<jsp:useBean id="empService" class="com.nt.service.EmployeeMgmtServiceImpl" type="com.nt.service.IEmployeeMgmtService" scope="application"/>

<!-- invoke business method -->
 <%
   empService.generatePayslipDetails(empDTO);
 %>


<!-- Display results -->
<h1 style="color:red;text-align:center"> Employee Details  </h1>
<b> Employee name  ::  <jsp:getProperty property="ename" name="empDTO"/> </b> <br>
<b> Employee desg  ::  <jsp:getProperty property="desg" name="empDTO"/> </b> <br>
<b> Employee basicSalary  ::  <jsp:getProperty property="basicSalary" name="empDTO"/> </b> <br>
<b> Employee grossSalary  ::  <jsp:getProperty property="grossSalary" name="empDTO"/> </b> <br>
<b> Employee netSalary  ::  <jsp:getProperty property="netSalary" name="empDTO"/> </b> <br>

<!--  home hyperlink -->
<a href="input.html">home</a>
```

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

Problem:: => we can not send JDBc ResultSet obj over the n/w from one Java Project to another Java Project becoz it is not Serializable obect

=> In Layered Apps like MVC1,MVC2 Application we can not move ResultSet object to every layer becoz writing JDBC code in All layers is bad pratice

Solutin1:: Stop using ResultSet and start using RowSet (bad pratice)
=> RowSets are Serializable objects by default
=>Every few JDBC drivers support Rowsets

Solution2:: Copy ResultSet obj records to Collection(List/Set/Map) and send Collection from 1 layer to another layer or 1 from project to another Project. (Best solution)

=>All collections are serializable objects by default
=> All JDBC drivers support ResultSet
=>Sending collection from 1 layer to another layer either within the same Project or acrosss the multiple projects is possible becoz collections will be treated normal java code to use.

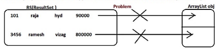
=>We can generaly ResultSet object records to ArrayList Collection becoz
a) It presevers the insertions
b) can access the elements through index
c) Sorting is possible
d) It is not symchronized collection .. while accessing records we can use multiple threads simultaenously.

note1:: After placing data in Collection .. if u r planning to perform both read and write operations then use prefer synchronized collections for thread safety becoz they allow 1 thread at time and we can avoid data corruption.
eg:: Vector , Hashtable and etc..

note2:: After placing data in Collection .. if u r to perform only read operations then use prefer non-synchronized collections for perfomance
eg:: ArrayList , Hashmap and etc..

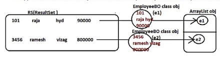
Copying ResultSet object records to ArrayList Collection
==

=>Each Record of REsultSet container muliple values and where as element of ArrayList allows only object , so we can not copy each record of RS to each element of ArrayList directly.



Solution::
==========

=>Copy each record of RS to a object of Java bean class (BO class) and add that object List collection element.

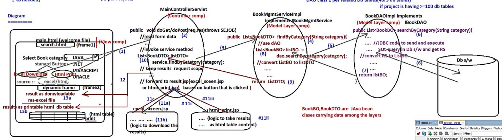


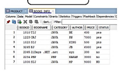
Typical java based web application as layered App based on MVC2 architecture
===

html/jsp page -----request obj data / List of DTO objs----->ControllerServlet -----DTO class obj / List of DTO objs----->SErvice class/BusinessDelegate class -----BO class object / List of BO objs----->DAO class -----SQL query / RS multiple records-----> Db s/w

(presentation logic/ UI logics) [as needed]
(Monitoring logic/ Controlling logic) [1 per Project]
[B.logic /service logic -->calcs,sorting, TxMgmt, filtering] [Service class is 1 module]
[Persistence logic-CURD operations]
[DAO class 1 per Db table if project is having <100 db tables
DAO class 1 per related Db tables(4or5 db tables) if project is having >=100 db tables

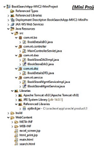
Diagram
==========





Points to be observed
=====================

(a) Developed based MVC2 Aritecture based Layered App
(b) These Layer getting and passing data as Java bean calss objs (DTO ,BO class objs)
(c) DAO class is converting RS object records to ListBO class objs to send to next layer called service class
(d) JAva script will be used for submtting requests from the form page, to perform form validation and to take printout of the webpage
(e) Options are given to see reusits in different formats (a) As downloadable ms-excel file (b) Html table style report
(f) Using Frames we are displaying results in the web page from where request is generated.



(Mini Project based on MVC2 architecture)

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

Problem:: => we can not send JDBc ResultSet obj over the n/w from one Java Project to another Java Project becoz it is not Serializable obect

