Introduction to Jsp
==========

Nov 08 Jsp Intro- NOV8th

List of server side web techologies
---

servlet ---> from sun Ms /oracle corp (java based)
jsp ----> from sun Ms /oracle corp (java based)
php ---> from apache (non-java)
asp.net --- > from microsoft(non-java)
express Js ---> from strong loop and etc. (non-java)
asp ---> from microsoft (non-java)
cold fusion ---> from adobe (non-java)
and etc..

note:: These server side web technologies are given to develop dynamic web comps which are capable generating dynamic webpagpes.

When Sun Ms/oracle corp is already having servlet as server side technology why did they given as another server side web technology called jsp?

Ans) Before the release of servlet technology asp was popular server side technology becoz it is supporting tag based pprogramming like html programming.. After servlet release of servlet technology every one appriciated the servlet features but no one has moved servlet technology instead asp technology becoz to work with servlet technology strong java knowledge is requireed and more over servlet does not support tag based porgamming.. When sun Ms oberserved this . they have released their own tag based technology called jsp (java server pages) by inspiring from asp (active server pages) supporting tag based programming and giving all the features of servlet internally..

=>asp.net in extension of asp
=>Every jsp comp/prg internally converts into servlet comp/prg in order to complete the execution.. So in jsp programming we can enjoy tag based programming and also all features of servlet technology

jsp = tags + servlet features.

=>In the initial days jsp the s/w indurstry has used only jsp technolgy to develop java based web application.. later they started using both servlet and jsp technologies togather in java web application devlopment.

>Servlet for writing b.logic (calculations, analyzations, sorting ,filtering and etc..)
> jsp for writing presentation logic
(presenting data by applying styles)

Limitations of servlet technology
==========

a) Strong java knowledge is required to develop the servlet based web applications
b) Does not support tag based programming
c) Not suitable for Non-Java programmers
d) Placing html code (presentation logic) in servlet comp is quite complex and error prone
e) We need to mix up both presentation logic (html code) and business logic (java code) ..
we are forced to write html code in pw.println(-) method(we is wrting html code in java code)
f) Configuration of servlet comp i mapping with url patten is mandatory..
g) We can not place servlet comp in public area of the web application..
h) No built-in exception handling support
i) No spport for built-in objects.
=> request,response , servletConfig,ServletContext objs are the Container created objects ..and write some code to access those objects .. so we can not call them built-in objects .. they are just containerCreated object..
=>"this","super" are called implicit ref variables becoz we can use them in our app directly with out writing any addtional code.
=> main(-) methods's args[] , catch block Exception obj are not called implcit objects they are just called JVM created and supplied objects becoz we need to write some addtional code to get access those objects

=>implicit objects or reference variables are those things that are created underlying by JVM or container and use can use them directly in the app/code with out writing any code to access them
eg: "this","super" jsp 9 implicit objects.
=>Container/ JVM created objects means they created by JVM or container but we need to write some addtional code to access them
eg:: main(-) 's args[], catch block Exception object and etc..
eg:: In servlet programming req,res , servletConfig, Servletcontext objs
we need to override service(-,-)/doXxx(-,-) methods
ServletConfig cg=getServletConfig();
ServletCotnext sc=getSerlvetContext();

j) Learning cure(learning time) for Servlet technology is more..
k) for every modification done servlet comp's source code we need to recompile servlet comp and we need to reload the web application.

In java learning we have
->Languages :: java
->technologies :: servlet,jsp,jndi,jdbc, ...
->frameworks :: hibernate,spring ,spring boot, jsf and etc..

Features of jsp
==========

=> Supports tag based programming
=>Suitable for both java and non-java programmers
=>Strong java knowledge is not required
=>we can place java code (business logic) and html code(prensetation logic) seperately in jsp programming
=> placing html code jsp program is not error prone process.
=> Every jsp comp/prg will be converted into servlet comp .. So we can use all the fetures of servlet technology in jsp programming
=> gives 9 implicit objects to use in jsp programming
=> Learning cure/learning time of jsp technology is very less.
=>the modification done in jsp code will be reflected dymaically in the output with out seperate recompilaton and execution
=> we can place jsp comp either in public area or in private area..
when it is placed in private area .. its configuration with url pattern is mandatory..
whe it is placed in public area .. its configuration with url patter is optional
=>gives lots built-in jsp tags and allows to work with user-defined tags and third party supplied tags..
and etc..

note:: As on today .. the indurstry is preferring to use servlet,jsp technology togather either directly or though frameworks like spring mvc , spring boot mvc ,jsf and etc..

What is difference between servlet and jsp?

Nov 09 Jsp Intro2

servlet	jsp
(a) Does not support tag base programming	(a) supports tag based programming
(b) To work with servlet strong java knowledge is required	(b) Strong java knowledge is not required
(c) Not suitable for non-java programmers	(c) Suitable for both java and non-java programmers
(d) Its is programming java code based	(d) its programming html tags bassed
(e) Servelt comp must be placed only in private area of the web application	(e) can be placed either in private area or in public area of the web application
(f) makes programmer to mixup both presentation logic(html code) and b.logic(java code)	(f) makes progammer to seperate presentation logic(html code) from b.logic (java code)
(g) To excute servlet comp , we need Servlet container	(g) To execute jsp comps we need jsp container which internally uses servlet container
(h) Gives very few container created objects like req,res ServletConfig ,ServletCotext objs and No implicit objs	(h) Gives 9 implicit objects
(i) does not support implicit exception handling i.e we need to catch and handle exceptions explicitly..	(i) Supports implicit exception handling .. the jsp equalent servlet comp takes care of exception handling..
(j) Servlet comp code executes directly by Servletcontaienr	j) jsp comp code execution generates the equalent servlet comp and that comp executes..
(k) servlet life cycle methods are init(ServletConfig), service(req,res) , destroy()	(k) jsp life cycle methods are jspInit() , _jspInit() , _jspService(req,res) , jspDestroy()
(l) coding style is tradtional java coding	(l) coding style is free style programming..
(m) the modifications done in servlet comp will reflect only after recompilation of servlet comp and reloading of the web application	(m) The modifications done is jsp page will reflect dynamically with out any explicit recompilation and reloadings..

Jsp latest version :: 3.0

Jsp full form ::
Java server pages (old)
jakarta server pages (new)

In tomcat server
==========

To execute servlet comp ,it gives Servletcontainer (CATALINA)
To execute jsp comp , it gives Jsp container (JASPER) which internally uses the ServletContainer(CATALINA)
To convert jsp page/file/comp (.jsp file) into an equalent servlet comp we need "Jsp Page compiler" that will be given by Jsp container (jasper). In tomcat server the jsp page compiler name is "JspC".

=><Tomcat_home>\lib\servlet-api.jar file represents servlet api
=><Tomcat_home>\lib\jsp-api.jar file represents jsp api
=><Tomcat_home>\lib\catalina.jar file represents Serlvetcontainer / ServletEngine
=><Tomcat_home>\lib\jasper.jar file represents jsp container / Jsp Engine
|---->contains "jsp page compiler" whose name is "JspC".

request to jsp file ------(jspC) page compiler---------> .java (JES source code) ------javac-------> .class (JES bytecode)

Executed by
jsp container +Servletcontainer

response to browser

JES :: JSP equalent servlet

Structure jsp code in .jsp file
==========

<filename>.jsp

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

=>if .jsp file is placed in public area
its cfg in web.xml mapping with url pattern is optional

=>if .jsp file is placed in pirvate area
its cfg in web.xml mapping with url pattern is mandatory.

<% .... %> :: called scriptlet to place java code in jsp page

ordinary text+ html code ::: Template Text
This text or code always generate static content /fixed content..

```
<% .....
   .....
   ..... %>
```
scriptlet code is generally java code and always generate dyanmic content. (varying content)

=>.jsp file is called jsp comp /jsp file /jsp page (best)
=> Jsp pages generately dynamic web pages.. In that dynamic web pages the fixed content comes becoz template text (html code +ordinary text) and dynamic content comes becoz of java code.

first.jsp

```
<b> welcome to jsp </b>
 <br>
 <%  out.println("date and time::"+new java.util.Date()); %>
<br>
end of jsp
```

Generated dynamic web page of first.jsp

welcome to jsp
date and time:: 9/11/2021 ISI 4:48 pm
end of jsp

"out" is an implicit obj of jsp page

jsp page generates dynamic webpages .. The fixed content of those dynamic web pages will be generated by template text and varying content will be generated by the java code placed in jsp pages.

Developing and deploying first jsp pages based web application
==========

step1) create Deployment directory structure

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

step2) develop the first.jsp and web.xml file

first.jsp
---
same as above

web.xml

<web-app/>

=>There is no need of adding jsp-api.jar file to the CLASSPATH becoz we do not compile jsp pages from the command prompt.. the jsp pages page compilation takes place internally by Jsp page compiler of Jsp Container when we give request to jsp page (.jsp file)

step3) Start tomcat server

use <Tomcat_home>\bin\tomcat 9 .exe file

step4) Deploy the web application..

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

step5) Test the application.. by giving request to jsp page

http://localhost:2525/JspApp1/first.jsp

How many types of objects can be placed in jsp programming?

Two types of objects can be seen
a) Explicit objects
=>created by the programmers manually
eg: <%=new java.util.Date() %>
b) Implicit objects
=>Given by JES (Jsp Equalent Servlet) automatically

9 implicit objs of jsp
---
request,response
page,pageContext
config,application
out,exception,session

"config" in jsp = servletconfig obj is servlet
"application" in jsp = ServletCotnext obj in servlet
"out" of jsp is like PrintWriter of servlet

=>Imlicit ref variables in java app are :: "this","super"
=>implicit threads in java app are "main","gc"
=>implicit properties in java app are "length", "class"

Is jsp technology replacement of servlet technology?

Ans) No ... It compliments servlet technology i.e it internally uses Servlet technology .. when we request give jsp page .. an equalent servlet comp will be generated internally ..

Nov 10 Jsp life cycle

=>The process of transalating jsp page into an equalent servlet source code is called jsp page compilation
=>The process of generate java byte code from source code is called java compilation

.jsp --------jsp page compiler----------> .java -------java compiler----------> .class
(JspC) (servlet source code) (javac) (servlet byte code)

html	jsp
(a) html pages/files generate static web pages	(a) jsp pages/files generate dynamic web pages
(b) Given by w3c (wold wide web consortium)	(b) given Sun Ms (oracle corp)
(c) does not allow to place java code	(c) allows to place java code
(d) html tags and attributes are not case-sensitive	(d) jsp tags and attribues are case-sensitive
(e) html programming is not strictly programming	(e) jsp proramming is strictly typed programming
(f) Does not allow user-defined tags and third party tags .. allows only pre-defined tags	(f) allows user-defined jsp tags , third party jsp tags along with pre-defined jsp pags
(g) To execute HTML code , we need HTML Interpeter	(g) To execute jsp code , we need jsp container + servlet container
(h) Html is client side web technology	(h) jsp is server side technology
(i) Does not give any implicit objs	(i) Gives 9 implicit objects.

Servlet container executes servlet comp by raising 3 life cycle events and calling 3 life cycle methods
a) instantiation event ----------> public void init(ServletConfig cg)throws SE
b) Request processing event --------> public void service(ServletRequest req, ServletResponse res)throws SE,IOE
c) Destruction event ----------> public void destroy()

=>jsp container+ servlet container togather executes jsp page by generating jsp equalent servlet class and called calling life cycle methods for 3 life cycle events..

jsp life cycle methods are internally called through servlet life cycle methods

a) instantiation event ----------> i)public void jspInit() -> To place programmer choice initialization logic like creating jdbc con
(When JES class object is created) ii) public void _jspInit() --> To container specific initialization logic like activating engines like EL engine and etc..
(Both these methods are called init(-) method)
servlet life cycle method
"_" symbol in class name or in method name indicates they are container /JVM generated dynamic classes or methods..

EL :: Epression Language
(EL engine is rquired to execute EL)

b) request processing event ----------> i)public void _jspService(req,res) ---> Contains programmer supplied java code based req processing logics
(When JES class obj is ready to process the request)
(This method is internally called from service(req,res) method)
servlet life cycle method

c) Destruction Event
(When JES class obj is about to destroy)
i) public void _jspDestroy() -> contains the container specific unintialization logics like deactivating EL engine
ii) public void jspDestroy() -> can be used to place programmer's choice uninitialzation logics.
(These methods are internally called from destroy() method)
servlet life cycle method

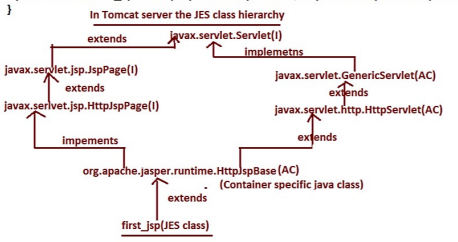
=>In Tomcat server the JES class of first.jsp belonging JspApp1 web application is first_jsp.java (JES soruce code) and first_jsp.class (JES compiled code) .. they come in <Tomcat_home>\work\Catalina\localhost\JspApp1\org\apache\jsp
app name
pkg name
|--->first_jsp.java (JES source code)
|--->first_jsp.class (JES by code)

=> Every JES class (Jsp equalent servlet comp) extends from container specific class (AC) which internally extends from javax.servlet.http.HttpServlet (AC)
=> The supper class of JES class is container specific class (In case of Tomcat server that is org.apache.jasper.runtime.HttpJspBase) contains servlet life cycle methods calling jsp life cycle methods intenrally

HttpJspBase.java
==========

```
package org.apache.jasper.runtime;

import java.io.IOException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.jsp.HttpJspPage;
import org.apache.jasper.compiler.Localizer;

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");
  }

  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;
}
```

In Tomcat server the JES class hierarchy



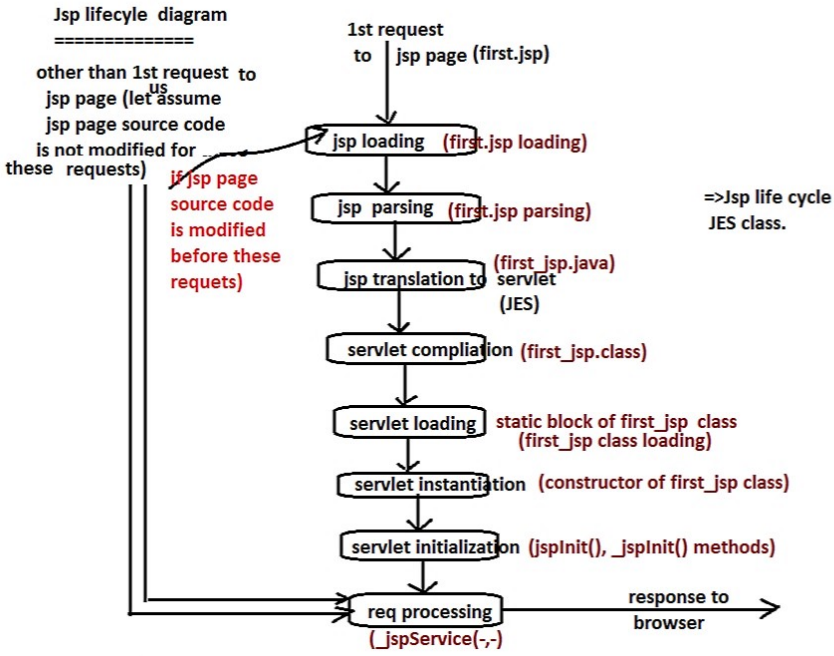
=>Instantiation event on JES class object ---> init(servletConfig) on JES class obj (not therer) ---> init(ServletConfig) of HttpJspBase executes--> that intenrally calls jspInit() and _jspInit() methods on JES class obj

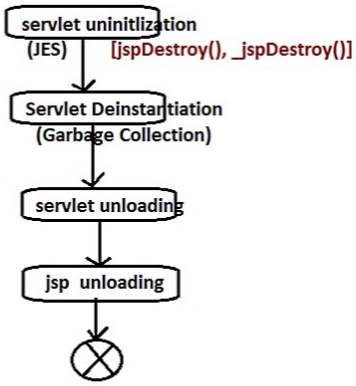
=>request event on JES class object ---> service(req,res) on JES class obj (not there) ---> service(req,res) of HttpJspBase executes--> that intenrally calls _jspService(-,-) method on JES class obj

=>Destruction event on JES class object ---> destroy() on JES class obj (not there) ---> destroy() of HttpJspBase class executes--> that intenrally calls _jspDestroy() , jspDestroy() methods on JES class obj

=>By default every JES class contains _jspInit() ,_jspService(req,res), _jspDestroy() methods .. the jspInit() and jspDestroy() methods will come only when they are placed in jsp page.

Nov 11 Jsp life cycle



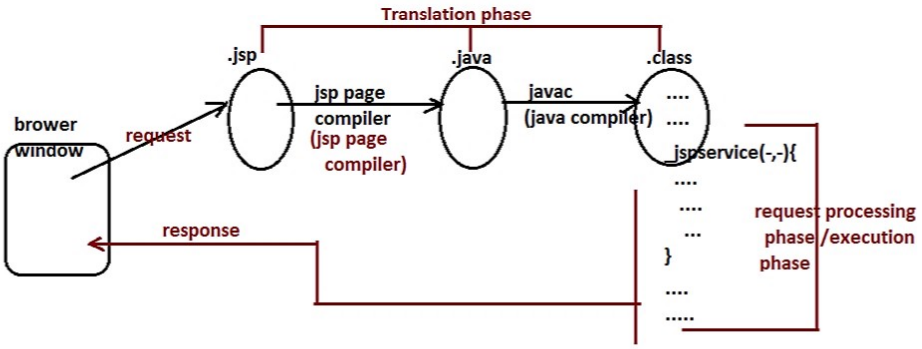


=>if the web application is stopped or reloaded or undeployed or if the server stopped or crashed.