=> In Layered Apps like MVC1,MVC2 Application we can not move ResultSet object to every layer becoz writing JDBC code in All layers is bad pratice

Solutin1:: Stop using ResultSet and start using RowSet (bad pratice)
=> RowSets are Serializable objects by default
=>Every few JDBC drivers support Rowsets

Solution2:: Copy ResultSet obj records to Collection(List/Set/Map) and send Collection from 1 layer to another layer or 1 from project to another Project. (Best solution)

=>All collections are serializable objects by default
=> All JDBC drivers support ResultSet
=>Sending collection from 1 layer to another layer either within the same Project or acrosss the multiple projects is possible becoz collections will be treated normal java code to use.

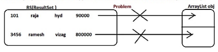
=>We can generaly ResultSet object records to ArrayList Collection becoz
a) It presevers the insertions
b) can access the elements through index
c) Sorting is possible
d) It is not symchronized collection .. while accessing records we can use multiple threads simultaenously.

note1:: After placing data in Collection .. if u r planning to perform both read and write operations then use prefer synchronized collections for thread safety becoz they allow 1 thread at time and we can avoid data corruption.
eg:: Vector , Hashtable and etc..

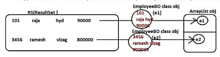
note2:: After placing data in Collection .. if u r to perform only read operations then use prefer non-synchronized collections for perfomance
eg:: ArrayList , Hashmap and etc..

Copying ResultSet object records to ArrayList Collection
==

=>Each Record of REsultSet container muliple values and where as element of ArrayList allows only object , so we can not copy each record of RS to each element of ArrayList directly.



Solution::
==========
=>Copy each record of RS to a object of Java bean class (BO class) and add that object List collection element.

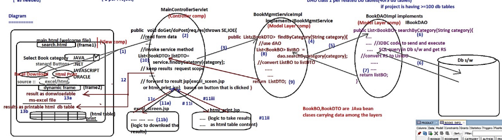


Typical java based web application as layered App based on MVC2 architecture
==

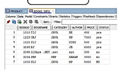
html/jsp page ----request obj data / List of DTO objs---->ControllerServlet ----DTO class obj / List of DTO objs---->SErvice class/BusinessDelegate class ----BO class object / List of BO objs---->DAO class ----SQL query / RS multiple records----> Db s/w

(presentation logic/ UI logics) [as needed]
(Monitoring logic/ Controlling logic) [1 per Project]
[B.logic /service logic -->calcs,sorting, TxMgmt, filtering] [Service class is 1 module]
[Persistence logic-CURD operations]
[DAO class 1 per Db table if project is having <100 db tables
DAO class 1 per related Db tables(4or5 db tables) if project is having >=100 db tables

Diagram
==========



BookBO,BookDTO are JAva bean clases carrying data among the layers



	BOOKID	BOOKNAME	CATEGORY	AUTHOR	PRICE	STATUS
1	1010	TIJ	JAVA	BE	400	yes
2	1223	CRJ	JAVA	HS	7000	yes
3	1023	SIJ	JAVA	KING	500	yes
4	3245	EJ	JAVA	JB	6000	yes
5	2345	21Days .NET	.net	xyz	300	no
6	2234	PHP	PHP	KARAN	3000	no
7	1000	SIJ	JAVA	HP	5000	NO

Points to be observed
====================

(a) Developed based MVC2 Aritecture based Layered App
(b) These Layer getting and passing data as Java bean calss objs (DTO ,BO class objs)
(c) DAO class is converting RS object records to ListBO class objs to send to next layer called service class
(d) JAva script will be used for submtting requests from the form page, to perform form validation and to take printout of the webpage
(e) Options are given to see reusits in different formats (a) As downloadable ms-excel file (b) Html table style report
(f) Using Frames we are displaying results in the web page from where request is generated.

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

Problem:: => we can not send JDBc ResultSet obj over the n/w from one Java Project to another Java Project becoz it is not Serializable obect

=> In Layered Apps like MVC1,MVC2 Application we can not move ResultSet object to every layer becoz writing JDBC code in All layers is bad pratice

Solutin1:: Stop using ResultSet and start using RowSet (bad pratice)
=> RowSets are Serializable objects by default
=>Every few JDBC drivers support Rowsets

Solution2:: Copy ResultSet obj records to Collection(List/Set/Map) and send Collection from 1 layer to another layer or 1 from project to another Project. (Best solution)

=>All collections are serializable objects by default
=> All JDBC drivers support ResultSet
=>Sending collection from 1 layer to another layer either within the same Project or acrosss the multiple projects is possible becoz collections will be treated normal java code to use.

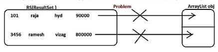
=>We can generaly ResultSet object records to ArrayList Collection becoz
a) It presevers the insertions
b) can access the elements through index
c) Sorting is possible
d) It is not symchronized collection .. while accessing records we can use multiple threads simultaenously.

note1:: After placing data in Collection .. if u r planning to perform both read and write operations then use prefer synchronized collections for thread safety becoz they allow 1 thread at time and we can avoid data corruption.
eg:: Vector , Hashtable and etc..

note2:: After placing data in Collection .. if u r to perform only read operations then use prefer non-synchronized collections for perfomance
eg:: ArrayList , Hashmap and etc..

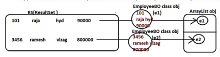
Copying ResultSet object records to ArrayList Collection
===

=>Each Record of REsultSet container muliple values and where as element of ArrayList allows only object , so we can not copy each record of RS to each element of ArrayList directly.



Solution::
==========

=>Copy each record of RS to a object of Java bean class (BO class) and add that object List collection element.

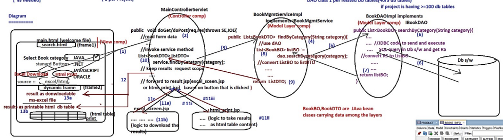


Typical java based web application as layered App based on MVC2 architecture
==

html/jsp page ----request obj data / List of DTO objs---->ControllerServlet ----DTO class obj / List of DTO objs---->SErvice class/BusinessDelegate class ----BO class object / List of BO objs---->DAO class ----SQL query / RS / multiple records----> Db s/w

(presentation logic/ UI logics) [as needed]
(Monitoring logic/ Controlling logic) [1 per Project]
[B.logic /service logic -->calcs,sorting, TxMgmt, filtering] [Service class is 1 module]
[Persistence logic-CURD operations] [DAO class 1 per Db table if project is having <100 db tables DAO class 1 per related Db tables(4or5 db tables) if project is having >=100 db tables

Diagram
==========



	BOOKID	BOOKNAME	CATEGORY	AUTHOR	PRICE	STATUS
1	1010	TIJ	JAVA	BE	400	yes
2	1223	CRJ	JAVA	PH	7000	yes
3	1023	SIJ	JAVA	KING	500	yes
4	3245	EJ	JAVA	JB	6000	yes
5	2345	21Days .NET	.net	xyz	300	no
6	2234	PHP	PHP	KARAN	3000	no
7	1000	SIJ	JAVA	MP	9000	NO

Points to be observed
====================

(a) Developed based MVC2 Aritecture based Layered App
(b) These Layer getting and passing data as Java bean calss objs (DTO ,BO class objs)
(c) DAO class is converting RS object records to ListBO class objs to send to next layer called service class
(d) JAva script will be used for submtting requests from the form page, to perform form validation and to take printout of the webpage
(e) Options are given to see results in different formats (a) As downloadable ms-excel file (b) Html table style report
(f) Using Frames we are displaying results in the web page from where request is generated.

CUSTOM Jsp Tag Library
=====================

Placing Java code/script code in jsp page is bad pratice and the limitations are

a) Kills the redability of js page
b) Not suitable for UI Developers whose works with Jsp pages
c) Against of Jsp programming principles (jsp pages must be developed as scriptletless pages)
d) Kills the reusability of java code that is placed in on jsp age.. across the multiple other jsp pages
d) No a indudstry standard..

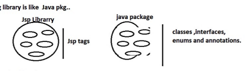
To overcome thse develop jsp pages as java code less/scriptless jsp pages by using the following options

a) Jsp Directive tags and standard Action tags (avoid scripting tags -scriptlet, expression, declaration)
b) Html tags
c) EL (Expression Language)
d) JSTL tags
e) Third pary tags (Struts tags, spring MVC tags and etc..)
f) Custom tags

=>Jsp Tag Library is a library that contains set of readily avaiable jsp tags/elements.
=> 4 types Jsp taglibaries

a) Built-in jsp tag library (part of jsp pages)
<jsp:useBean> ,<jsp:setProperty>,<jsp:forward> and etc...
b) JSTL (Jsp standard Taglibary)
=>Server supplied jsp tags for the specification given Sun Ms
c) Custom Jsp taglibary (developed by the progammmer)
d) third party Jsp Library (Given by third party f/w lik struts ,spring MVC and etc..)

=> Jsp tag library is like Java pkg..



=> Jsp Tag handler class
->The java class that defines the functionality of jsp tag is Jsp tag handler class.
-> This must extends javax.servlet.jsp.tagext.TagSupport(-) class directly or indirectly.
=> Every jsp tag handler class contains multiple callback methods i.e the methods that will be called by underlying Container (like jspcontainer/servlet container) dynamically.