Different phases of Jsp Execution
==========

(1) translation phase
[Here the jsp page will be translated into an equalent servlet comp source code and byte code]
.jsp --------> .java ----> .class

(2) execution phase /request processing phase
[Here jsp equalent servlet comp's byte code will be executed and the generated output goes browser as response]
.class execution (jsp life cycle method's execution through servlet life cycle methods)



=> The request given to jsp page directly participates in execution phase/request processing phase if the the source jsp page is not modified compare to previous requests and the byte code of JES class is avaliable otherwise the request given to jsp page first participates in translation phase and then participates in execution phase(request processingphase)

=>The Jsp container internally maintains the source code of jsp page for every request until next request comes .. to compare each request source code with the source code of previous request.. Based on the this the jsp page compiler takes the decision of participating in translationphase ...

=> In the translationphase of jsp , if jsp parsing error comes (error related to jsp tags and code structure) then the soruce code file (.java file) and byte code(.class file) of JES class will be deleted automatically.

=>In the translationphase of jsp , if JES class related compliation errors are coming (error related to java code placed scriitping tags like scriptlet)then the byte code(.class file) of JES class will be deleted automatically.

Nov 13 Jsp life cycle

what happens if modify source code JES class directly?

Ans) The modification related output does not reflect in the jsp page generated webpage

To recompile JES class (Jsp equalent servlet class) , we nedd to add the following jar files to the CLASSPATH or <java_home>\jre\lib\ext folder

=>servlet-api.jar
=>jsp-api.jar
=>catalina.jar
=>jasper.jar
=>el-api.jar
=>tomcat-api.jar
=>jasper-el.jar

from <Tomcat_home>\lib folder

Various elements/tags of jsp programming

1.Scripting tags (these are given to place java code in jsp pages)

a) scriptlet (<% ... %>)
b) Expression (<%= .. %>)
c) Declaration (<%! ... %>)

2.Directive tags (These are given to given directions to jsp page compiler to generate code in JES class)

a) page directive <%@page .... %>
b) include directive <%@include ... %>
c) taglib directive <%@taglib .... %>

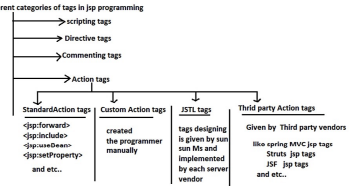
3. Jsp commenting tags

a) <%-- ..... --%> (To comment jsp tags based code)

4. Jsp Standard Action tags (These tags internally uses servlet api to complete the task dynamically at runtime)

<jsp:forward>
<jsp:include>
<jsp:useBean>
<jsp:setProperty>
<jsp:getProperty>
<jsp:plugin>
<jsp:fallback>
<jsp:param>
and etc..

Different categories of tags in jsp programming



Scripting tags

=>These are given to place java code /script code in jsp page
=> The java code placed jsp tags is called script code.. similarly the java script code placed in html tags is called script code.. The programming code surrounded by tags is called script code.

=> Jsp is giving 3 scriptint tags
a) scriptlet b) Expression c) Declaration

scriptlet

=> The code placed in scriptlet goes to _jspService(-,-) method of JES class as it is..
=>This tag is useful to place b.logic /request processing logic in jsp page.

standard syntax

```
<% .....
   ..... //java code
    ....
 %>
```

xml syntax

```
<jsp:scriptlet>
    .....
    ..... //java code
    .....
</jsp:scripet>
```

=>"jsp" of <jsp:scriptlet> is the prefix to different jsp built-in tags from other jsp tags like custom tags, third party tags , jstl tags.

The prefixes of jsp tags are very useful to differentiate two jsp tags when they have go same name.

<jsp:scriptlet> is different from <nit:scriptlet> tag.

=>Using scriptlet tag we can place java code to declare the variables and these variables become local variables to _jspService(-,-) method of JES clss.

first.jsp

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

in first_jsp.java (IN JES class)

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

 public void _jspService(req,res )throws SE,IOE{
    ....
    .... //logics to create implicit objects
    ....
   int a=10;
   int b=20;
    int c=a+b;
    out.println("result::"+c);

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

=> All the implicit objs of jsp are local variables to _jspService(-,-) and the code placed in scriptlet also goes to _jspService(-,-)method .. So we scriptlets we can use implicit objects

first.jsp

```
<%
  String browserName=request.getHeader("user-agent");
  out.println("current browser name ::"+browserName);
 %>
```

In _jspService(-,-) method of JES class

```
public void _jspService(request,response )throws SE,IOE{
  ....
  .... //logic to create implicit objs
   ....
   String browserName=request.getHeader("user-agent");
   out.println("current browser name ::"+browserName);

   ....
   ...
}
```

In scriptlet we can call methods.. but we can not define methods becoz java does not support nested method definitation

first.jsp

```
<%  public  int sum(int x,int y){
        return x+y;
}
                %>
<b> hello </b>
```

In _jspservice(-,-) method JES class

```
public void _jspService(req,res)throws SE,IOE{

  ....
  ....  // implicit objs creation
  .....
    public  int sum(int x,int y){
          return x+y;
    }

    out.write("\r\n");
    out.write("<b> hello </b>\r\n");
}
```

becomes nested method definitation of _jspService(-,-) method So JES compilation error will come

---

=>we can not place even method declaration in scriptlet becoz method declarations in the middle of the method definitations is not possible.

first.jsp

```
<%  public  int sum(int x,int y) ;  %>  //raises error
<b> hello </b>
```

In _jspservice(-,-) method JES class

```
public void _jspService(req,res)throws SE,IOE{

  ....
  ....  // implicit objs creation
  .....
    public  int sum(int x,int y);
                  //method declarations in the middle of method definitions are
  ....            not possible.
  ....
  ....
  }
```

Can we place class definitions inside the scriptlet?

Ans) yes ,possible becoz java supports local inner class
but that local inner class should not have any modifiers access like public and etc..

first.jsp

```
<%  class Test {
      int a;
  } %>

<b> hello </b>
```

In _jspservice(-,-) method JES class

```
public void _jspService(req,res)throws SE,IOE{

  ....
  ....  // implicit objs creation
  .....
  .....
   class Test{
     int a;
    }
  out.write("\r\n");
  out.write("<b>hello</b>");

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

java supports method local inner classes

Types of inner classes in java a) Inner calss b) static/nested inner cclass c) Local inner class d) Anonymous inner class.

Can we place Interface definitation in the scriptlet?

first.jsp

```
<%  interface Demo {
  } %>

<b> hello </b>
```

In _jspservice(-,-) method JES class

```
public void _jspService(req,res)throws SE,IOE{

  ....
  ....  // implicit objs creation
  .....
  .....
interface Demo{
  }

  out.write("\r\n");
  out.write("<b>hello</b>");

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

does not java support method level local inner interfaces..

=> class def inside another class def (allowed)
=>class def inside interface def (allowed)
=>interface def inside class def (allowed)
=>interface def inside interface def (allowed)
=>interface def inside method def (no)
=>class def inside method def (allowed)

Things that are not possible inside the script

=> method definitations
=> method declarations
=> method level inner interfaces
and etc..

Servlet filters and wraappers

=====Special Classes and Workshops by Mr.Nataraj=====
Link: zoom.us/j/97094106343
Pwd: 112233
timings :: nov14th :: 10:45 am
nov 20th :: 4:00pm
nov21st :: 10:45 am

Scripting tags — Jsp scriptlet

Nov 15 Scriptlet

Xml syntax of scriptlet

first.jsp

```
<jsp: scriptlet>
    int a=10;
  out.println("square value :"+(a*a) );
</jsp:scriptlet>
```

In JES clss

```
public  void _jspService(req,res)throws SE,IOE{
  ....
  .... //implicit objs
  ...
  int a=10;
  out.println("square value :"+(a*a));

}
```

"<" symbol problem with xml syntax based scriptlet

first.jsp

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

→ will not be taken as java relational/conditional operator .. It will be treated as the open tag of subtag under <jsp:scriptlet> tag.

error is ::
(org.apache.jasper.JasperException: /first.jsp (line: [2], column: [15]) Unterminated <jsp:scriptlet> tag)

Solution1:

Use standard syntax

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

note:: It is not solution for the problem
It is avoid the problem.

Solution2: (Best solution)

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

<![CDATA[..... .....]]>
body under <jsp:scriptlet>
indicate that the body content should be taken as ordinary text content and no xml meaning should be applied.

Xml tag body can be CDATA and PCDATA

CDATA :: Character Data (No xml meaning will be applied.. The content will be used as it is)

PCDATA :: Parsable Character Data (Xml meaning will be applied ..the content will used after parsing)

note:: In One jsp page we can both xml and standard syntax of same jsp tags or diffferent jsp tags.

note:: In Jsp page we can any place any jsp tag in any order for any number of times..

example:

```
<jsp:scriptlet>
  int a=10;
</jsp:scriptlet>
<%= int b=20;  %>

 <% out.println("sum is :"+(a+b)); %>
```

=>Nested scriptlets tags are not allowed i.e scriptlet tag Inside another scriptlet tag is not allowed

In first.jsp

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

Generates the error

Scripting tag- Expression tag

=>This tag is given to evaluate the given expression and to display the generated results/data on to the browser using out.print(-) method Internally..

=>Here Expression means any thing that gives data .. Expression could be variables, arithemetic operations, logical operations, instantiations ,method calls and etc..

=>Writing out.println(-) is scriptlet is like mixing html code(presentation logic) with java code (b.logic) (same of as java server coding) .. It is better to avoid it ..For that we can expression tag..

=> If expression tag is used effectively we can avoid out.println(-) or out.print(-) methods utilization perminently from our jsp pages.

=> The code placed in expression tag goes to _jspService(-,-) method of JES class as the argument values of out.print(-) methods.

standard syntax:

```
<%=  .... // expresion to evaluate
     %>
```

xml syntax:

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

=>we can use expresion tag to display variable values on to browser and to perform airthmetic and logical operations

first.jsp

```
<% int a=10;
   int b=20;
%>
 a value :: <%=a  %>  <br>
a>b value :: <%=a>b  %>  <br>
is a < b ? : <%=a<b %>
```

In JES class

```
public  void service(req,res)throws SE,IOE{
  .....  .....  //implicit  objs
  .....  .....
  out.write(" \r\n");
    out.write("  a value :: ");
    out.print(a );
    out.write("  <br>\r\n");
    out.write("  a>b value :: ");
    out.print(a>b );
    out.write("  <br>\r\n");
    out.write("  is a < b ? :  ");
    out.print(a<b );
    out.write("\r\n");
}
```

what is the difference between out.write(-) and out.print(-) methods?

Ans) =>out.write(-) can not display null values.. we attempt to display such values.. we will get NullPointerException

=>out.print(-) can display null values..

```
<% String s=null;  %>
<% out.print(s);   //displays  null
   out.write(s);   //throws NullPointerException
 %>
```

=>The JES class generally uses out.write(-) to write template text on to the browser becoz template text does not have java code and never gets null values to display.

=>The JES class generally uses out.print(-) to write java code results on to the browser becoz java code may have null values to display.

=>Implicit objects of jsp are visible expression tags becoz expression tags code go to _jspService(-,-) method and implicit objs are created as the local variables of _jspService(-,-) method.

first.jsp

```
<%=request.getHeader("user-agent") %>
```

In JES class

```
public void _jspService(-,-)throws SE,IOE{
   ....
   .... //implicit objs
   ...
   out.write(" browser name :: ");
     out.print(request.getHeader("user-agent") );
     out.write("\r\n");

}
```

We can expression tag to call the java methods and to display the generated results on to the browser

first.jsp

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

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

In JES class

```
public void _jspService(-,-)throws SE,IOE{
   .....
   .... //implicit objs
   ....

   String s="hello";
     out.write("\r\n");
     out.write("\r\n");
     out.write("   data :: ");
     out.print(s  );
     out.write("  <br>\r\n");
     out.write("   ");
     out.print(s);
     out.write("  length is :: ");
     out.print(s.length() );

   ...
}
```

=>Using expression tag we can call only those methods whose return type is other than void.

=>Using expression tag we can perform instantiation and we can display the initial data of the object on to the browser

first.jsp

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

In JES class

```
public void  _jspService(-,-)throws SE,IOE{
  ....
  ....  //Implicit objs
  ....
out.write("Data and time is :: ");
    out.print(new java.util.Date() );
  ...
  ...
}
```

Xml Syntax of expression tag

```
<jsp:scriptlet>
 int a=10;
 </jsp:scriptlet>

 value :: <jsp:expression>
             a
                    </jsp:expression>  <br>
square   value :: <jsp:expression>
                  a *a
                        </jsp:expression> <br>
cube    value :: <jsp:expression>
                  a *a*a
                        </jsp:expression> <br>
```

In JES class

```
p  void  _jspService(req,res)throws SE,IOE{
  ....
  ....  //implicit objs
  ....

  int a=10;

    out.write("\r\n");
    out.write("\r\n");
    out.write("   value :: ");
    out.print(
                a
                      );
    out.write("  <br>\r\n");
    out.write("square   value :: ");
    out.print(
                      a *a
                      );
    out.write("  <br>\r\n");
    out.write("cube   value :: ");
    out.print(
                      a *a*a
                      );
    out.write("  <br>\r\n");

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

While working with xml syntax based expression tag .. there is no solution for "<" symbol problem

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

 a less than b ?  <jsp:expression>
                      a<b
                  </jsp:expression>
```

//raise error
org.apache.jasper.JasperException: /first.jsp (line: [8], column: [36]) Unterminated <jsp:expression> tag

Using CDATA

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

 a less than b ?  <jsp:expression>
                    <![CDATA[
                      a<b
                    ]]>
                  </jsp:expression>
```

Again raises error

An error occurred at line: [11] in the jsp file: [/first.jsp]
The method print(boolean) in the type JspWriter is not applicable for the arguments ()

```
8:  a less than b ?  <jsp:expression>
9:                     <![CDATA[
10:                      a<b
11:                                ]]>
12:                  </jsp:expression>
```

Using Standard syntax

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

 a less than b ?  <%=  a<b  %>   //no error .. output goes to browser..
```

=>We can use one or expression tags in jsp page having only standard syntax or only xml syntax or mix of both.
=>Nested Expresion tags are not allowed..

Nov 17 Declaration

Declaration tag

=>This tag is given to place java code in JES before _jspService(-,-) method
=> This tag can be used to declare global variables (both static and instance) , to define java methods , to place jspInit() , jspDestroy() methods definitiations and also to place inner classes definitation.

standard sytax:

```
<%! .....
    .....  //global variables
    ..... // method definitations
    .....  //inner classes  definitation
%>
```

Xml syntax ::

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

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

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

In JES class

```
public class first_jsp extends ....{
     int a;  //gloal variable

 public void _jspService(-,-)throws SE,IOE{
     ....
    ....  //implicit objs
     ....
   out.write("value::");
   out.print(a);
   out.write("/r/n");
    out.write("square value:");
    out.print(a*a);
    ....
     ....
 }
}//class
```

---

if the variable names of scriptlet and declaration tags have got same name with different values then how can differentiate them scriptlet?

=>Scriptlet tag (<%... %>) tag variable becomes local variable to _jspService(-,-) method where as declaration tag (<%! ... %>) variable becomes global vairable in JES class, So differentiate them in scriptlet(in _jspService(-,-) method) we need to use "this" operator/keyword (best) or the implicit object "page"

=>The implicit obj "page" holds "this" value.
final java.lang.Object page = this;

first.jsp

```
<%! int a=10;  %>
<%  int a=20; %>

value(local) :: <%=a %> <br>
value(global) :: <%=this.a %> <br>
value(global) :: <%=((first_jsp)page).a %> <br>
```

(Here we are forced to do type casting with sub class (downcasting) but that code may raise ClassCastException becoz the JES class is not fixed.. and it will change server to server where we executing the App)

We can use declaration tag to place java method definitations

first.jsp

```
<%!  public int  sum(int x,int y){
        return x+y;
  } %>
result ::  <%=sum(10,20) %>
```

JES class

```
public class first_jsp extends  ....{
     '  public int  sum(int x,int y){
           return x+y;
     }
 public void  _jspService(-,-)throws SE,IOE{
      ....
      .... //implicit objs
      ...
      out.write("result::");
      out.print  (sum(10,20));
      ...
      ...
      }
}
```

=>We can generally place jspInit(), jspDestroy() and static blocks, constructor blokcs and etc.. with the support of declaration tags..

first.jsp

```
<%!
    static{
            System.out.println("first.jsp ::static block");
       }
    public first_jsp(){
            System.out.println("first.jsp ::0-param constructor");
       }

       public  void jspInit(){
            System.out.println("first.jsp ::jspInit() method");
       }

       public void jspDestroy(){
            System.out.println("first.jsp ::jspDestroy() method");
       }
       %>
  <%  System.out.println(" firsrt.jsp: _jspService(-,-) metod ");  %>

  <b>hello1 </b>
```

=>Give multiple requests to jsp page and observe the server console

---

=> Since java supports class level inner classes we can place inner classes definitation in the declaration tag

first.jsp

```
<%!  private static class  Test{
        int a;
   } %>

<b>hello </b>
```

JES class (Jsp equalent servlet comp)

```
public class first_jsp extends  ....{
  private static class  Test{
        int a;
   }                    //Nested inner class or
                        static inner class.
  ...
  public void _jspService(-,-)throws SE,IOE{
     ....
     .... //implicit objs
     ....
     ,,,
   }
}
```

With the xml syntax of declration tag we have problem with "<" symbol and we can overcome that problem using <![CDATA[...]]> concept

first.jsp (problem)

```
<jsp:declaration>
  public String findBig(float a,float b){
     if(a<b)
           return "big value ::"+b;
        else if(b<a)
           return "big value :"+a;
        else
           return "both are equal";
   }
 </jsp:declaration>

  result ::  <%=findBig(10,20) %>
                (or)
  result ::  <%jsp:expression> findBig(20,20);  %>
```

we have taken "<" symbol as java conditional operator .. but it will be treated xml sub tag's openining

raises error

org.apache.jasper.JasperException: /first.jsp (line: [2], column: [18]) Unterminated [<jsp:declaration>] tag

Solution through <![CDATA ....]]> body

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

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

Nov 18 Declaration

Can we enable <load-on-startup> on jsp? What is the advantage of it?

Ans) yes , possible The following activities of jsp life cycle takes place either during server startup or during the deployment of web application the moment we enable the load-on-statup on jsp page

a)jsp loading
b)jsp parsing
c) jsp translation to JES
d) JES compilation
e) JES Loading
f) JES Instantiation
g) JES Initialization

enabling <load-on-startup> on jsp page cfg is called enabling pre-translation and pre-instantiation on jsp page.

=>So the first request given to jsp page (with out modifying source code) directly parcipates in request processing

---

=> if jsp page is placed in private area of the web application then its cfg in web.xml file mapping with url pattern is mandatory ... Though jsp page is placed in public area if u want to enable <load-on-startup> on then jsp page cfg with url pattern in web.xml file is mandatory.

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

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>/firsturl</url-pattern>
  </servlet-mapping>

</web-app>
```

first.jsp
```
<%!
  static {
        System.out.println("first.jsp :static block ");
  }
    public  first_jsp(){
        System.out.println("first.jsp :0-param constructor");
    }

    public void jspInit(){
        System.out.println("first.jsp :jspInit()");
    }

    public void jspDestroy(){
        System.out.println("first.jsp :jspDestroy()");
    }
%>

<%
   System.out.println("first.jsp _jspService(-,-) method");
%>
  Date and time :: <%=new java.util.Date() %>
```

=>After cfg jsp page with url pattern in web.xml file we not should request same jsp page with file name..we should always request that jsp page with url pattern.

http://localhost:2020/JspApp1/firsturl (correct)
=>Directly particpates in request processing (req given to first.jsp page becoz <load-on-startup> is enabled)
http://localhost:2020/JspApp1/first.jps (not correct)
=> The <load-on-startup> enabled on the jsp page will not be applied.

=> if we cfg jsp page in web.xml file it automatically enables with <load-on-startup> activity so there is no need of placing <load-on-startup> tag seperately..

```
<web-app>
  <servlet>
    <servlet-name>f</servlet-name>
        <jsp-file>/first.jsp </jsp-file>
  </servlet>
  <servlet-mapping>
       <servlet-name>f</servlet-name>
            <url-pattern>/firsturl</url-pattern>
  </servlet-mapping>
</web-app>
```

Q) if multiple jsp pages are cfg with url patterns in web.xml with out <load-on-startup> priority value then what happens?

Ans) Server uses its own algorithm to decirde order of performing pre-transalation and pre-instantiation

Q) can we override _jspInit() or _jspDestroy() or _jspService(-,-) methods in our jsp page by using declaration tag?

Ans) Not possible becoz our _jspXxx() methods becom duplicate methods for already generated _jspXxx() methods of JES class.. and java supports method overloading .. but doesnot support methods duplication.

we

Q) Can override directly servlet life cycle methods in the declaration tags of jsp page?

Ans) not possible ... becoz all servlet life cycle methods are given as the final in the super class of JES class (HttpJspBase in case of Tomcat server) .. if we place servlet life cycle methods definitiation in jsp page .. they become overriden methods of JES class.. But final of super class (HttpJspBase) can not overridden in sub classes.

proof:
======

Source code HttpJspBase class
==========

```
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");
  }

  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;
}
```

---

note: =>In One jsp page we can place any no.of declarationt tags having both xml and standard syntaxes..
=>Nested declaration tags are not allowed..
=> 3 scripting tags (scriptlet, expression and declaration) allows us place java code in jsp age directly..

Q) Can use jsp implicit objects in the declatrion tags of jsp page?

Ans) Not possible to access to use directly .. becoz the code placed in declaration tag goes outside of _jspService(-,-) method and the jsp implicit objs are the local variables of _jspService(-,-) method?

first.jsp
```
<%! public void jspInit(ServletConfig cg){
      System.out.println(" first1.jsp ::init(-) method");
          out.println("<b>hello </b>");
} %>
```
Gives error

An error occurred at line: [5] in the jsp file: [/first1.jsp]
out cannot be resolved

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

Nov 19 Eclipse IDE - All Scripting Tavs

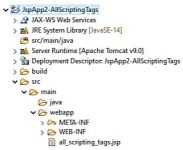
Developing Eclipse IDE web application having jsp page covering all the 3 scripting tags

step1) make sure the Tomcat server is cfg with eclipse IDE
step2) create Dyanamic web project in Eclipse IDE having JspApp2-AllScriptingTags
step3) Add jsp page to the src\main\webapp folder of the Project

all_scripting_tags.jsp

```
<%! public String generateWishMessage(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 wish Message
       if(hour<12)
            return "Good Morning::"+user;
       else if(hour<16)
            return "Good AfterNoon ::"+user;
       else if(hour<20)
            return "Good Evening ::"+user;
       else
            return "Good Night ::"+user;
  }
  %>

  <h1 style="color:red;text-align:center"> Welcome to Java server Pages </h1>
  <br>                                          expression tag
  <h1 style="color:green"> Date and time is :: <%=new java.util.Date() %> </h1>
  <% String name="raja"; %>  scriptlet
  <br> <b> The wish message is :: <%=generateWishMessage(name) %></b>
                                  expression tag
```

Declaratiob tag



note:: Mapping jsp page with url pattern in web.xml is required only when
a) if u want to enable <load-on-startup> on jsp page
b) if jsp page is in private area
c) a +b

step4) Run the application

Right click on the Project ---> run as ---> Run on server ----> .....
request url :: http://localhost:3031/JspApp2-AllScriptingTags/all_scripting_tags.jsp

In JES class for the jsp page of Eclipse IDE comes in Eclipse Tomcat server folders

G:\Worskpaces\advjava\NTAJ415\.metadata\.plugins\org.eclipse.wst.server.core
(work space folder)
\tmp0\work\Catalina\localhost\JspApp2-AllScriptingTags\org\apache\jsp
(web application name) pkg name

Commnents in jsp pages

=>jsp page allows 3 types of comments

a) scriptting /java comments /code comments
=>To comment java code of jsp page in scripting tags
// -->single commnets
/* .... */ --> multiline comments
/** .... */ ---> Documentaton comments

b) Jsp Comments / Hidden Comments
=>To comment jsp tags code in jsp page
=> These commnents are not visible in any phase of jsp page execution except jsp's source page ,
So they are called hidden comments.
<%-- .....
.... --%>

c) Html Comments / output comments /Template text commnents
=>These commnents are given to commnet html code or template text (plain text +html code)
of the jsp page ..
=>These commnets go to the browser along the with output/response to browser ..So these comments
are called output comments.
<!-- .....
..... -->

all_scripting_tags.jsp

```
<%! public String generateWishMessage(String user){
      //int a=10;                                  *
    //get System date and time                     *  -->java commnets/
     java.time.LocalDateTime ldt=java.time.LocalDateTime.now();   scripting comments
    // get current hour of the day                 *
      int hour=ldt.getHour();
     //generate wish Message                       *
     if(hour<12)
          return "Good Morning::"+user;
     else if(hour<16)
          return "Good AfterNoon ::"+user;
     else if(hour<20)
          return "Good Evening ::"+user;
     else
          return "Good Night ::"+user;
}
%>
                                          ** html comments
<!-- <h1 style="color:red;text-align:center"> Welcome to Java server Pages </h1> -->
<br>
<h1 style="color:green"> Date and time is :: <%=new java.util.Date() %> </h1>
<% String name="raja"; %>
<br> <b> The wish message is :: <%-- <%=generateWishMessage(name) %> --%></b>
                                  *** jsp comments
```

=>ctrl+shift+c toggle key for commenting single line
=> if the line contains only java code then it raises only java comments
=> if the line contains only html code then it raises only html comments/template text comments
=> if the line contains only jsp code then it raises only jsp comments
=> if the line contains only jsp +html code then it raises only jsp comments

Comment type	Visibility: In jsp page source code	In JES class source code	JES class byte code	html code that goes to browser	response on browser
Hiddent comments/ jsp commnet <%-- ..... --%>	yes	no	no	no	no
java comments/ scripting commnets	yes	yes	no	no	no
html comments/ output comments	yes	yes	yes	yes	no

Can i commnent jsp tags code using html comments?
Ans) yes , we can do but not recomanded

Can i comment hmlt tags code using jsp comments?
Ans) yes , we can do but not recomanded to do

Can i comment java code with jsp or html comments?
Ans) not possible

=> java comments will be recongized by java compiler "javac" ,So the commented code will not recorded into JES .class file
=> jsp comments will be recongized by jsp parser ,So the commented code will not recorded into JES source code (.java file)
=> html comments will be recongized by html interpeter ,So the commented code will not be displayed on to the browser as content of the webpage.

Scopes in jsp programming

=> page scope :: specific to each jsp page
=> request scope :: specific to each request i.e in all the web comps that are processing a request
=> session scope :: specific to each browser s/w of a client machine ..
=> application scope :: specific to a web application

Nov 20 Servelt Filters

Developing ServletFiter that allows requests to all the web comps of the web application between 9am to 5pm

step1) Keey any web application ready
step2) add Filter Comp to web application having the url pattern "/*" as shown below

TimeCheckFilter.java
==========

```
package com.nt.filter;

import java.io.IOException;
import java.io.PrintWriter;
import java.time.LocalDateTime;

import javax.servlet.FilterChain;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
                                                    (request filter)
@WebFilter("/*")  //annotation are alternate for  xml cfgs
public class TimeCheckFilter extends HttpFilter {

    @Override
    public void doFilter(HttpServletRequest req, HttpServletResponse res, FilterChain chain)
                throws IOException, ServletException {
        //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set response content type
        res.setContentType("text/html");
        // get  System date and time
        LocalDateTime ldt=LocalDateTime.now();
        //get current hour of the day
        int hour=ldt.getHour();
        if(hour>=9 &&  hour<=17)
            chain.doFilter(req, res);  //executes next filter or destination web comp
        else {
            pw.println("<h1 style='color:red;text-align:center'> Request must be given between 9 am  to 5 pm </h>");
            return ;
        }
        //close the stream
        pw.close();
    }


}
```

=> The websiste that allows stock market trading
allows to pefrom trading activities between 9am to 5pm

step3)Run the application...

=>Developing ServletFilter comp that allows requests from certain browser s/w (chrome) only?

step1) Keep the web application ready

step2) Develop the ServletFitler comp having url pattern "/*"

```
@WebFilter("/*")          (request filter)
public class BrowserCheckFilter extends HttpFilter {

    @Override
    public void doFilter(HttpServletRequest req, HttpServletResponse res, FilterChain chain)
                throws IOException, ServletException {
        //get PrintWriter object
        PrintWriter pw=res.getWriter();
        //set content type
         res.setContentType("text/html");
         // get browser name
         String browser=req.getHeader("user-agent");
         System.out.println(browser);
         if(browser.contains("Chrome")) {
             chain.doFilter(req, res); // executes next filter or dest web comp
         }
         else {
             pw.println("<h1 style='color:red;text-align:center'>Request must be given from Chrome browser </h1>");
             return;
         }
         //close stream
         pw.close();
        }//doFiter
}//class
```

=>if multiple filters are cfg on servlet comp using xml cfgs then in which order the filter comps are cfg in web.xml file the request will be trapped and the response will be trapped in the reverse order..

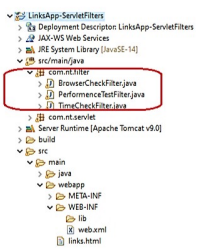
=> if the same multiple filters are cfg using annotations ...then based on the alphabetic order of Filter class names the request trapping and response trapping order will be decided

---

=> use Sevrlet Filter placing optional pre-request processing and post-response processing logics which can be enabled or disabled on demand basis with out effecting other logics in other web comps of the web application.

=>Develop Filter comp that evaluates the performence of each web comp by getting its request processing time
[This filter comp must be developed as request-response filter]

```
//PerformenceTestFilter.java   (request -response filter)
package com.nt.filter;

import java.io.IOException;

import javax.servlet.FilterChain;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebFilter("/*")
public class PerformenceTestFilter extends HttpFilter {

    @Override
    public void doFilter(HttpServletRequest req, HttpServletResponse res, FilterChain chain)
                throws IOException, ServletException {
        long startTime=0L  , endTime=0L;
        startTime=System.currentTimeMillis(); //request trapping time
        chain.doFilter(req, res);  //goes to next filter or  dest comp
        endTime=System.currentTimeMillis();  //response trapping time

        System.out.println(req.getRequestURI()+" has taken "+(endTime-startTime)+" ms to process the request");

    }
}
```



ServletWrappers

Nov 21 ServletWrappers

=>A Wrapper is layer around existing item .. So we can use existing item facilities and wrapper facilities through wrapper

=>Wrapper data types are object versions of simple data types
Integer wrapper class obj

=>ServletWrappers are given to develop Custom request obj and custom response obj as wrappers on the SErvletcontainer created request , response objs.

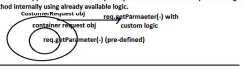
=> We can use custom request object and custom response object to cutomize the funcationalaties of certain methods..and using existing logics for other methods.

Need of customer request object

email Id nit@gmail.com @gmail.com
=>If write @gmail.com in the mail id it should take it as it is
If we do not write it should @gmail.com automatically

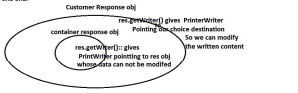
email Id nit @gmail.com
it takes email id by appending @gmail.com (overall it is nit@gmail.com)

The ready made req.getParameter(-) does not have thid capability...So we need to develeop custom logic for req.getParameter(-) method internally using already available logic.

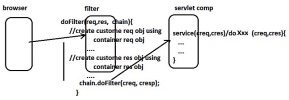


Need of CustomReponse object

=> If we want to add signature(additional contetnt) to every response .. we need to write our choice logics for res.getWriter() method where we create our choice stream object writing out choice destination , So that we can take that written content to modify it or to add sinature contenet at end end.



=>After creating these Custom request and Custom response objects to by using container created request ,response objs To pass to servlet comp we need to take support of Filter by calling chain.doFilter(-,-) with cusotm req , custom res objs



Developing Custom Request and CustomResponse classes becomes easy If we maken them extending Request ,response Wrapper classes beccu they come with container created req,res objs ..So what ever methods we not redefining they will executed directly on container created req,res object.

problem1:

```
public class CustomReequest implemetns HttpSErvletRequest{
   ....
   ... //need to implements all the methods (burden to the programmers)
   ...
   ..
}

public class CustomResponse implemetns HttpSerVletResponse{
   ....
   ... //need to implements all the methods (burden to the programmers)
   ...
   ..
}
```

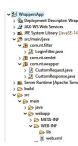
Different ReadymadeWrapper classes

ServletRequestWrapper ServletResponseWrapper HttpServletRequestWrapper HttpServletResponseWrapper	Contains defintiation for all methods internally calling same on container created req , res objs.

Solution1:

```
public class CustomRequest extends HttpServletRequestWrapper{

   ....
   ... //override only those in which u r intrested in (bo burden on the programmers)
   ..
}

public class CustomResponse extends HttpServletResponseWrapper{

   ....
   ... //override only those in which u r intrested in (bo burden on the programmers)
   ..
}
```

Example App



login.html

```
<h1 style="color:red;text-align:center"> Login Page</h1>

<form action="loginurl" method="POST">
  <table align="center" bgcolor="cyan">
    <tr>
       <td> email id:: </td>
       <td> <input type="text" name="email">@gmail.com</td>
    </tr>
    <tr>
       <td> password :: </td>
       <td> <input type="text" name="pwd"></td>
    </tr>
    <tr>
       <td> <input type="submit" value="Login"> </td>
       <td> <input type="reset" value="cancel"></td>
    </tr>

  </table>
</form>
```

LoignServlet.java

```
package com.nt.servlet;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/loginurl")
public class LoginServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {

        System.out.println("LoginServlet.doGet(-,-)");
        System.out.println("req obj type::"+req.getClass()+"  "+res.getClass());
    //get PrintWriter
        PrintWriter pw=res.getWriter();
        //set content type
        res.setContentType("text/html");
        //read form data
        String mail=req.getParameter("email");
        String pwd=req.getParameter("pwd");
        System.out.println(mail +"     "+pwd);
        //write b.logic
        if(mail.equalsIgnoreCase("raja@gmail.com") && pwd.equalsIgnoreCase("rani")) {
            pw.println("<h1 style='color:green;text-align:center'> Valid credentials </h1>");
        }
        else {
            pw.println("<h1 style='color:red;text-align:center'> inValid credentials </h1>");
        }
        //add home hyperlink
        pw.println("<br><a href='login.html'>home </a>");
        //close stream
        pw.close();