```
public class ABCTag extends TagSupport{

    public int startTag(){   (do)
        .... //logic for open jsp tag    (will be called automatically for open <ABC> tag)
        ....
    return ...;
    }
    public int endTag(){   (do)
        .... //logic for closing jsp tag    (will be called automatically for closing </ABC> tag)
        ....
    return ...;
    }
}
```

=> Every Custom jsp tag and details must be mapping tag Handler class and tits in tld file(xml file)
Tag library Descriptor

Steps to develop and use Custom Jsp Library in web application
===

step1) Design Jsp Library

NIT Taglibrary
|------------> <ABC>
|------------> <XYZ>

step2) Develop Jsp Tag handler classes

WEB-INF/classes/ABCTag.java (src/com/nt/tags/ABCTag.java)

```
package com.nt.tags;

public class ABCTag extends TagSupport{

    public int doStartTag(){
        .... //logic for open jsp tag    (will be called automatically for open <ABC> tag)
        ....
    return ...;
    }
    public int doEndTag(){
        .... //logic for closing jsp tag    (will be called automatically for closing </ABC> tag)
        ....
    return ...;

    }
}
```

WEB-INF/classes/Xyz Tag.java (src/com/nt/tags/ XyzTag.java)

```
package com.nt.tags;

public class XyzTag extends TagSupport{

    public int doStartTag(){
(g) |   .... //logic for open jsp tag    (will be called automatically for open <XYZ> tag)
    |   ....
    return ...;
    }
    public int doEndTag(){
(g) |   .... //logic for closing jsp tag    (will be called automatically for closing </XYZ> tag)
    |   ....
    return ...;              (h) -----> writes the generted
                                        output to response object.
    }                                        |
}                                            v (i)
                                        goes to browser
```

step3) Develop tld file Mapping Jsp tags with Tag handler classes..

WEB-INF/nit.tld (xml content)

<ABC> ---------> com.nt.tags.ABCTag (tag handler class)
<XYZ> ---(f)---> com.nt.tags.XYZTag (tag handler class)

step1 to step3 is develop of Custom Jsp TagLibrary

step4) Configure Custom Jsp Library with web.xml file

web.xml

```
<jsp-config>
  <taglib>              www.  (d) ---> Jsp TagLib uri
    <taglib-uri> http://nit.com/tags/nit </talig-uri>
    <taglib-location> /WEB-INF/nit.tld</taglib-location>
  </taglib>                (e) name and location of tld file..
</jsp-config>
```

Every Jsp taglibary is indentified with its taglib uri.

step5) Use the jsp tags of jsp taglibary in the jsp pages by importing jsp taglibrary.

test.jsp
=======

```
<%@taglib uri="http://www.nit.com/tags/nit" prefix="nit"%>
                              (c)
    <nit:ABC> <br>
(b) <nit: XYZ>
      (a)
```

=>To import any jsp library to jsp page , we need to specify its taglib uri with the support of <%@taglib> directive tag
=> <%@taglib %> tag is also useful to import Standard jsp taglibraries , custom jsp taglibraries and Third jsp libaries to the jsp page by specifying their required taglib uris
=> if two jsp tags of two jsp taglibaries are having same jsp tag name then they can be differentiated by using their prefixes.

=>In every jsp tag handler class we can get access to one protected "pageContext" obj as inherited property of TagSupport class. and use that inherited PageContext obj in our tag handler classes to get access to other implicit objs of jsp page.

=> Using pageContext obj we can get access to all the implicit objects of current jsp page in taghandler classes and in other classes ..

	Provides convenient access to error information.
abstract java.lang.Exception	getException() The current value of the exception object (an Exception).
abstract java.lang.Object	getPage() The current value of the page object (In a Servlet environment,
abstract ServletRequest	getRequest() The current value of the request object (a ServletRequest).
abstract ServletResponse	getResponse() The current value of the response object (a ServletResponse).
abstract ServletConfig	getServletConfig() The ServletConfig instance.
abstract ServletContext	getServletContext() The ServletContext instance.
abstract HttpSession	getSession() The current value of the session object (an HttpSession).

=> Once the call back method completes its execution then it gives instruction JspContainer as numeric value asking to perform next activity.

These are int constants of TagSupport(c)
- EVAL_PAGE (6)
- SKIP_BODY (0)
- EVAL_BODY_INCLUDE (1)
- EVAL_BODY_AGAIN (2)
- SKIP_PAGE (5)

Custom Jsp Library (NIT)
|---><nit:message> :: Display one line of messsage
|---><nit: prime> :: displays all primer members b/w 1 to 10
|---><nit: prime n="20"> :: displays all prime members b/w 1 to 20
"attrbte "n" defualt value :: 20

"n" is optional attribute with the defaiult value 10

|---><nit: dispaly font="ariel" size:"40px" >
.... // :: displays the message having size
....
hello
</nit:dispay>

|---><nit: dispaly font="ariel" >
.... // :: displays the message having size defualt 20px
....
hai
</nit:dispay>

font is mandatory attribute
and "size" is attributed with default value 20

Custom Jsp Library (NIT)

|---><nit:message> :: Display one line of messsage
|---><nit: prime> :: displays all primer members b/w 1 to 10
|---><nit: prime n="20"> :: displays all prime members b/w 1 to 20
"attrbte "n" defualt value :: ?n

"n" is optional attribute
with the defaiult value 10

|---><nit: dispaly font="ariel" size:"40px" >
.... // :: displays the message having size
....
hello
</nit:dispay>

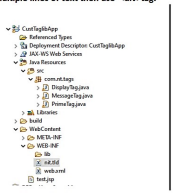
|---><nit: dispaly font="ariel" >
defualt
.... // :: displays the message having size
....
?opx
hai
</nit:dispay>

font is mandatory attribute
and "size" is attributed with default value 20

<n:display family="verdana" size="30px">
hello
<n:display>

<span style="font-family=+family+";font-size="+size>
hello

=>nowadays <font> is deprecated and as an alternate , tags are given
=> To apply styles on one line of text then use tag.. whereas to apply styles of multiple lines of text then use tag.



nit.tld

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
    PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
   "http://java.sun.com/j2ee/dtd/web-jsptaglibrary_1_2.dtd">

<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>2.2</jsp-version>
    <short-name>nit</short-name>

  <tag>
     <name>message</name>
     <tag-class>com.nt.tags.MessageTag</tag-class>
     <body-content>EMPTY</body-content>
  </tag>

  <tag>
     <name>prime</name>
     <tag-class>com.nt.tags.PrimeTag</tag-class>
     <body-content>EMPTY</body-content>
     <attribute>
       <name>n</name>
       <required>false</required>
     </attribute>
  </tag>

  <tag>
     <name>display</name>
     <tag-class>com.nt.tags.DisplayTag</tag-class>
     <body-content>JSP</body-content>
     <attribute>
       <name>family</name>
       <required>true</required>
     </attribute>
     <attribute>
       <name>size</name>
       <required>optional</required>
     </attribute>
  </tag>

</taglib>
```