    }

    @Override
    public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        System.out.println("LoginServlet.doPost(-,-)");
      doGet(req, res);
    }
}
```

CustomerRequest.java

```
package com.nt.wrapper;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletRequestWrapper;

public class CustomRequest extends HttpServletRequestWrapper {
  private HttpServletRequest request;
      public CustomRequest(HttpServletRequest request) {
            super(request);
            System.out.println("CustomRequest.1-param constructor");
            this.request=request;
      }

      @Override
            public String getParameter(String name) {
            System.out.println("CustomRequest.getParameter(-)");
              String paramVal=request.getParameter(name);
               if(name.equals("email")) {
                     if(!paramVal.endsWith("@gmail.com"))
                           paramVal=paramVal+"@gmail.com";
               }
              return paramVal;
            }
}
```

CustomResponse.java

```
package com.nt.wrapper;

import java.io.CharArrayWriter;
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpServletResponseWrapper;

public class CustomResponse extends HttpServletResponseWrapper {
  private  HttpServletResponse response;
  private CharArrayWriter  writer;
  public CustomResponse(HttpServletResponse response) {
            super(response);
            System.out.println("CustomResponse:: 1-param  constructor");
            this.response=response;
            writer=new CharArrayWriter(); // internally creates buffer and takes it as destination

      }

    @Override
        public PrintWriter getWriter() throws IOException {
          System.out.println("CustomResponse.getWriter()");
            PrintWriter pw=new PrintWriter(writer);  // Here indirectly  PrintWriter stream is taking buffer as destination
            return pw;
        }

    @Override
        public String toString() {
          System.out.println("CustomResponse.toString()");
              return writer.toString();  //content collected from buffer is returned custom response obj content
        }
    }
```

//LoginFilter.java

```
package com.nt.filter;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.FilterChain;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.nt.wrapper.CustomRequest;
import com.nt.wrapper.CustomResponse;

@WebFilter("/loginurl")
public class LoginFilter extends HttpFilter {

@Override
public void doFilter(HttpServletRequest req, HttpServletResponse res, FilterChain chain)
            throws IOException, ServletException {
        //create CustomRequest object
        CustomRequest  creq=new CustomRequest(req);
        //create CustomResponse obj
        CustomResponse  cres=new CustomResponse(res);

        chain.doFilter(creq, cres);  //calls next filter or destination comp
        // collect content from custom response obj
        String output=cres.toString();
        //modify the content
        output=output+"<br> -to Naresh IT</to>";
        //get PrinterWriter  pointing to container created response obj
        PrintWriter pw=res.getWriter();
        pw.println(output); //writes container created response obj --> container --> server --> browser window

        //close stream
        pw.close();

    }

}
```

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>WrappersApp</display-name>
  <welcome-file-list>
    <welcome-file>login.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Nov 22 Eclipse IDE - implicit objs in jsp page and html to jsp to Db interaction

Scopes in jsp programming

=> page scope :: specific to each jsp page
=> request scope :: specific to each request i.e in all the web comps that are processing a request
=> session scope :: specific to each browser s/w of a client machine ..
=> application scope :: specific to a web application

Implicit objects in jsp page

Techincally speaking , it is better to call implicit objs as the implicit ref variables becoz objects generally do have names.. they are reffered with reference variable names.

```
Test t=new Test();
  "t" referent type :"Test"
  "t" object type : "Test"

Object o=new Test();
  "o" reference type :: "java.lang.Object"
  "o" object type :: "Test"
```

For all implicit objects /reference variables of jsp the reference type is the class name /interface name/abstract class name given by servlet api or jsp api .. where object type is class name given by underlying servlet cintainer or jsp container .. The object type class name is varying name based on the containenr or server we use where as reference type fixed becoz it is give by the common servlet api or jsp api.

note:: Object type is always class name becoz we can create object only for classes
note: reference type can be a class or interface or abstract class name

implicit obj/ref variable	reference type	scope (class= concrete class)
request	javax.servlet.http.HttpServletRequest(I)	request scope
response	javax.servlet.http.HttpServletResponse(I)	request /response scope
out	javax.servlet.jsp.JspWriter (AC)	page scope
config	javax.servlet.ServletConfig(I)	page scope
application	javax.servlet.ServletContext (I)	application scope
page	java.lang.Object(c)	page scope
pageContext	javax.servlet.jsp.PageContext(AC)	page scope
session	javax.servlet.HttpSession (I)	session scope
exception	java.lang.Throwable (c)	page scope

test.jsp

```
<h1 style='color:red;text-align:center'> Jsp implicit objs information</h1>

out object class name <%=out.getClass() %> <br>
                      org.apache.jasper.runtime.JspWriterImpl

request object class name <%=request.getClass() %> <br>
                      org.apache.catalina.connector.RequestFacade

response object class name <%= response.getClass() %> <br>
                      org.apache.catalina.connector.ResponseFacade

application object class name <%=application.getClass() %> <br>
                      org.apache.catalina.core.ApplicationContextFacade

config object class name <%=config.getClass() %> <br>
                      org.apache.catalina.core.StandardWrapperFacade

session obj class name <%=session.getClass() %> <br>
                      org.apache.catalina.session.StandardSessionFacade

page object class name <%=page.getClass() %> <br>
                      org.apache.jsp.test_jsp (JES class name)

pageContext object class name <%=pageContext.getClass() %> <br>
                      org.apache.jasper.runtime.PageContextImpl
```

note:: The implicit exception does not come every jsp page automatically.. It will come only in the jsp page that is cfg as error jsp page

config obj in jsp = ServletConfig obj in servlet
application obj in jsp = ServletContext obj in sevlet

note:: All implicit jsp objs are the objects of underlying server/container supplied java classes implementing the servlet api/jsp api interfaces or extending from servlet api/jsp classes/abstract classes.

For example "out" is not object of javax.servlet.jsp.JspWriter(AC) .. It is the object of container supplied java class extending from the JspWriter(AC) .. In Tomcat server that class name is org.apache.jasper.runtime.JspWriterImpl

---

wrong assumption:: we need not to perform exception handling jsp page becoz the JES class will take of the exception handling
fact :: The above statement is correct only for the code that goes _jspService(-,-) method .. For the code placed in declaration tags we need to take care exception handling on our own.

problem: in first.jsp page

```
<%! public void jspinit(){
      Class.forName("oracle.jdbc.driver.OracleDriver");
      System.out.println("driver loaded");
    } %>
```

gives error saying unhandled ClassNotFoundException

solution::

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

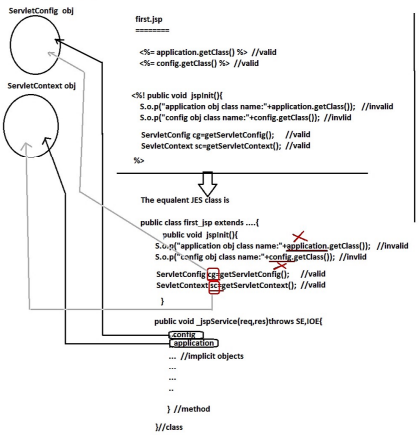
catches and handles the exception .. So error will be raised.

Wrong assumption :: we can not use implicit objs in the declaration tags code

fact :: yes .. we can not use container created object in declaration code through implicit object objects/ ref variables of jsp becoz they are local variables to _jspService(-,-) method and the decaration tag code goes out side of _jspService(-,-) method. But we can create our choice direct reference to container created objects to access them in our choice places of jsp programming

=>ServletConfig object is visible in all parts of jsp page and JES class but if u try to access through "config" implicit object/ref variable it can be accessed only in _jspService(-,-) method code

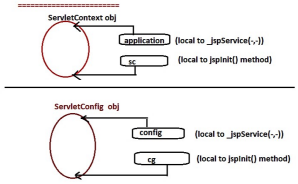
=>ServletContxt object is visible in all parts of jsp pages/JES classes,servlet comps but if u try to access through "application" implicit object/ref variable it can be accessed only in _jspService(-,-) method code



note: If u can not access certtain Container created objects in our jsp page using jsp implicit objs/ref varoales then try to create direct ref variable pointing those objects in the place u wanted.

=one object can multiple reference variable belong to same scope or different scopes

In the object above example



first.jsp

```
<%!
  public void jspinit(){
  //  System.out.println("application obj class name::"+application.getClass());
    // System.out.println("config obj class name::"+config.getClass());
      ServletConfig cg=getServletConfig();
      ServletContext sc=getServletContext();
        System.out.println("application obj class name::"+sc.getClass()+"  "+sc.hashCode());
      System.out.println("config obj class name::"+cg.getClass()+"  "+cg.hashCode());
  }
%>
 <b> driver loaded</b>
application object class name <%=application.getClass() %>  <%=application.hashCode() %> <br>
config object class name <%=config.getClass() %>  <%=config.hashCode() %> <br>
```

gives errors

valid

valid

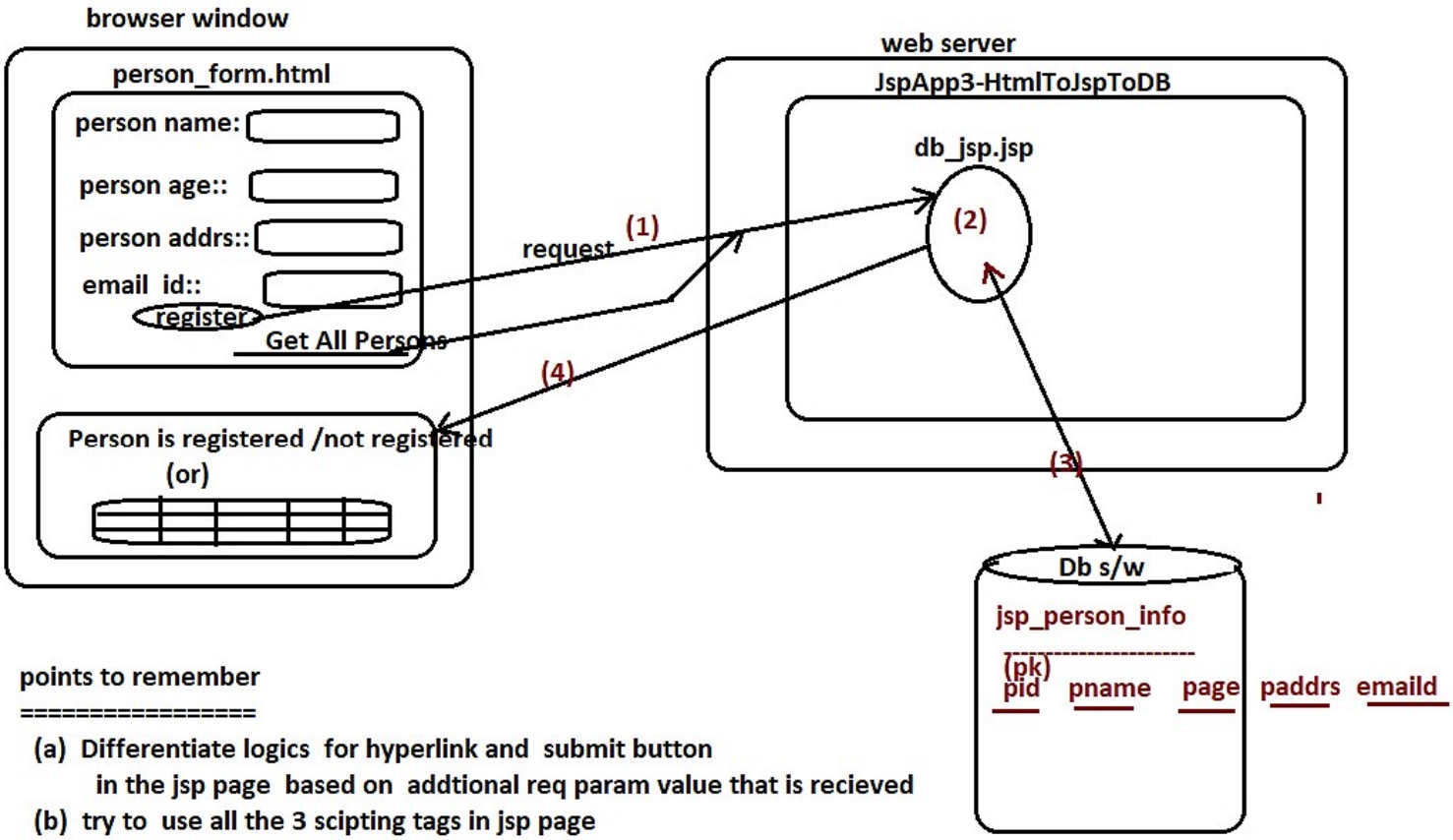
=>By Accessing ServletConfig , SErlvetcontext object in the declaration tag code we can read jsp init parameter values and context parameter values ....

Html to jsp to DB s/w interaction
==========

Nov 24 Eclipse IDE - html to jsp to DB communication

=>For html to jsp communication , we need to place jsp file name or url pattern as "href" attribute value of <a> or "action" attribute value of <form> tag

=> For jsp to to Db s/w we need to place jdbc code in the jsp page/comp/file



points to remember
==========

(a) Differentiate logics for hyperlink and submit button
in the jsp page based on addtional req param value that is recieved
(b) try to use all the 3 scipting tags in jsp page
(c) try to use all the 3 jsp life cycle method
(d) gather jdbc properites from web.xml file as jsp init param values
(e) Try to avoid out.prntln()/print() and out.write()/writeln() methods in the jsp coding .. prefer working with expression tags..

```
CREATE TABLE "SYSTEM"."JSP_PERSON_INFO"
  (    "PID" NUMBER(15,0) NOT NULL ENABLE,
       "PNAME" VARCHAR2(20 BYTE),
       "PAGE" NUMBER(3,0),
       "EMAILID" VARCHAR2(20 BYTE),
       "PADD" VARCHAR2(20 BYTE),
        CONSTRAINT "JSP_PERRSON_INFO_PK" PRIMARY KEY ("PID"));
```

PID1_SEQ (sequence in oralce)
-==========

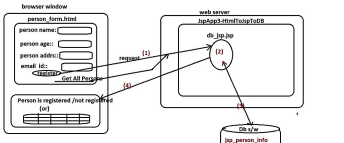
```
CREATE SEQUENCE "SYSTEM"."PID1_SEQ" MINVALUE 100 MAXVALUE 10000 INCREMENT BY 1 START WITH
100 CACHE 20 NOORDER NOCYCLE ;
```

Nov 27 html to jsp to DB communication

Html to jsp to DB s/w interaction

=>For html to jsp communication , we need to place jsp file name or url pattern as "href" attribute value of <a> or "action" attribute value of <form> tag

=> For jsp to to Db s/w we need to place jdbc code in the jsp page/comp/file



points to remember

(a) Differentiate logics for hyperlink and submit button
in the jsp page based on additional req param value that is recieved
(b) try to use all the 3 scipting tags in jsp page
(c) try to use all the 3 jsp life cycle method
(d) gather jdbc properties from web.xml file as jsp init param values
(e) Try to avoid out.println()/print() and out.write()/writeln() methods in the jsp coding .. prefer working with expression tags..

```
CREATE TABLE "SYSTEM"."JSP_PERSON_INFO"
(    "PID" NUMBER(15,0) NOT NULL ENABLE,
     "PNAME" VARCHAR2(20 BYTE),
     "PAGE" NUMBER(3,0),
     "EMAILID" VARCHAR2(20 BYTE),
     "PADD" VARCHAR2(20 BYTE),
     CONSTRAINT "JSP_PERSON_INFO_PK" PRIMARY KEY ("PID"));
```

ctrl+a , ctrl + I :: select all + indentation

PID1_SEQ (sequence in oralce)

```
CREATE SEQUENCE "SYSTEM"."PID1_SEQ" MINVALUE 100 MAXVALUE 10000 INCREMENT BY 1 START WITH
100 CACHE 20 NOORDER NOCYCLE ;
```

person_form.html

```
<h1 style="color:red;text-align:center">HTML TO JSP TO DB Interactions</h1>


<form action="jspdburl" method="POST">
  <table align="center" bgcolor="cyan">
    <tr>
      <td> Person name :: </td>
      <td> <input type="text" name="pname"> </td>
    </tr>
    <tr>
      <td> Person age :: </td>
      <td> <input type="range"  min="1" max="123" name="page"> </td>
    </tr>
    <tr>
      <td> Person address :: </td>
      <td> <input type="text"  name="paddrs"> </td>
    </tr>
    <tr>
      <td> Person email id :: </td>
      <td> <input type="email"  name="mailid"> </td>
    </tr>
    <tr>
      <td><input type="submit"  value="register" name="p1">  </td>
      <td> <input type="reset"  value="cancel" > </td>
    </tr>
  </table>
</form>
  <h2 style="color:red;text-align:center"> <a href="jspdburl/p1+link1" >Get All Person Details</a></h2>
```

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-HtmlToJspToDBInteraction</display-name>
  <welcome-file-list>
    <welcome-file>person_form.html</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>db</servlet-name>
    <jsp-file>/db_jsp.jsp</jsp-file>
    <init-param>
      <param-name>driverClass</param-name>
      <param-value>oracle.jdbc.driver.OracleDriver</param-value>
    </init-param>
    <init-param>
      <param-name>jdbcurl</param-name>
      <param-value>jdbc:oracle:thin:@localhost:1521:xe</param-value>
    </init-param>
    <init-param>
      <param-name>duser</param-name>
      <param-value>system</param-value>
    </init-param>
    <init-param>
      <param-name>dpwd</param-name>
      <param-value>manager</param-value>
    </init-param>

  </servlet>
  <servlet-mapping>
    <servlet-name>db</servlet-name>
    <url-pattern>/jspdburl</url-pattern>
  </servlet-mapping>
</web-app>
```

db_js.jsp

```
<%@ page import="java.sql.*"%>
<%!
   Connection con=null;
   PreparedStatement ps1=null;
   PreparedStatement ps2=null;
   private static final String  INSERT_QUERY="INSERT INTO JSP_PERSON_INFO VALUES(PID1_SEQ.NEXTVAL,?,?,?,??)";
   private static final String  SELECT_QUERY="SELECT PID,PNAME,PAGE,PADD,EMAILID FROM  JSP_PERSON_INFO";

   public void jspInit(){
         //get SErvletconfig  object directly
         ServletConfig cgrngetServletConfig();
         //read init param values  (jdbc properties)
         String driver=cg.getInitParameter("driverClass");
         String url=cg.getInitParameter("jdbcurl");
         String user=cg.getInitParameter("duser");
         String pwd=cg.getInitParameter("dbpwd");
         try{
               //Load jdbc driver class
               Class.forName(driver);
               //establish the connection
               con=DriverManager.getConnection(url, user, pwd);
               //create PreparedStatement objs
               ps1=con.prepareStatement(SELECT_QUERY);
               ps2=con.prepareStatement(INSERT_QUERY);
         }//try
         catch(SQLException se){
               se.printStackTrace();
         }
         catch(Exception e){
               e.printStackTrace();
         }
   }  %>
<%
    //read additional  req param value
    String pval=request.getParameter("p1");
    if(pval.equalsIgnoreCase("register")){  //for submit button
          //read form data
          String name=request.getParameter("pname");
               int age=Integer.parseInt(request.getParameter("page"));
               String addrs=request.getParameter("paddrs");
               String mail=request.getParameter("mailid");
          //set values to query params (?)
          ps2.setString(1,name);
          ps2.setInt(2,age);
          ps2.setString(3,mail);
          ps2.setString(4,addrs);

          //execute the query
          int resultrps2.executeUpdate();
          //process the results
          if(result==0){ %>
<h1 style="color: red; text-align: center">Record not inserted</h1>
<% }
          else{ %>
<h1 style="color: red; text-align: center">Record inserted</h1>
<%   }//else
    }//if
    else{
          //execute Query
          ResultSet rs=ps1.executeQuery(); %>
<table border="1" bgcolor="cyan" align="center">
      <tr>
            <th>Pid</th>
            <th>Name</th>
            <th>Age</th>
            <th>Address</th>
            <th>email</th>
      </tr>

      <%
          //process the Result
          while(rs.next()){ %>
      <tr>
            <td><%=rs.getInt(1) %></td>
            <td><%=rs.getString(2) %></td>
            <td><%=rs.getInt(3) %></td>
            <td><%=rs.getString(4) %></td>
            <td><%=rs.getString(5) %></td>
      </tr>
      <%}//while
          %>
</table>
<%    }//else
          %>
<center> <a href="person_form.html">Home</a></center>


<%!public void jspDestroy(){
      try{
          //close jdbc obks
          if(ps1!=null)
               ps1.close();
          }
          catch(SQLException se){
               se.printStackTrace();
          }


      try{
      if(ps2!=null){
          ps2.close();
      }
          }//try
      catch(SQLException e){
               e.printStackTrace();
      }

          try{
          //close jdbc obks
          if(con!=null)
               con.close();
          }
          catch(SQLException se){
               se.printStackTrace();
          }
    }//jspDestroy() method
    %>
```

Directive tags

=>These tags give directions to jsp page compiler to generate java code JES class ...
=>These tags do not allow to place java code directly.. but we can place directions or instructions for jsp page compiler to reflect the java code in jsp page.

standard syntax
<%@ <directive tag> attributes %>
xml syntax
<jsp:<directive tag name> attributes />

3 types of directive tags
a) page directive
b) include directive
c) taglib directive

page directive

=>gives instructions to jsp page compiler through its attributes to generate global or common java code instructions in JES (jsp equalent sevlet class) source code.

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

the attributes are

a) language (default value :: java)
b) isThreadSafe (default value :: true)
c) isELIgnored (default value :: false)
d) contentType (default value :: text/html;charset=ISO-8859-1
e) pageEncoding (default value :: ISO-8859-1)
f) import (no default value)
g) session (default value ::true)
h) buffer (default value : 8kb)
i) autoFlush (default value : true)
j) info (No default value)
k)errorPage (No default value)
l)isErrorPage (default value ::false)
m)extends (Container supplied JES class)
and etc..

language

=>Allows to specify the language that can be used to write script code in jsp page (especially in scripting tags)
=> As of now "java" is only one language that can be specified and used here.
=> "java" is default language and only the "language" that can specified.

eg1: <%@ page language="java" %> //valid
eg2: <%@ page language="c" %> //invalid
org.apache.jasper.JasperException: /test1.jsp (line: [1], column: [2])
Page directive: invalid language attribute

=>This attribute related information does not reflect in JES class.. It just helps to improve the readability of jsp page code..

note:: If jsp page compiler generates the error in jsp page compilation then the already generated JES soruce code and byte code will be deleted..

JES class for the jsp page of eclipse Project
G:\Workspaces\advjava\ATAN215\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\work\Catalina\localhost\JspApp4-PageDirective\org\apache\jsp

Go to workspace folder serach for "work" or wtpwebapps folder open other folders accordingly..

Security in web applications Nov 28 SEvlet SEcurity

security : authentication + authorization

authentication :: checking the identity of users
authorization :: verifying the access permissions of a users

Banking App

=>To get into Banking App every user should be authenticated
=> Based on the role/desg of the logged-in user the access permissions on the resources will be given or will be denied.
eg: Customer role users can operate only Daily Txs
Clerk role users can operate Daily Txs , Loan operations
MANAGER role users can operate all operations of the App

=> In Security Apps the place where username,password and roles are maintained is called Security realm or Authentication Info Provider

uname	password	ROLES
raja	rani	CLERK,MANAGER
ramesh	hyd	CLERK,MANAGER
suresh	vizag	CUSTOMER
ramana	delhi	VISITOR

=> while performing authorization.. Do not do it by username .. always prefer doing it by role becoz if user is deleted or promoted that would not give problems.

Process

=> As part of user registration process ,insert username,password and role(s)
=> as part of user login activity
->collect username and password to perform authentication
-> collect roles of a user logged-in to perform authorization
i.e to allow or deny access to user on different services/operations offered by the app

Two imp comps Security Implementation

a) Authentication Provider /Authentication Info Provider/ SecurityRealm
b) Authentication Manager

a) Authentication Provider /Authentication Info Provider/ SecurityRealm

=>It is the place where usernames,passwords and roles will stored and managed and will be used for authentication and authorization activities
eg: Xml file | here we need to encrypt
properties file | password manuall .. we can password though forget it
DB s/w
LDAP Server (best) | here passwords automatically encrypted and if we forget password .. we can only reset password i.e we can not get old password

=>Tomcat server is giving support only for xml file by default..
=>Spring/spring boot security supports all types authentication info providers.

LDAP :: Light weight Directory Access Protocol

b) Authentication Manager

=>It is the component that perform both authentication and authorization activies by talking security realm/Auth info provider.

=>More over , it gives error page with error number accordingly
ErrorCode 401 ---> Authetication failed
Error code 403 ---> Authorization failed..

realm = small storage unit

Two types of Authentication manages
a)Programatic Auth Manager
-> We need to develop Authentication Manager manually to perform authentication and authorization activities
b) Declarative Auth Manager
->By adding entries in web.xml file.. we can use the undelying server/container supplied ready made Authentication manager to perform authenticaiton and Authorization activities

The Declarative Auth Manager can perform Authentication activities in four modees

a) BASIC
b) DIGEST
c) FORM
d) CLIENT-CERT

a) BASIC

=>Internally uses base64 algorithm to enconde username and password in travelling
=>Makes the browser to generate dialog box for collecting username,paassword from endusers.
=>Works with all browsers..

b) DIGEST

=>Same as BASIC , but uses MD5 algorithm for encoding and deconding

MD5 :: Message Digestive 5

c) FORM

=>Same as BASIC .. but allows the programmer to design his choice form page for gathering credentials (username,password) from endusers
=>Also allows to cfg error pages of programmer choice for authentication and authorization failure.

d) CLIENT-CERT

=>It is no way related .. Authentication and authozation..
It is all about configuring Digital certificates in the server by enabling "https" protocol.. and bring those digital certficates at client side and using them for encdoing and decoding activities.

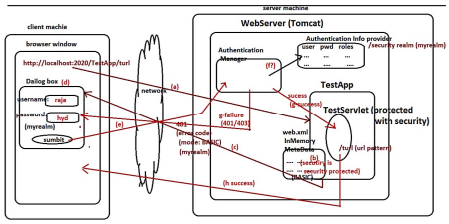
Procedure to place usernames, passwords, roles in tomcat server by taking xml file as the Authentication Info Provider

Go to servers folder of Eclipse Project Explorer
--> Tomcat ---> tomcat-users.xml file

```
<role rolename="ROLE_CUSTOMER"/>
<role rolename="ROLE_MANAGER"/>
<role rolename="ROLE_CLERK"/>
<user username="raja" password="rani" roles="ROLE_CUSTOMER"/>
<user username="ramesh" password="hyd" roles="ROLE_MANAGER,ROLE_CLERK"/>
<user username="anil" password="vizag" roles="ROLE_CLERK"/>
```

Definning roles

users with usernames,passwords and roles



Example on BASIC/DIGEST Mode of Authentication

step1) keep any web application ready and select one or more comps on which u want to apply security

step2) write the following entries in web.xml file to enabl security u r choice comp(s) for u r choice request mode(s)

```
<?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>LinksApp-WorkingMultipleHyperlinks</display-name>
<welcome-file-list>
<welcome-file>links.html</welcome-file>
</welcome-file-list>

<security-constraint>
<web-resource-collection>
<web-resource-name>securingLinks</web-resource-name> <!-- any logical anme -->
<url-pattern>/linksurl</url-pattern> <!-- we can use this tag for multiple times to secure multiple web comps -->
<http-method>GET</http-method> <!-- We can place this for multiple to specify different request modes -->
</web-resource-collection>
<auth-constraint>
<role-name>ROLE_CLERK</role-name>
<role-name>ROLE_MANAGER</role-name>
</auth-constraint>
</security-constraint>
<login-config>
<!-- <auth-method>BASIC</auth-method> --> <!-- Auth mode -->
<auth-method>DIGEST</auth-method>
<realm-name>myrealm</realm-name> <!-- default realm name in Tomcat server-->
</login-config>
```

security cfg

Role name for Authorization

error page cfg for error code

```
<error-page>
<error-code>403</error-code>
<location>/access_denied.html</location>
</error-page>
```

```
<servlet>
<servlet-name>links</servlet-name>
<servlet-class>com.nt.servlet.LinksServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>links</servlet-name>
<url-pattern>/linksurl</url-pattern>
</servlet-mapping>
</web-app>
```

servlet mapping

Nov 28 SEvlet SEcurity

FORM mode Authentication in Servlet Security

=> Allows the programmer to cfg their choice form page and error page

Example App

step1) Create custom form page having fixed action url and fixed form comp names

login_form.html

```
<h1 style="color:red;text-align:center"> Login Page</h1>

<form action="j_security_check" method="POST">
  <table border="1" bgcolor="cyan" align="center">
    <tr>
      <td> user name :: </td>
      <td> <input type="text" name="j_username"> </td>
    </tr>
    <tr>
      <td> password :: </td>
      <td> <input type="password" name="j_password"> </td>
    </tr>
    <tr>
      <td colspan="2"><input type="submit" value="Login"> </td>
    </tr>
  </table>

</form>
```

step2) create custom error page for authentication failure.

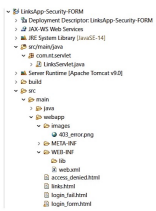
login_fail.html

```
<h1 style="color:red;text-align:center"> Authorizaion failed</h1>
<br>
<br>
<center> <img src="images/403_error.png"> </center>
```

step3) ***Add the follwoing entire in web.xml file enabling FORM Authentication***

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>LinksApp-WorkingMultipleHyperlinks</display-name>
  <welcome-file-list>
    <welcome-file>links.html</welcome-file>
  </welcome-file-list>

  <security-constraint>
    <web-resource-collection>
       <web-resource-name>securingLinks</web-resource-name> <!-- any logical anme -->
       <url-pattern>/linksurl</url-pattern>  <!-- we can use this tag for multiple times to secure multiple web comps -->
       <http-method>GET</http-method>  <!-- We can place this for multiple to specify different request modes -->
    </web-resource-collection>
    <auth-constraint>
       <role-name>ROLE_CLERK</role-name>
       <role-name>ROLE_MANAGER</role-name>
    </auth-constraint>
  </security-constraint>
  <login-config>
    <auth-method>FORM</auth-method>
    <realm-name>myrealm</realm-name> <!-- default realm name in Tomcat server-->
    <form-login-config>
       <form-login-page>/login_form.html</form-login-page>
       <form-error-page>/login_fail.html</form-error-page>
    </form-login-config>
  </login-config>

  <error-page>
    <error-code>403</error-code>
    <location>/access_denied.html</location>
  </error-page>

  <servlet>
    <servlet-name>links</servlet-name>
    <servlet-class>com.nt.servlet.LinksServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>links</servlet-name>
    <url-pattern>/linksurl</url-pattern>
  </servlet-mapping>
</web-app>
```



step4) Run the Web application..

note:: in the target web comp we can write the following code to get more info about logged in users.

```
pw.println("<br><b>Auth mode ::"+req.getAuthType()+"</b>");
pw.println("<br><b>Logged in username ::"+req.getRemoteUser()+"</b>");
```

CLIENT -CERT Model

=> This will be cfg with respect to server by enabling protocol https

https: http over SSL
SSL : Secured Socket Layer

=> Here we prepare digital certificate using RSA or some other Algorithm and we cfg that certificate in server by enabling "https" ..So if the web applications are requested with protocol "https" then the digital cerficate will be downloaded browser side (client machine) ..Now onwords the browser and web application will encode /encrypt the data using digital cerificates..

=> Since digital certificate contains personalized information .. the encryption and decryption that happens is very much stronger and hacker can not decode it /decrypt it..

usecase:: while sending login details over the network
while sendling credit/debit card details over the network
and etc..

procedure

step1) create digital certificate using tool called "keytool" and RSA algorithm
jdk supplied one

```
C:\Users\Naresh1t>keytool -genkey -alias tomcat -keyalg RSA
Enter keystore password:raja1234
Re-enter new password:raja1234
What is your first and last name?
  [Unknown]:  nataraj  ravula
What is the name of your organizational unit?
  [Unknown]:  NIT
What is the name of your organization?
  [Unknown]:  IT Company
What is the name of your City or Locality?
  [Unknown]:  hyd
What is the name of your State or Province?
  [Unknown]:  Telengana
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN="nataraj  ravula", OU=NIT, O=IT Company, L=hyd, ST=Telengana, C=IN correct?
  [no]:  yes
```

creates digital certificate as ".keystore" file in c:\users\<Windows usename> folder

step2) cfg the the digital certificate with Tomcat server by enabling https In server.xml file

In server.xml

```
<!-- Define an SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector
  protocol="org.apache.coyote.http11.Http11NioProtocol"
  port="8443"
  maxThreads="150"
  SSLEnabled="true">
  <SSLHostConfig>
    <Certificate
    certificateKeystoreFile="${user.home}/.keystore"
    certificateKeystorePassword="raja1234"
    type="RSA"
    />
  </SSLHostConfig>
</Connector>
```

step3 :: Run any application using "https" protocol and its port number

https://localhost:8443/CoronaVaccineApp1/ (from firefox)

(1) View Certificate

Accept Risk (2)

(3) proceed further..

In one App we can apply multiple Authentication modes at a tme
a) use BASIC/DIGEST/FORM for Authetication + Authorization
b) use CLIENT-CERT model for sending and recieving data in encrypted manner using the cfg Digital Certificate.

Page Directive tag

Nov 29 Page Directive tag

=>gives global instructions to jsp page compiler to generate java code in JES class

info

=>Useful to provide short description about jsp page
=> This attribute info reflects in JES class by overriding getServletInfo() method

eg:: <%@page info="test jsp page"%>

In JES class

```
public String getServletInfo(){
   return "test jsp page";
}
```

---

import

=>Specifies the java packages to be imported to JES class
=> In the "import" attribute , we can specify multiple packages as the comma seperate list of values.
or by repeating "import" attribute for multiple times.

eg1: <%@page import="java.net.*, java.util.*" %> (or)
eg2: <%@page import="java.net.*" import="java.util.*" %>

In JES class

```
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import java.util.*;
import java.net.*;
```

These are default packages that will imported to every JES class

---

contentType

=>allows to specify the response content type explicitly from jsp page
=> default value is "text/html;charset=ISO-8859-1"

eg1: test1.jsp

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

web page

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

The JES class internally uses res.setContentType(-) method to specify the response content type

eg2: <%@page contentType="text/html"%>

web page

hello

eg3: test1.jsp