EL
==

=> Given to avoid or minimize java code in jsp pages
=> snytax :: ${ <expr> }
=>Allows to perform airthemetic and logical operations with out having java code
=>EL code not only evaluates the expression .. but sends the generated output to the browser.

first.jsp

```
<%@page isELIgnored="true" %>
    ${4+5}  -->gives  ${4+5}
```

first.jsp

```
<%@page isELIgnored="false" %>
    ${4+5}  -->gives  9
```

=>Upto Jsp 1.x we can use EL only on the tags.. but Jsp 2.x the EL the can be applied on Jsp tags and Template text.

EL provides
1) EL Operators (same as Java Operators)
2) EL Implicit objs (another 10 objects)
3) EL Functions

EL Operators
===========

=> same as java operators but the same operators can be written having english words

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 objs
==============

=>Along with reqular 9 implicit objs of Jsp .. the EL provides extra 10 implicit objs
They are
requestScope,pageScope,sessionScope,applicationScope,
header,headerValues,param,paramValues,cookie,initParam

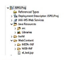
requestScope,pageScope,sessionScope,applicationScope,

=>These objects can be used to read and display different scopes of data given by other web comps of the web application. Every useful in layered Apps.

Example app
==============

TestServlet.java

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

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        //create different scopes of data
            req.setAttribute("attr1", "val1"); //req scope

            HttpSession ses=req.getSession();
            ses.setAttribute("attr2", "val2"); //session scope

            ServletContext sc=getServletContext();
            sc.setAttribute("attr3", "val3");

            //forward to  el_test1.jsp
            RequestDispatcher rd=req.getRequestDispatcher("/el_test.jsp");
            rd.forward(req, res);

    }

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

}
```



```
<%@ page isELIgnored="false" %>

attr1  attribute (request)value :: ${requestScope.attr1} <br>   internally uses request.getAttribute(-)
attr2  attribute (session)value :: ${sessionScope.attr2} <br>   internally uses session.getAttribute(-)
attr3  attribute (application)value :: ${applicationScope.attr3} <br>internally uses application.getAttribute(-)
<hr>
attr1  attribute (request)value :: ${attr1} <br>
attr2  attribute (session)value :: ${attr2} <br>      internally pageContext.findAttribute(-) method
attr3  attribute (application)value :: ${attr3} <br>
```

request url ::   *http://localhost:2525/JSPELProj/testurl*

---

param,paramValues
===================

=>These are given to read one req param and multiple request param values respectively
=> If request param is having one value then use "param" otherwise use "paramValues"
=> example App

el_test1.jsp
===========

```
<%@ page isELIgnored="false" %>

uname  req param value :: ${param.uname} <br>   hyd                  (raja)
sadd req pram multipe values :: ${paramValues.sadd[0]},${paramValues.sadd[1] }   (vizag)
```

request url :: http://localhost:2525/JspProj1?uname=raja&sadd=hyd&sadd=vizag

header, headerValues

=>Given to read request header values (browser gives addtional info to server along with the request as request header values)
=>request header names are fixed and but values will be chnaged based on browser s/w and client machine we are using

=>use "header" object to get specific request header value and use "headerValues" to get all request header names and values

```
<hr>
 user-agent  request header value ::  ${header['user-agent']} <br>
  <br>rquest header values :: ${headerValues}
```

list of request header names are ::
accept , accpet-language ,host,user-agent, referefer ,cookie and etc..

---

init-param
=>given to read context param values (global init params) through Servletcontext obj internally

web.xml
========

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
version="4.0">
  <display-name>JSPELProj</display-name>
   <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>
</web-app>
```

el_test1.jsp
==========

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

cookie
=====
=>useful to read and display cookie name and cookie value

el_test1.jsp

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

=>The session Id of SEssion object goes to browser from web application along with response as cookie and comes back to web application from browser along with the request as cookie.