```
<%@ page info="This is report page" import="java.util.*, java.net.*"
 contentType="text/html;charset=UTF-8" %>
    <b> hello </b> <br>
    <b>नमस्ते</b> <br>
    <b>హలో</b> <br>
    <b>வணக்கம்</b> <br>
    <b>नमस्कार</b> <br>
    <b>ನಮಸ್ಕಾರ</b> <br>
```

makes the browser to recognize localized content..

(Generated web page)

hello
नमस्ते
హలో
வணக்கம்
नमस्कार
ನಮಸ್ಕಾರ

---

set
what happens if we call response.contentType(-) explicitly after having "contentType" attribute in <%@page %> tag.

Ans) The MIME type specified in response.setContentType(-) method overrides the MIME type specified in contentType attribute of <%@page %> directive tag

In test1.jsp

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

<%@ page info="This is report page" import="java.util.*, java.net.*"
 contentType="text/html;charset=UTF-8" %>

<b> hello </b> <br>
```

web page

```
<b> hello </b> <br>
```

In _jspService(-,-) of JES class

```
response.setContentType("text/html;charset=UTF-8");
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;

out.write("\r\n");
out.write("  ");
response.setContentType("text/plain");
```

---

pageEncondig

=>Allows us to specify charset directly.. instead specifying in contentType attribute we can specify here as shown belwo

eg: test1.jsp

```
<%@ page contentType="text/html" pageEncoding="UTF-8" %>
    <b> hello </b> <br>
    <b>नमस्ते</b> <br>
    <b>హలో</b> <br>
    <b>வணக்கம்</b> <br>
    <b>नमस्कार</b> <br>
    <b>ನಮಸ್ಕಾರ</b> <br>
```

(Generated web page)

hello
नमस्ते
హలో
வணக்கம்
नमस्कार
ನಮಸ್ಕಾರ

---

session

=>specifies wheather implicit obj "session" should be created or not
=> By default the implicit obj "session" will be created.. If u do not want to create then use session="false"
=> we do not "session" object in jsp page.. If that jsp page is not participating in session tracking activites In such cases in we can disable the creation of "session" object by using session="false"

<%@ page session="false" %>
=>The implicit obj "session" will not created

<%@page session="true"%>
=>The implicit obj "session" will be created

---

extends

=>Allows the programmer to specify his choice class as the super class of JES class but that class should perform the following operations
a) should extend from javax.servlet.http.HttpServlet(AC)
b) should override servlet life cycle methods calling jsp life cycle methods as final methods
c) should jsp life cycle methods as the null method defiinitions..
note:: By taking our choice class as the super class of JES class and performing all the above operations is quite complext task .. So better not use attribute. So the container supplied java class becomes JES class automatically.

eg1: test1.jsp

```
<%@ page session="true" extends="com.nt.Test" %>
    <b> hello </b> <br>
```

IN JES class code

```
public final class test1_jsp extends com.nt.Test
    implements org.apache.jasper.runtime.JspSourceDependent,
               org.apache.jasper.runtime.JspSourceImports {
  ....
  ...
}
```

This class must be developed as described avove

---

buffer & autoFlush

=>servlet comp's PrintWriter stream writes given content directly to response object with out taking the support of buffer /cache

buffer/cache :: It is a temporary memory that holds data for temporary period

servlet comp

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

writes to

response object

=>The jsp comp/file writes given content to response object through buffer/cache
=>Once the buffer is filled up, the content will be flushed out to response object automatically or by calling out.flush() method we can also flush the buffer content.

jsp comp

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

buffer/cache

....

(default size is 8kb)

response content

=>The process of moving content from buffer to its destination is called flushing..

The jsp buffer (buffer associated with out object) will be flushed automatically to write its content to destination called response object in the following situations
a) if the buffer is filledup
b) if jsp page execution is completed and the buffer is not filledup

Here we can "buffer" attribute to jsp buffer size and "autoFlush" attribute to enable or disabling autoFlushing
(default value 8kb) (default value "true")

eg1: <%@ page buffer="10kb" autoFlush="true" %>

eg2: <%@ page autoFlush="true" %> *treats 8kb as buffer size*

eg3: <%@ page buffer="10mb" autoFlush="true" %> //Invalid only kbs are allowsed , not the mbs and gbs

=>page directive and its attributes case-sensitive
=> unknown attributes are not allowed
=>except "import" attribute no other can have multiple values seperated with "," symbols
=>except "import" attribute no other can be repeated with different values either in same page directive or in multiple directivees of a jsp page
=> In jsp page , we can place 0 or more page directive tags.

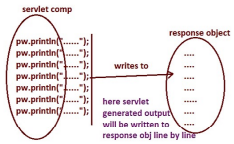
buffer & autoFlush

Page Directive

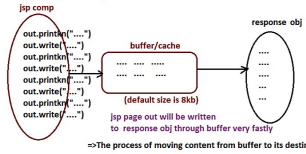
Nov 30 Page Directive tag

=>servlet comp's PrintWriter stream writes given content directly to response object with out taking the support of buffer /cache

buffer/cache :: It is a temporary memory that holds data for temporary period



=>The jsp comp/file writes given content to response object through buffer/cache
=>Once the buffer is filled up, the content will be flushed out to response object automatically or by calling out.flush() method we can also flush the buffer content.



=>The process of moving content from buffer to its destination is called flushing..

The jsp buffer (buffer associated with out object) will be flushed automatically to write its content to destination called response object in the following situations
a) if the buffer is filledup
b) if jsp page execution is completed and the buffer is not filledup

Here we can "buffer" attribute to jsp buffer size and "autoFlush" attribute to enable or disabling autoFlushing
(default value 8kb) (default value "true")

eg1: <%@ page buffer="10kb" autoFlush="true" %>

eg2: <%@ page autoFlush="true" %> takes 8kb as buffer size

eg3: <%@ page buffer="10mb" autoFlush="true" %> //invalid only kbs are allowsed , not the mbs and gbs

=>page directive and its attributes case-sensitive
=> unknown attributes are not allowed
=>except "import" attribute no other can have multiple values seperated with "," symbols
=>except "import" attribute no other can be repeated with different values either in same page directive or in multiple directivees of a jsp page
=> In jsp page , we can place 0 or more page directive tags.

problem code::

eg4:
```
<%@ page buffer="1kb" autoFlush="false" %>

<%
  for(int i=1;i<=4000;++i){
     out.print("raja"+i+"<br>");
  }
%>
```

raises
java.io.IOException: Error: JSP Buffer overflow
becoz webpage is generate more than 1kb data to buffer and autoFlush is disabled .. So overflow on the buffer content takes place.

solution1: enable autoFlush

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

<%
  for(int i=1;i<=4000;++i){
     out.print("raja"+i+"<br>");
  }
%>
```

once the 1kb buffer is filled up its content will be flushed to response obj

solution2: Perform manul flush using out.flush() method

test1.jsp

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

<%
  for(int i=1;i<=4000;++i){
     if(i%20==0)
        out.flush();
     out.print("raja"+i+"<br>");
  }
%>
```

here for every 20 iterations of loop the buffer content will be flushed out to response obj.

How to disable jsp buffer and how to make jsp page content going to response obj directly?

```
<%@page buffer="none" autoFlush="true"%>
        (or)
<%@page buffer="0kb" autoFlush="true"%>
```

It is valid Combo

This indicates jsp page content should go to response obj directly

```
<%@page buffer="0kb" autoFlush="false"%>
        (or)
<%@page buffer="none" autoFlush="false"%>
```

invalid combo becoz we can not disable autoFlushing on buffer when buffer is not there.
raises
org.apache.jasper.JasperException: /test1.jsp (line: [1], column: [2])
jsp.error.page.badCombo

=>In servlet progamming we use PrintWriter stream obj to write the output to response obj.
=>In jsp progamming we use out obj whose type is JspWriter to write the output to response obj.

What is the difference between PrintWriter and JspWriter ?

ctrl + shift +T :: To get the source code of any class

PrintWriter	JspWriter
=>Does not maintain buffer	=> mtaintains buffer
=> It is concreate class of java.io package	=> It is abstract class of javax.servlet.jsp package
=> print() and write() methods of this class does not throw IOException .. rather they catch handle same exception inside the method definitation	=>print() and write() methods of this class throws IOException ..
=> Used in servlet programming as stream object pointing to response obj	=> Used in jsp programming the as the type of "out" object
=> PrintWriter stream object does not use JspWriter object internally..	=> JspWriter object (out obj) intenrally uses PrintWriter object if the jsp page's buffer is not enabled <%@page buffer="none" autoFlush="true">(or)<%@page buffer="0kb" autoFlush="true">

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

Ans)

page	pageContext
a) "page" object type is JES class and reference type java.lang.Object final java.lang.Object page = this;	a) pageContext object type is "org.apache.jasper.runtime.PageContextImpl" and the reference type is "javax.servlet.jsp.PageContext"
b) "page" obj holds the reference of JES class obj	b) "pageContext" obj holds multiple details about current jsp page like request,response , error page name, autoFlush mode, session mode, buffer size and etc.. PageContext pageContext = _jspxFactory.getPageContext(this, request, response, null, false, 1024, false); error page ⇩ session mode ⇩ buffer size ⇩ auto Flush mode
c) useful to differentiate scriptlet tag variable from declaration tag variable being from scriptlet when both have got same name	c) useful to access all implicit objs of jsp page by accessing the "pageContext" obj . Also useful to create different scopes of attributes like page scope,request scope, session scope and application scope attributes

isELIgnored (default value "false")

=>It is always recomanded to minimize or avoid java code directly in jsp page i.e minimize or avoid scripting tags utilization in jsp pages. But performing arithmetic and logical operations using tags is not possible.. To make it possible with out java code jsp supports EL (Expression Language)

EL syntax:
${ <expression> }

It evaluates given expression writes the generated output to browser by using out.print(-) internally..

eg: ${4+5} =>gives 9

=>we can isELIgnore attribute of <%@page %> directive to enable or disable EL Engine

eg1: <%@page isELIgnored="false"%>
${4+8} ----> gives 12 becoz EL is recognized , So arithmetic operations are perfomed
eg2: <%@page isELIgnored="true"%>
${4+8} ----> gives ${4+8} becoz EL is not recognized .. So the EL code will be treated as templace text.

page directive attributes

Dec 01 Page Directive tag

isThreadSafe (default value is "true")

=>Makes the JES class of jsp page as thread safe by implementing SingleThreadModel(I)

eg1: <%@page isThreadSafe="false"%> (meaning is reverse)

=> Makes the JES class of jsp page implementing javax.servlet.SingleThreadModel(I)
to make the JES class thread safe having container support

eg2: <%@page isThreadSafe="true"%> (meaning is reverse)

=>Makes the JES class of jsp page not implementing javax.servlet.SingleThreadModel(I)
So we need to use synchronized blocks explicitly in the code of scriptlet tag.

errorPage (no default value)
=======

=>To specify error page for current jsp page

note: The page that executes when the exception raised in the jsp page is called error page
if the error page is jsp page, we can use the implicit obj exception in that jsp page to display
the details of the exception that is raised.

isErrorPage (default value is false)
==========

=>makes the current jsp page error page having access to the implicit object "exception" to display
exception message.

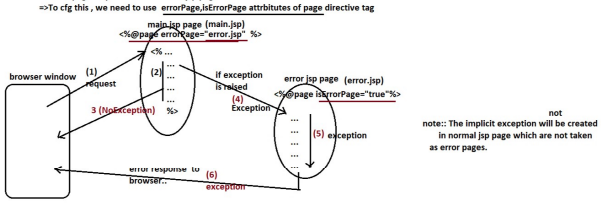
Error pages cfg in jsp programming
==========

=> The page that executes only when exception is raised in other jsp pages is called
Error page .. and it is use to display exception related message as non-technical
guiding messages to enduser..

Two types of error pages cfg in jsp page
(a) Local error pages cfg
(b) Global error pages cfg

Local Error pages cfg
==========

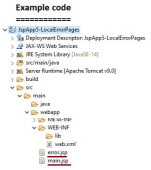
=> Error page is specific to one main jsp page
=>To cfg this , we need to use errorPage,isErrorPage attrbitutes of page directive tag



note:: The implicit exception will be created in normal jsp page which are not taken as error pages.

note: The implicit object "exception" is created only in the jsp page
that is acting as error page
note: we can html file as error page .. but we can not work with "exception"
object in that.

Example code
==========



main.jsp

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

<h1 style="color:red;text-align:center"> main jsp page </h1>

<%    int a =Integer.parseInt("a10"); %>
<br> value :: <%=a %>

<b> end of main.jsp</b>
```

For this NumberFormatException the error page executes.

error.jsp

```
<%@ page isErrorPage="true" %>
<h1 style="color:red;text-align:center">Internal Problem -Try again</h1>
<hr>
Problem is :: <%=exception.toString() %>
```

=>For a main jsp page ,we can configure only one error page at a time

Global Error Pages cfg
==========

=> This error page executes for exception raised all the main jsp pages of a web application
=> That means this error page common for multiple main jsp pages of a web application.
=> This error page cfg we do in web.xml file using <error-page> tag.

using <error-page> tag we can same error page for all exception raised in all main jsp pages
by taking java.lang.Exception as <exception type>

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

using <error-page> tags , we can cfg different error pages for different exceptions.

In web.xml

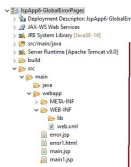
```
<error-page>
    <exception-type>java.lang.NumberFormatException</exception-type>
    <location>/error.jsp</location>
</error-page>
```

Error page for Exception1

```
<error-page>
    <exception-type>java.lang.NullPointerException</exception-type>
    <location>/error1.html</location>
</error-page>
```

Error page for Exception2

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



main1.jsp

```
<h1 style="color:red;text-align:center"> main1 jsp page </h1>

<%    java.util.Date d=null;
      int year=d.getYear();%>
<br> year :: <%=year %>

<b> end of main1.jsp</b>
```

main.jsp

```
<h1 style="color:red;text-align:center"> main jsp page </h1>

<%    int a =Integer.parseInt("a10"); %>
<br> value :: <%=a %>

<b> end of main.jsp</b>
```

error.jsp

```
<%@ page isErrorPage="true" %>

<h1 style="color:red;text-align:center">Internal Problem -Try
again</h1>

<hr>
Problem is :: <%=exception.toString() %>
```

error1.html

```
<h1 style="color:red;text-align:center">Internal Problem -Try again</h1>
<hr>
```

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" version="4.0">

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

error page for all exceptions

(or)

```
  <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>/error1.html</location>
  </error-page>
```

error pages cfg for specific exceptions

=>we can use <error-page> tag, to configure error page for error codes like 401,403,404 and etc.. that raised

```
<error-page>
    <error-code>404</error-code>
    <location>/my404.html</location>
</error-page>
```

my404.html

```
<h1 style="color:red;text-align:center">
  Check the url typed (resource not found)</h1>
<hr>
```

---

if cfg global error page and local error page for the same exception raised in main jsp page
which error page will execute?

Ans) Local Error page executes.

Dec 03 Include Directive

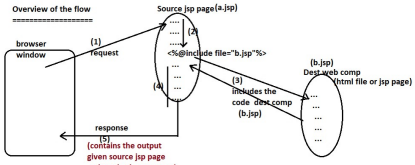
In Jsp programming we have 3 directive tags
a) page b) include c) taglib

Include Directive
==========
=>This tag is given to include the code given web comp like jsp page or html file to the JES class source code of jsp page.
=>This tag is not given to include the output of dest comp with source jsp page JES class source code .. It is given to include the ecode.

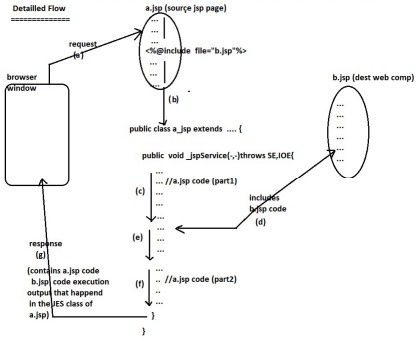
=> This include is also called static include or @include

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

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



=> There will not be seperate execution for dest web comp .. But its code will be included to the JES class of source jsp page i.e in the JES class of source jsp page the jsp code and dest web comp code will be placed togaher



note:: we can not servlet comp as the destination web comp while working with Directive include becoz if servlet comp soruce code is include to the JES class of source jsp page then the entire execution process will be distrubed.
=>if the dest comp is html file or jsp page , then there will not seperate execution for dest web comp but its code will be included to the JES class code of source jsp page.

JES class for a.jsp (source jsp page) having the of a.jsp(soruce page) and b.jsp(dest web comp)

Taglib Directive
==========
=> Jsp tag library is a library that contains set of readily avaiable jsp pages , It is like java package
=> Every Jsp tag library is indentified with its taglib url (some url) .. So to use jsp tags of jsp taglibaries in jsp pages we take the support of Taglib directive (It like import statement that imports java packages)

standard
syntax:
<%@taglib uri=" ....." prefix="...."%>

xml syntax::
<jsp:directive.taglib uri="....." prefix="...." %>

note:: will be discussed and used more while learning Custom tags developments and JSTL

Action tags in jsp
==========

=>These tags internally uses multiple apis including servlet api to place the equalent code in JES class and to complete the requirement.

Different types action tags in jsp
a) Standard action tags
b) Custom Action tags
c) JSTL tags (Jsp standard Tag library tags)
d) Third party Action tags (given by third party vendors like struts , spring mvc and etc..)

note:: Since Action tags coming different vendors , there is no standard syntax for it.. only syntax that is available is xml syntax.

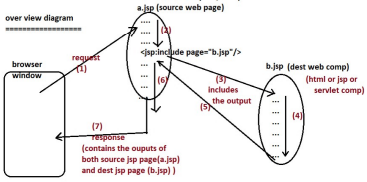
Standard Action tags are
==========
a)<jsp:include>
b)<jsp:forward>
c)<jsp:useBean>
d)<jsp:setProperty>
e)<jsp:getProperty>
f) <jsp:plugin>
g) <jsp:fallback>
h) <jsp:param>
I) <jsp:params>
and etc..

==========
Action include /Dynamic include /<jsp:include>
==========
=>This tag intenrally uses rd.include(-,-) and performs output inclusion of dest web comp to source jsp page
-> This tag does not perform code inclusion of destination comp , but performs output inclusion of dest comp to source jsp page.

=>Directive include is static include (becoz of code inclusion)
where as Action clude is dynamic include (becoz of output inclusion)

Xml syntax:
==========
<jsp:include attribues />

attributes:
page :: Specifies the dest web page/comp
flush :: indicates wheather the source jsp web page's buffer should be flushed or not before including the output of dest webcomp to the buffer of source jsp page (true/false) defualt is "false"



=> Here there will be seperate execution for Dest web comp and its output will be included to output of source jsp page by taking the support of rd.include(-,-) intenally)

Dec 03 Include Directive

==========

Action include /Dynamic include /<jsp:include>

==========

=>This tag internally uses rd.include(-,-) and performs output inclusion of dest web comp to source jsp page

=> This tag does not perform code inclusion of destination comp , but performs output inclusion of dest comp to source jsp page.

=>Directive include is static include (becoz of code inclusion)
where as Action clude is dynamic include (becoz of output inclusion)

Xml syntax:
==========

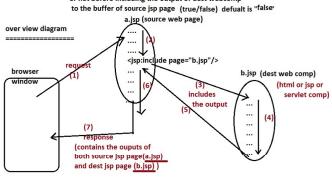
<jsp:include attribues />

attribubes:

page :: Specifies the dest web page/comp

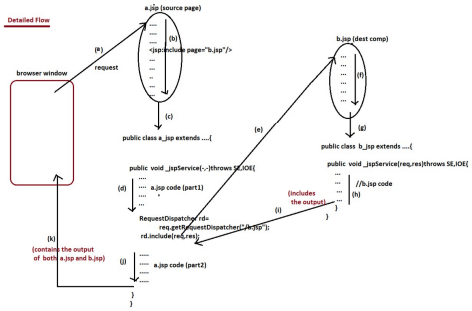
flush :: indicates wheather the source jsp web page's buffer should be flushed or not before including the output of dest webcomp to the buffer of source jsp page (true/false) defualt is "false"

over view diagram



=> Here there will be seperate execution for Dest web comp and its output will be included to output of source jsp page by taking the support of rd.include(-,-) internally)

Detailed Flow



a.jsp
=====

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

b.jsp
====

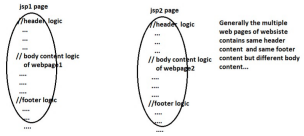
```
<b> start of b.jsp</b> <br>
<%=new java.util.Date() %> <br>
<b> end of b.jsp</b> <br>
```

What is the difference b/w Directive include and Action Include?

Directive Include	Action Include
a) Peforms Code inclusion, So it is called static include or static binding or comaple time binding	a) Performs output inclusion ,So it is called dynamic include or dynamic binding or runtime binding
b) This tag is having both standard and xml syntax	b) This tag having only xml syntax
c) Does not allow to take servlet comp as dest comp	c) allows
d) if the dest comp is jsp comp, there will not any JES class generation for it ..but its code will be inlcuded to the JES class code of source jsp page (a.jsp)	d) if the dest comp is jsp page , then there will be seperate execution for it and its ouput will be included to the output of source jsp pag
e) There will not be seperatte execution for dest web comp	e) there will be seperate execution for destination web comp
f) does not use rd.include(-,-) internally	f) uses rd.include(-,-) internally
g) The Code inlcusion of dest web comp takes place to the JES class code of source JES class at translation phase (page compilation phase)	g) The output inclusion of dest web comp takes place to the source jsp page at execution phase/ runtime
(h) useful , if the destination web comp is static web comp (.html file)	h) useful , if the dest web comp is Dynamic web comp (like servlet, jsp comps)

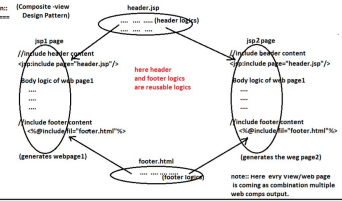
problem::
========

=>Here header, footer logics are not reusable logics.



Generally the multiple web pages of websiste contains same header content and same footer content but different body content...

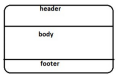
Solution:: (Composite -view Design Pattern)
==========



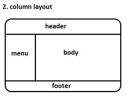
note:: Here evry view/web page is coming as combination multiple web comps output.

Using Compositive Desing pattern we can design multiple web pages of a web application having same layout like

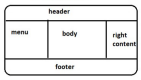
1.Classic Layout



2. column layout

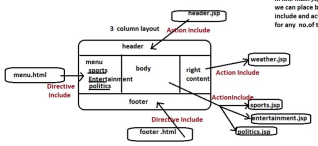


3 column layout

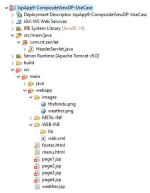


For all these things we need to take web page html table having 100% width and 100% heigth we can those partiations as the columns and rows...

In one main jsp page we can place both directive include and action include for any no.of times.



This application needs four main jsp pages
a)web page b) sports page c)entertain page c) politics page
in which following jsp/html pages content will be included
a)menu.html b) weather.jsp c) footer.html d) header.jsp



<jsp:forward>
==========

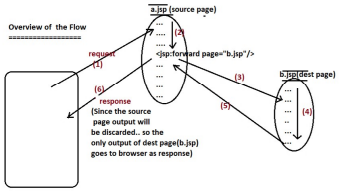
Dec 05 Action Forward

=>This tag internally uses rd.forward(-,-) and perform forward request operation.
=>There is Forward action tag ..but there is no directive forward tag.

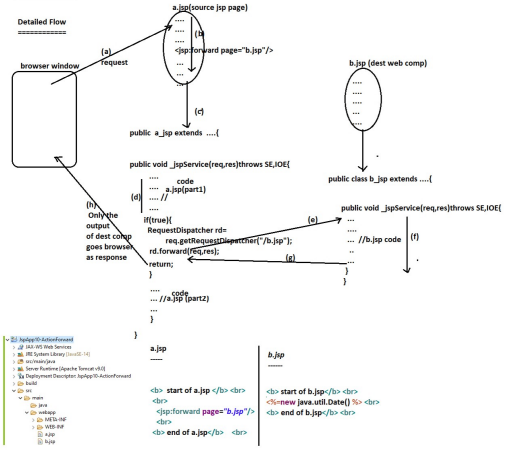
syntax :: <jsp:forward attributes />

attributes
==========
page -- The dest comp's url pattern or jsp page

Overview of the Flow
==========



Detailed Flow
==========



a.jsp

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

b.jsp

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

if the source jsp page is interacting with dest web comp using <jsp:forward> or <jsp:include> tags
then we can say the source jsp page and dest web comp are using same req,res objs .. In that process
we pass data to dest web comp from source jsp page as addtional req param values using <jsp:param> tags

<jsp:param> must be used only the sub tag of <jsp:include> or <jsp:forward>tags.

a.jsp

```
<b> start of a.jsp </b> <br>
<br>
 <% float price= 500 + 500*0.1f; %>
 <jsp:forward page="b.jsp">
   <jsp:param  name="bookName"  value="CRJ"/>
   <jsp:param  name="cost"  value="<%=price%>"/>
 </jsp:forward>
   <br>
<b> end of a.jsp</b>   <br>
```

b.jsp

```
<b> start of b.jsp</b> <br>
<%=new java.util.Date() %> <br>
<b>The req param values
        <%=request.getParameter("bookName") %>,
        <%=request.getParameter("cost") %>
        </b> <br>
<b> end of b.jsp</b> <br>
```

Jsp Communication

The source jsp page can communicate with dest web comp in multiple ways

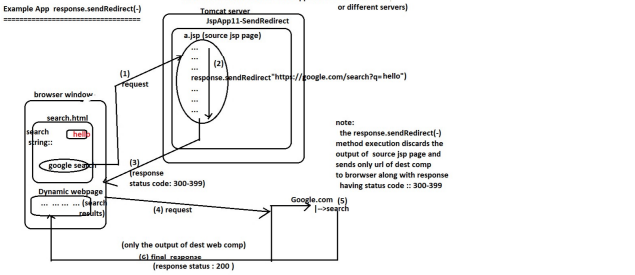
if the source jsp page and dest web comp are using same req,res objs
---
a) <jsp:forward> b) <jsp:include> (Use this if source jsp page and dest web comp are there in the same web application)

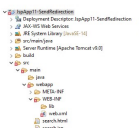
if the source jsp page and dest web comp are not using same req,res objs
---
use response.sendRedirect(url)

note:: There is no tag to perform sendRedirection .. we must use the java code response.sendRedirect(-) explicitly..

(Use this if source jsp page and dest web comp are there in two different web applications of same server or different servers)

Example App response.sendRedirect(-)
==========





search.html

```
<h1 style="color:red;text-align:center"> Example on SendRedirection</h1>

<form action="search.jsp"  method="POST">
  <table 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

```
<%
//reaad req param vlaues
 String ss=request.getParameter("ss");

//perform sendRedirection
response.sendRedirect("https://www.google.com/search?q="+ss);


%>
```

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>JspApp11-SendRedirection</display-name>
  <welcome-file-list>
    <welcome-file>search.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Why there is no directive forward? (when directive include is availalble)

Ans) =>Forward request opeation needs discard the output of source jsp page and only the output of dest comp should go to browser as response.
=>The directive tags perform their operation at traslationphase by including java code in JES class , In this process discarding of output is not possible.. So there is directive forward tag.

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

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

a) request attrbutes (if source jsp page and dest comp are using same req,res objs)
(scope ::request scope)

b) session attributes (if the source jsp page and dest comp are getting request from browser s/w of a client)
(scop: session scope)

c) application attributes (if source jsp page and dest comp are not using same req,res objs and not getting request from same browser)
(scope : application)

d) pageContext attributes
(can create different scopes attributes page , request , session and application)

Since the pageContext obj is having access to all other implicit objects of jsp page, we can use that object to create multiple scope attributes..

not

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

Append query String to the url placed in respsonse.sendRedirect(-) method

```
response.sendRedirect("url?queryString");
```

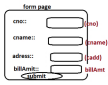
Dec 07 PageContext Attributes

Java bean
=======
=>Java bean is java class that is developed by following standards..
=> It is always used as helper class to carry data from class to another class with in the Project
and from one Project to another Project over the network
=> The standards are
=>class must be public class
=>Recomanded to implement java.io.Serilizable(i)
=>All memeber variables (bean properties) must be private and non-static
=> Every Bean property should have one setter method and 1 getter method
(setter method is given to set/modify the data of bean property where as
getter method is given to read data from bean property)
=>Recomanded to place 0-param constructor directly(given by programmer)
or indirectly (given by java compiler as default constructor)

=>The Form data coming to servlet/jsp comp will be placed in java bean class obj
to send to other classes called service/DAO class. So the multiple values collected from
the form page can be sent to service class(contains b.logic) or DAO class (contains persistnece
logic) as single object

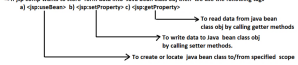
=> If servlet comp wants to put form data into Java bean class obj then it has to create
java bean class obj manually should call setter methods explicitly..

```
//Java bean
class CustomerInfo implements Serilizable{
 //bean properties
 private int cno;
 private String cname;
 private String cadd;
 private float billAmt;
 //setters (or) getters
  ....
  ...
}
```



In servlet comp (storing form data into
============== Java bean class obj)
CustomerInfo customer=new CustomerInfo();
customer.setCno(Integer.parseInt(req.getParameter("cno")));
customer.setCname(req.getParameter("cname"));
customer.setCadd(req.getParameter("cadd"));
customer.setBillAmt(Float.parseFloat(req.getParameter("billAmt")));

=>If jsp comp wants to store form data into Java bean class obj then we use the following tags
a) <jsp:useBean> b) <jsp:setProperty> c) <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 (class name with pkg)
c) scope :: bean class obj scope
page(default) | request | session | application
d) type :: To specify seperate reference type for java bean class obj
(Allow to sepecify the super class name as
reference type)

```
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 use 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 st .setSname(request.getParameter("name")) to assign "name" request param value
to "sname" property.

get
<jsp:getProperty>
==========
==>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"/>
"st.getSname()" method will be called internally to read display
"sname" bean property value

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

//Java bean class
==========
StudentInfo.java

```
package com.nt.beans;

import java.io.Serializable;

public class StudentInfo implements Serializable {
    //bean properties
    private int sno;
    private String sname;
    private String sadd;
    private  float avg;

    public StudentInfo() {
        System.out.println("StudentInfo:: 0-param constuctor");
    }

    //setter and getters  (ctrl+shift+s, r -->select all --> ok)
    public int getSno() {
        System.out.println("StudentInfo.getSno()");
        return sno;
    }
    public void setSno(int sno) {
        System.out.println("StudentInfo.setSno()");
        this.sno = sno;
    }
    public String getSname() {
        System.out.println("StudentInfo.getSname()");
        return sname;
    }
    public void setSname(String sname) {
        System.out.println("StudentInfo.setSname()");
        this.sname = sname;
    }
    public String getSadd() {
        System.out.println("StudentInfo.getSadd()");
        return sadd;
    }
    public void setSadd(String sadd) {
        System.out.println("StudentInfo.setSadd()");
        this.sadd = sadd;
    }
    public float getAvg() {
        System.out.println("StudentInfo.getAvg()");
        return avg;
    }
    public void setAvg(float avg) {
        System.out.println("StudentInfo.setAvg()");
        this.avg = avg;
    }



}
```

request urls
http://localhost:3031/JspApp13-JspToJavaBeanPOC/set_values.jsp
http://localhost:3031/JspApp13-JspToJavaBeanPOC/show_values.jsp

set_values.jsp

```
<%@ page import="com.nt.beans.*" %>


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

<!-- set values to bean properties  -->
<jsp:setProperty name="st"  property="sno"  value="1001"/>
<jsp:setProperty name="st"  property="sname"  value="rakesh"/>
<jsp:setProperty name="st"  property="sadd"  value="vizag"/>
<jsp:setProperty name="st"  property="avg"  value="90.77f"/>

<b> values are set java bean properties</b>
```

show_values.jsp

```
<%@ page import="com.nt.beans.*" %>


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

<!-- set values to bean properties  -->
<jsp:setProperty name="st"  property="sno"  value="1001"/>
<jsp:setProperty name="st"  property="sname"  value="rakesh"/>
<jsp:setProperty name="st"  property="sadd"  value="vizag"/>
<jsp:setProperty name="st"  property="avg"  value="90.77f"/>

<b> values are set java bean properties</b>
```

To set form data/request param values to java bean class obj properties , we can use
param attribute attribute of <jsp:setProperty> tag.

set_values.jsp
==========

```
<%@ page import="com.nt.beans.*" %>


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

<!-- set values to bean properties  -->
<%-- <jsp:setProperty name="st"  property="sno"  value="1001"/>
<jsp:setProperty name="st"  property="sname"  value="rakesh"/>
<jsp:setProperty name="st"  property="sadd"  value="vizag"/>
<jsp:setProperty name="st"  property="avg"  value="90.77f"/>
 --%>
 <!-- setting form data as the req param values -->
  <jsp:setProperty name="st"  property="sno"  param="sno"/>
  <jsp:setProperty name="st"  property="sname"  param="sname"/>
  <jsp:setProperty name="st"  property="sadd"  param="sadd"/>
  <jsp:setProperty name="st"  property="avg"  param="avg"/>
<b> values/req param values are set Java bean properties</b>
```

urls:
http://localhost:3031/JspApp13-JspToJavaBeanPOC/set_values.jsp?sno=1022&sname=soni&sadd=hyd&avg=90.1

http://localhost:3031/JspApp13-JspToJavaBeanPOC/show_values.jsp

---

setting form data as Java bean property values use property="*" symbol
If req param names are matching with java bean class obj property name

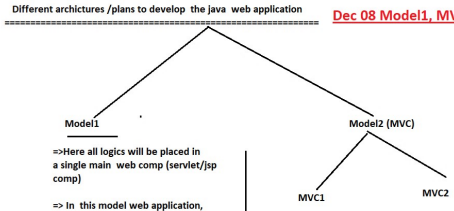
set_values.jsp

```
<%@ page import="com.nt.beans.*" %>


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

<!-- setting form data as Java bean property values use property="*" symbol
  If req param names are matching with java bean class obj property name
 -->
  <jsp:setProperty name="st"  property="*"/>