EL functions
============
=>these functions internally represents java static methods .. i.e we can avoid java code in jsp pages with the support of EL functions.
=>These are alternate to custom jsp tags development towords avoiding or minimizing java code jsp page..

example
=======
step1) take one EL function class having EL function method as the static method.

WishMessageGenerator.java

```java
package com.nt.comp;


public class WishMessageGenerator {

        public static  String generate(String user) {
                return "GoodMorning::"+user;
        }

}
```

step3) cfg EL function with java method in tld file

web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<taglib xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee/web-jsptaglibrary_2_1.xsd" version="2.1"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://java.sun.com/xml/ns/j2ee">
  <description>EL Functions</description>
  <tlib-version>1.0</tlib-version>
  <uri> http://www.nit.com/el </uri>
  <function>
    <description>gives message</description>
    <name>generate</name>
    <function-class>com.nt.comp.WishMessageGenerator</function-class>
    <function-signature>java.lang.String generate(java.lang.String)</function-signature>
  </function>
</taglib>
```

step3) import EL function taglib uri and use the el functions in the jsp page

el_test1.jsp

```
<%page isELIgnored="false" %>
<%@taglib uri="http://www.nit.com/el"  prefix="n1" %>
                                          importing EL tagriry
 message ::   ${n1:generate("raja")}
                  calling EL function
```

JSTL (Java standard Tab Libraries)
==============================

=>Contains set of jsp taglibaries having jsp tags to minimize or avoid java code in jsp page
=> These Taglibaries tags are designed by sun Ms ..but implemented through the development of tag handler calsses by underlying servers like tomcat, weblogic , websphere and etc...

=> In tomcat server upto8.x
- jstl-<ver>.jar
- standard-<ver>.jar

represents JSTL tags impl

=>from tomcat9.x onwords
- taglibs-standard-impl-1.2.5.jar
- taglibs-standard-spec-1.2.5.jar

represents JSTL imol

⇓ Tld files and tag handler classes

location is ::
E:\Tomcat 9.x\webapps\examples\WEB-INF\lib folder

JSTL gives 5 jsp taglibraries
============================

a) Core TagLibary (basic control flow opeartions) ✓
b) SQL Tag Librari (For DB interaction)
c) Formatting tag Library (To enable I18n) ✓
d) Xml Tab library (To read and xml tags data)
f) JSTL functions (Given manipuate strings) ✓

I18n:: Internationalization

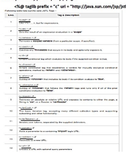
a) Core TagLibary (basic control flow opeartions)

=> tld file :: c.tld
=> finxed taglib uri ::
http://java.sun.com/jsp/jstl/core

=>recomanded prefix:: c

=>To import this jsp library in jsp page

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

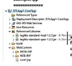


<c:set> ,<c:remove> ,<c:out>
==========================

<c:set> :: To create data having scope
<c:remove> :: To remove data from scope
<c:out> :: It is like expression tag..To display data on to the browser eithe template text or evaluated expression data.

=>we can use EL working with JSTL tags and attribues

ExampleApp
=============



also move these jar files to WEB-INF/lib folder using Deployment assembly option.

core1.jsp
========

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

<!-- To create variables having scope -->
 <c:set var="msg" value="hello" scope="request"/>

 <!-- To display data from scope -->
 value :: <c:out value="${msg}" /> <br>
value :: ${requestScope.msg} <br>
value :: ${msg} <br>
<!-- To remove variables from scopes -->
<c:remove var="msg" scope="request"/>

 <!-- To display data from scope -->
 value :: <c:out value="${msg}" /> <br>
value :: ${requestScope.msg} <br>
value :: ${msg} <br>
```

<c:if>
====
=>To apply condition
=> note: There is no <c:else> tag..

core2.jsp

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

<!-- To create variables having scope -->
 <c:set var="msg" value="hello" scope="request"/>

 <c:if test="${param.uname ne null }" >
      ${msg} Mr./Miss./Mrs. ${param.uname}
 </c:if>
```

To read and display request parameter uname

request url :: http://localhost:2525/JSTLApp1-CoreTags/core2.jsp?uname=raja

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

=> To bring the effect of "if ..else if .. else" or switch ...case

core3.jsp

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

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

</c:choose>
```

request lurl:: http://localhost:2525/JSTLApp1-CoreTags/core3.jsp?p=-10

<c:forEach>
==========
=> To peform "for" loop activities either as traditional "for loop" or as enhaced "for loop"

core4.jsp
========

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

<!-- traditional for loop -->
<b> tradtional for loop</b> <br>
<table border="1" align="left">
 <c:forEach var="i" begin="1" end="10" step="1" >
  <tr>
    <td>2 * ${i } = </td> <td> ${2*i } </td>
  </tr>
  </c:forEach>
</table>
<hr>
<br>
<b> enhanced for loop1</b> <br>
 <jsp:scriptlet>
   String nickNames[]=new String[]{"kanna","munna","chinna","pappu"};
   request.setAttribute("nNames", nickNames);
 </jsp:scriptlet>

  <c:forEach var="nick" items="${nNames}">
    ${nick} <br>
  </c:forEach>
```

---

core5.jsp

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

 <b> all request param names and values </b>
 <c:forEach var="p" items="${paramValues}">
     param name :: ${p.key} <br>
     param values ::
     <c:forEach var="pv" items="${p.value}">
         ${pv},
     </c:forEach>
 <br>
 </c:forEach>
```

displays req param name

gives all req param values as an array

=>displays all request param names and values

http://localhost:2525/JSTLApp1-CoreTags/core5.jsp?sno=101&sname=raja&sadd=hyd&sadd=vizag

core5.jsp
=======

```
  <hr>

  <b> all request header names and values </b> <br>
  <c:forEach var="h" items="${headerValues}">
      header name :: ${h.key} <br>
      header values ::
      <c:forEach var="hv" items="${h.value}">
          ${hv},
      </c:forEach>
  <br>
  </c:forEach>
```

gives all reuest heder and values.

<c:forTokens>
==========
spilts the given String message in to multiple tokens based on the given delimiter.

core6.jsp

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

<c:set var="msg" value="hello, how ar, e u"/>

<b>content after splittings</b>
<c:forTokens var="token" items="${msg}" delims="ab">
  ${token},
</c:forTokens>
```

Internally it may use "Annotations " Txmgt does occess the data from differn scops/
=>Interllay usse StringTokneizer to cimplete the task.

<c:url> ,<c::redirect>
================
=><c:url> defines the url having scope
<c:redire> perfoms sendRedirection activity through tags.

example App
============

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

  <c:url var="googleURL" value="https://www.google.com/search?q=ameerpet"/>

 <c:redirect url="${googleURL}" />
```

http://localhost:2525/JSTLApp1-CoreTags/core7.jsp?

<c:import>

=> To import/include one jsp page content in anothe jsp page

core8.jsp

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


<b> from the strt of core8.jsp</b> <br>

 <c:import url="core1.jsp"/>

<br> <b> from  theend of core8.jsp</b>
```

<c:catch>

=>given for exception handling and display exception related messages as non-technical guding messages..

core9.jsp

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

<c:catch var="e">
  <jsp:scriptlet>
     int a=Integer.parseInt("1x0");
     out.println("value ::"+a);
  </jsp:scriptlet>

</c:catch>
<c:if test="${!empty e }">
  Exception  ${e}
</c:if>
```

---

realtime utliazation of JSTL core Tag Library tags

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

<c:choose>
  <c:when  test="${!empty  booksInfo}">
   <h1 style="color:red;text-align:center">Books Belonging to  ::${param.category} </h1>
    <table border="1" bgcolor="cyan" align="center">
      <tr> <th>BookId </th> <th>BookName </th> <th>Category </th> <th>Author </th><th> price <th>Status</th> </th> </tr>
      <c:forEach var="book"  items="${booksInfo}">
       <tr>
           <td>${book.bookId} </td>
           <td>${book.bookName} </td>
           <td>${book.category} </td>
           <td>${book.author} </td>
            <td>${book.price} </td>
        <td>${book.status} </td>
        </tr>
      </c:forEach>
      </table>
  </c:when>
  <c:otherwise>
        <h1 style="color:red;text-align:center"> Books are not found </h1>
  </c:otherwise>
</c:choose>
```

JSTL SQL TagLibrary

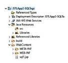
=>These tags are given to perform DB linteractions and for CURD operations.
=>fixed taglib url :: *http://java.sun.com/jsp/jstl/sql*
=> recomaded prefix :: sql
=>tld file name:: sql.tld

This tag libary tags are usefull only model1 architecture based web application devleopment.

S.No.	Tag & Description
1	<sql:setDataSource> Creates a simple DataSource suitable only for prototyping
2	<sql:query> Executes the SQL query defined in its body or through the sql attribute.
3	<sql:update> 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

```
===========
<%@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 select Query and getResultSet  -->
<sql:query dataSource="${ds}"  sql="SELECT PID,PNAME,PRICE,QTY,STATUS FROM PRODUCT"  var="rs"/>

<!-- Process the ResltSEt -->
<c:forEach  var="row"  items="${rs.rows}">
   ${row.pid }    ${row.pname }    ${row.price }    ${row.qty }    ${row.status } <br>
</c:forEach>
```



sql2.jsp

```
=======
<%@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 no-select Query  -->
<sql:update var="count" dataSource="${ds}"  sql="UPDATE PRODUCT SET PRICE=PRICE+? WHERE PRICE>?" >
  <sql:param>100</sql:param>
  <sql:param>8000</sql:param>
</sql:update>

<b> no.of record that are effeted :: ${count}</b>
```

Formatting Tag library

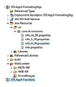
=>These are given to format date values, numbers ,currency to colect mesages from properties based on Local that is set.. Simply enabling Internationlization (I18n)

note:: The process of making app working for diffrent Locales is called enabling I18n
Locale means language+country

eg: en-US
fr-FR
fr-CA
de-DE
and etc..

fixed tag liburl :: *http://java.sun.com/jsp/jstl/ fmt*

recomanded prefix: fmt
tld file name: fmt.tld



S.No.	Tag & Description
1	<fmt:formatNumber> To render numerical value with specific precision or format.
2	<fmt:parseNumber> Parses the string representation of a number, currency, or percentage.
3	<fmt:formatDate> Formats a date and/or time using the supplied styles and pattern.
4	<fmt:parseDate> Parses the string representation of a date and/or time
5	<fmt:bundle> Loads a resource bundle to be used by its tag body.
6	<fmt:setLocale> Stores the given locale in the locale configuration variable.
7	<fmt:setBundle> Loads a resource bundle and stores it in the named scoped variable or the bundle configuration variable.
8	<fmt:timeZone> Specifies the time zone for any time formatting or parsing actions nested in its body.
9	<fmt:setTimeZone> Stores the given time zone in the time zone configuration variable
10	<fmt:message> Displays an internationalized message.
11	<fmt:requestEncoding> Sets the request character encoding

info.properties

```
------------------------
#Base file (English)
welcome.msg= Good Morning
```

info_de_hindi_.proprties

```
#Base file (hindi )
welcome.msg= \u0936\u0941\u092D \u092A\u094D\u0930\u092D\u093E\u0924
```

info_de_DE.properties

```
#Base file (germna)
welcome.msg= Guten Morgen
```

```
#Base file (french)
welcome.msg= Bonjour
```

all properties fiels in com.nt.commns pkg of "src" folder.

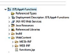
```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"  %>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt"  prefix="fmt" %>
<%@page contentType="text/html; charset=UTF-8" %>

 <fmt:setLocale value="fr_FR/>

 <fmt:formatNumber var="fno"  value="40006656066" type="number" />
 Formatted number :: ${fno}.
 <br>

 <fmt:formatNumber var="fno1"  value="50006656066" type="currency" />
 Formatted currency number :: ${fno1}.

<br>

<jsp:useBean id="dt"  class="java.util.Date"/>
<fmt:formatDate var="fdt"  value="${dt}"  type="both"  timeStyle="full" />
 Formatted date value :: ${fdt}.
<br>

<!-- To format vegetable -->
 <fmt:setBundle basename="com/nt/commons/info"/>
 <fmt:message  key="welcome.msg"/> <br>
```

---

4) JSTL functions
=>These are JSTL functions (internally EL functions) to perform string manipulation

S.No.	Function & Description
1	fn:contains() Tests if an input string contains the specified substring.
2	fn:containsIgnoreCase() Tests if an input string contains the specified substring in a case insensitive way.
3	fn:endsWith() Tests if an input string ends with the specified suffix.
4	fn:escapeXml() Escapes characters that can be interpreted as XML markup.
5	fn:indexOf() Returns the index withing a string of the first occurrence of a specified substring.
6	fn:join() Joins all elements of an array into a string.
7	fn:length() Returns the number of items in a collection, or the number of characters in a string.
8	fn:replace() Returns a string resulting from replacing in an input string all occurrences with a given string.
9	fn:split() Splits a string into an array of substrings.
10	fn:startsWith() Tests if an input string starts with the specified prefix.
11	fn:substring() Returns a subset of a string.
12	fn:substringAfter() Returns a subset of a string following a specific substring.
13	fn:substringBefore() Returns a subset of a string before a specific substring.
14	fn:toLowerCase() Converts all of the characters of a string to lower case.
15	fn:toUpperCase() Converts all of the characters of a string to upper case.
16	fn:trim() Removes white spaces from both ends of a string.

fixed taglib uri is *"http://java.sun.com/jsp/jstl/functions"*
recomaneded prefix is *"fn"*
tld fld tile functions.tld



functions.jsp

```
<%@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="hello how are u?"/>

Uppercase :: ${fn:toUpperCase(msg)} <br>
lowercase :: ${fn:toLowerCase(msg)} <br>
lenth :: ${fn:length(msg)} <br>
```

Most regulary used JSTL Libraries are :: Core , Formating , functions