<b> values/req param values are set Java bean properties</b>
```

request urls:
http://localhost:3031/JspApp13-JspToJavaBeanPOC/set_values.jsp?sno=4545&sname=rakesh&sadd=hyd&avg=56.777
http://localhost:3031/JspApp13-JspToJavaBeanPOC/show_values.jsp

Different architures /plans to develop the java web application

Dec 08 Model1, MVC1 and MVC2 archictures



Model1

=>Here all logics will be placed in a single main web comp (servlet/jsp comp)

=> In this model web application, if servlets are used, the jsp will not be used and vice-versa.

=> So for we have developed all out web application by taking either servlet or jsp as main the web comps . So the apps developed so far comes under Model1 architecture

Model2 (MVC)

=>Here multiple technologies will be used in multiple layers to develop the web applications.

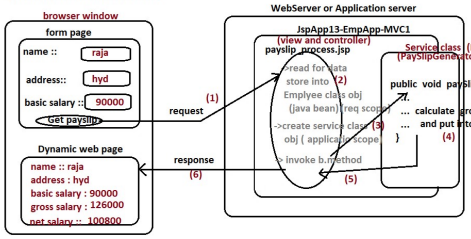
MVC :: Model View Controller

=>In MVC1 Archicture single web comp (servlet/jsp) will be having both View , controller logics and seprate comp like java class will be taken for model layer logics

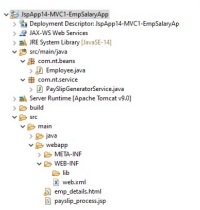
=>In MVC2 architecture , we take seperate comps for view layer (html ,jsp files) and seperate comp for controller (servlet) and seperate comps for Model layer (java classes)

=>MVC1, MVC2 apps layered Apps where
->the java class that contains b.logic is called Serivce class
-> The java class that contains persistence is called DAO class . DAO (DataAccess object class)
->Model layer contains either only service class or service class + DAO class

Example App MVC1 architecture




```
//model class
==========
public class Employee  implements java.io.Serializable{
   private String eadd;
  private String ename;
  private  float salary;
   private  float grossSalary;
   private  float netSalary;
   ...
   ...
   }
```




```
//Java bean class
===============
package com.nt.beans;

import java.io.Serializable;

public class Employee implements Serializable {
	private String ename;
	private String  eadd;
	private  Float basicSalary;
	private  Float grossSalary;
	private  Float netSalary;

	//settes & getters  (alt+shift+s ,r)
	public String getEname() {
		return ename;
	}
	public void setEname(String ename) {
		this.ename = ename;
	}
	public String getEadd() {
		return eadd;
	}
	public void setEadd(String eadd) {
		this.eadd = eadd;
	}
	public Float getBasicSalary() {
		return basicSalary;
	}
	public void setBasicSalary(Float basicSalary) {
		this.basicSalary = basicSalary;
	}
	public Float getGrossSalary() {
		return grossSalary;
	}
	public void setGrossSalary(Float grossSalary) {
		this.grossSalary = grossSalary;
	}
	public Float getNetSalary() {
		return netSalary;
	}
	public void setNetSalary(Float netSalary) {
		this.netSalary = netSalary;
	}

}
```

emp_detials.thml

```
<h1 style="color:red;text-align:center">Employee details (MVC1 Architecture)</h1>

<form action="payslip_process.jsp" method="POST">
 <table border="1" align="center" bgcolor="cyan">
    <tr>
     <td>Employee name:: </td>
     <td><input type="text" name="ename"> </td>
    </tr>
    <tr>
     <td>Employee address:: </td>
     <td><input type="text" name="eadd"> </td>
    </tr>
    <tr>
     <td>Employee Basic salary:: </td>
     <td><input type="text" name="basicSalary"> </td>
    </tr>

    <tr>
     <td colspan="2"><input type="submit" value="GEnerate PaySlip"> </td>
    </tr>

 </table>

</form>
```

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>JspApp14-MVC1-EmpSalaryApp</display-name>
  <welcome-file-list>
    <welcome-file>emp_details.html</welcome-file>
  </welcome-file-list>
</web-app>
```

payslip_process.jsp

```
<%@ page  import="com.nt.beans.*, com.nt.service.*"%>

<!-- create/Locate Java bean class object -->
<jsp:useBean id="emp"  class="com.nt.beans.Employee"  scope="request"/>

<!-- Write form data to Java bean class object -->
<jsp:setProperty name="emp" property="*" />

<!-- Create/Locate  Service class object -->
<jsp:useBean id="service" class="com.nt.service.PaySlipGeneratorService" scope="application"/>

<!--  invoke b.method -->
<%
   service.generatePaySlip(emp);
%>

<!-- Read the inputs and results from Java bean class obj and display on the browser -->
  <h1 style="color:red;text-align:center"> Employee paySlip Report</h1>
 <table align="center"  bgcolor="cyan"  border="1">
  <tr>
     <td> emp name: </td>
     <td> <jsp:getProperty property="ename" name="emp"/>  </td>
  </tr>
  <tr>
     <td> emp address: </td>
     <td> <jsp:getProperty property="eadd" name="emp"/> </td>
  </tr>
  <tr>
     <td> emp basicSalary: </td>
     <td> <jsp:getProperty property="basicSalary" name="emp"/>  </td>
  </tr>
  <tr>
     <td> emp grossSalary: </td>
     <td> <jsp:getProperty property="grossSalary" name="emp"/>  </td>
  </tr>
  <tr>
     <td> emp netSalary: </td>
     <td> <jsp:getProperty property="netSalary" name="emp"/>  </td>
  </tr>
 </table>

<br><br> <a href="emp_details.html">home</a>
```

service class

```
package com.nt.service;

import com.nt.beans.Employee;

public class PaySlipGeneratorService {

	public  void  generatePaySlip(Employee emp) {
		//calculate Gross salary and net salary
		float grossSalary=emp.getBasicSalary()+ emp.getBasicSalary()*0.4f;
		float netSalary=grossSalary-(grossSalary*0.2f);
		//set the generate gross,netSaalry back to Java bean class obj
		emp.setGrossSalary(grossSalary);
		emp.setNetSalary(netSalary);
	}

}
```

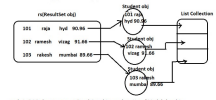
Mini Project using MVC2 architecture

Dec 12 MVC2 Architecture MiniProject

M ---Model Layer (b.logic + persistence logic) ----> java classes (service class + DAO class)
V ---View Layer (presentation logic) ----> html files, jsp files
C ---Controller Layer (Monitoring logic /Navigation Logic) ------> Servlet comp

=>Service class contains b.logic (calculations, filtering,validations, sorting and etc..)
=> DAO class (Data Access object class) contains persistence logics (CURD Operations)
=> Controller class is servlet comp and contains navigation logics i.e logics to talk with service class and the logics to talk view comps.

=>According Layered Apps, we should place jdbc code only in the DAO calss .. not in other classes/layers.
=> So if we get ResultSet obj in DAO class having bunch of records then we can not send that ResultSet object to service class ,controller,view comps as it is. we should convert RS object records into List of Java beanclass objs (Model class objs) then send to Service class to DAO class to View comp.




```
//Model class (Java bean)
public class Student implements Serilizable{
private int sno;
private String sname;
private String sadd;
private float avg;
 //setters  & getters
 ....
 ....
}
```

code in DAO class to convert ResultSet obj records to List of Model clss objs

```
PreparedStatement pss
    con.prepareStatement("SELECT SNO,SNAME,SADD,AVG FROM STUDENT");
ResultSet rs=ps.executeQuery();
  List<Student> lst=new ArrayList();
  while(rs.next()){
    Student st=new Student();
    st.setSno(rs.getInt(1));
    st.setSname(rs.getString(2));
    st.setSadd(rs.getString(3));
    st.setAvg(rs.getFloat(4));
    lst.add(st);
  }
```

Model class :: Java bean
Model Layer :: service classes + DAO classes +model classes

Model class(Java bean) is used as helper class to send multiple values from one layer to another layer or from one class to another class.

Where did use Java bean in a r project?

Ans)= To copy ResultSet records into List collection we have take the support of java bean class as shown above (ResultSet object records to List of Java bean class objs)
=> If u want to send the form page form data to service class as single object then put that data in Java bean class object
=> If want to place multiple values in requst scope/session scope /application scope as single value then combine multiple values in to single value taking the support of java bean class obj (model class obj)

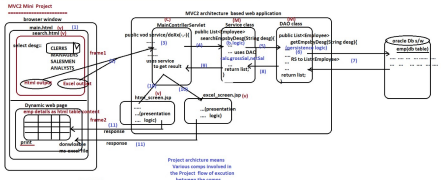
and etc..

When should we Synchronized collection and non-synchronized collection?

Ans) =>After placing data into Collection ,if u want to perform only read operations on it then go for non-Synchronized collocections for better performence becoz they allow multiple threads/Clients /Apps reading data from the collection at a time
eg: ArrayList, HashMap

=>After placing data into Collection ,if u want to perform both read and write operations on it then go for Synchronized collection for thread safety activities
eg: Vector , Hash table

MVC2 Mini Project



Project architecture means
Various comps involved in the Project flow of excution between the comps

Key Points

=>Project is based MVC2 architecture (Layed App)
=> we take the support of Java script for
a) To send request based on local events like onclick,onmouseover etc
b) To make the printout of the web page

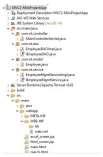
=> Exception raised in DAO calss should be propagated to Service class to controller to view comp using exception rethrowing concept (throws, throw together) in order present to enduser as non-technical message.

=>We are presenting search results in the html table style report and also in the form of downloadable excel file

=>generally , the MVC2 Project contains
a) 1 Main Controller class on 1 per project basis
b) service classes on 1 per module basis
c) DAO classes on 1 per db table basis
d) view comps as needed.

=>If class is having Read Only state or no state it is better to create one object and use it for multiple times Generally DAO ,Service classes contain ReadOnly state and ControllerServlet does not contain State So it is recommended to create only object for those classes .

Order developing comps
a) db tables creation (if needed)
b) pkgs creation in Project
b) model classes (Java beans)
c) DAO classes
d) Service classes
e) View comps giving inputs
f) Controller Servlet
g) View comps displaying the results
and etc..



EL /JSP-EL (Jsp- Expression Language)

=> Writing java code in jsp page is not recomanded...
=> It is recomanded to avoid java code or atleast minimize the java code in jsp page
=> To make jsp page as the java code less jsp page (scriptless jsp page) we can use following things
a) avoid /minimize scripting tags
b) html tags
c) jsp directive tags and action tags
d) JSP EL
e) JSTL tags
f) Third pary tags
g) Custom tags

Asestages of avoiding/minimizing java code in jsp page
a) Improves the reliability of jsp page
b) Suitable for both java developes and UI developers
c) Satifies Jsp Programming priciple called Tag based Programming
d) The java code representing by tags is having reusability in multiple places.

JSP EL
=====
=> By avoiding java code in jsp pages we can not airthmetic and logical operations and we can not those operations using html tags or jsp pags.. So EL is given to write such arithmetic and logical operations code

syntax:
${<expression>}
(It is like expression tag i.e it not only evaluates the expression but also writes the generated output to browser)

To activate and EL Engine

first.jsp

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

${4+5} -->gives 9
(Treated as EL code)
```

EL engine convert EL code to JES class java code

To Deactivate EL Engine

first.jsp

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

${4+5}  --> gives ${4+5}
(Treated as Template text)
```

EL supports
a) EL Operators
b) EL Implicit objects
c) EL Functions

EL Operators

EL is offering both symbols based and text based operation to both arithmetic and logic operations

=>EL Operators are same as Java Operators, But for some Operators text symbols are provided

Category	Operators
Arithmetic	+, -, (binary), *, /, div, %, mod
Logical	and, &&, or, ||, not, !
Relational	==, eq, !=, ne, <, lt, >, gt, <=, le, >=, ge
Conditional	A ? B : C
Empty test checking	empty

test.jsp

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

<br> add ::${4+5}</br> <br>
4 lessthan 5 ? ${4+5} <br>
4 lessthan 5 ? ${4 lt 5} <br>

4 greater than 5 ? ${4 gt 5} <br>
4 greater than 5 ? ${4 + 5} <br>
4 equals to 5 ? ${4==5 } <br>
4 equals to 5 ? ${4 eq 5 } <br>
```

EL Implicit objects

EL gives another 10 implicit objs on top jsp's 9 implicit objs to minimize or avoid java code in jsp page

pageScope
requestScope
sessionScope
applicationScope
cookie
initParam
param
paramValues
header
headerValues

xxxScope implicit objects can be used to read different scopes of data

set_data.jsp

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

<%
pageContext.setAttribute("attr1", "val1");
pageContext.setAttribute("attr2", "val2",PageContext.REQUEST_SCOPE);
pageContext.setAttribute("attr3", "val3",PageContext.SESSION_SCOPE);
pageContext.setAttribute("attr4", "val4",PageContext.APPLICATION_SCOPE);
%>

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

read_data.jsp

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

  attr1 (page  scope) value ::  ${pageScope.attr1} <br>
  attr2 (request  scope) value ::  ${requestScope.attr2}  <br>
  attr3 (session  scope) value ::  ${sessionScope.attr3}  <br>
  attr4 (application  scope) value ::  ${applicationScope.attr4}  <br>
```

=>In The Implicit object cookie is given to read cookie name and values
=> Every jsp page contains session objects and the session id of session object goes to browser in the form of inMemeory cookie whose name is JSessionId.

Cookie name :: ${cookie.JSESSIONID.name}
Cookie value(sessionId) :: ${cookie.JSESSIONID.value}

=> "initParam" Inmplicit object is given to read global init params or context param of the application which are placed in web.xml file

In 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">

  <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>
```

In test.jsp

```
<br>
 context param db user value :: ${initParam.dbuser} <br>
 context param db pwd value :: ${initParam.dbpwd} <br>
```

E) Implicit objs
=============
Dec 14 Custom Jsp Tag Library
=> parseScope,request/scope,session/out/app ,applicationScope, cookie, initParam,
headerValues/values, params, paramValues and etc.

header - It is required to read single request header value
headerValues - It is required to read multi request header values and values.
param - It is required to read single request param value
paramValues - It is required to read all request param names and values

<< Intro Jsp

<!-- create req param value : ${param.username} -->

 <!-- add/+ req params values : ${paramValues.add[0]}, ${paramValues.add[1]} </br>

<!-- create req param values : ${paramValues.username[0]} -->

 <!-- add/+ req params values : ${paramValues.add[0]}, ${paramValues.add[1]} </br>

 <!-- create request header value : ${header["user-agent"]} </br>

 <!-- accept request header value : ${header["accept"]}${header["accept"][1]} </br>

Important Link : http://mathon1992.github.io/11.hr/pm.jsp?answer=ans&edit=true&id=new

---

Custom Jsp Tag Library

=> Jsp tag library is a library that contains set of readily available jsp tags
=> If that Jsp tag library is user-defined tag library then it is called custom tag library
=> Custom Jsp Tag Library contains .tld and .jsp tags representing the relevant java code to execute

Terminologies

=> Jsp Tag Handler class : The Java class extending from javax.servlet.jsp.TagSupport class representing the functionality of jsp tag.
=> .tld file : Tag library descriptor file (xml file) having extension (.tld) contains various tag details like tag name, tag library file, tag name, tag handler class name, attributes names and etc...
=> tagfile.attrib : Jsp tag library is identified with its tagfile.attrib it can be lengthy or it shorter.
.tld and reuse related to it.Internal.
Every jsp tag library is identified with its tagfile.tld given to web.xml file or .tld file (recommended) and To use the attributes we need to import jsp library by specifying its tagfile.tld in <%@taglib> tag of .tld attributes

=>Use page
<%@taglib uri="_____" prefix="_____"%>
_____ _____

Procedure to develop custom tag tag library

eval() Design Jsp tagLibrary
1)Write tag handler class
2)Write tld
3)Use tags

create() Customize the tag handler classes For Jsp Tag

ADD Tag.java
public class ADD Tag extends TagSupport{
.....
public int doStartTag()
throws ...{
.....//Logic to process open tag and its body
.....
return ...;
}
public int doEndTag()
throws ...{
.....//logic to process closing tag.
.....
return ...;
}
}
(or)
Tag Interface
Tag Tag.java
public class Tag implements TagSupport{
public int doStartTag()
throws ...{
.....//Logic to process open tag and its body
.....
return ...;
}
public int doEndTag()
throws ...{
.....//Logic to process closing tag and its body
.....
return ...;
}
}

=>doStartTag() and doEndTag() are the two callback methods which will be executed automatically for open jsp tag <> and closing jsp tag </> tag respectively.

=>The doStartTag() and doEndTag() methods return numberic value representing the work that has to be performed next (indirectly it is showing the jsp container)

javax.servlet.jsp.tagext.Tag#fields
public static final int EVAL_BODY_INCLUDE = 1;
public static final int SKIP_BODY = 0;

javax.servlet.jsp.tagext.IterationTag
public static final int EVAL_BODY_AGAIN = 2;
public static final int EVAL_BODY_INCLUDE = 1;
public static final int SKIP_BODY = 0;

note: Every jsp tag handler class page2.connect object is nullable. so Inherited property and has not access every jsp implicit object from the pageContext object.

step1) Configure the Jsp tags of Custom Jsp Library tld file (Tag Library Descriptor File)

web.xml(no tab) (not Req)
<taglib>
<taglib-uri>http://mywebfc.com/mytags</taglib-uri>
<taglib-location>/WEB-INF/mytag.tld</taglib-location>
</taglib>

<%@ taglib uri="mytags" prefix="my" %>
<%@ taglib uri="http://mywebfc.com/mytags" prefix="my" %>

Tag information will be repeated for multiple tags.

<%@ taglib uri="/WEB-INF/mytag.tld" prefix="my" %>

step2) Import Jsp library in jsp page by specifying the uri and start using the jsp tags in the jsp page. test.jsp

<%@taglib uri="http://mywebfc.com/mytags" prefix="my"%>

<my:tag1/>

<my:tag2/>

The generated outputs of <my:tag1/> tag goes to response obj

Compiler
Container
Browser

Types jsp tags
1)Empty tag
=>Empty tag (<tl:tagLibraryPrefix:tagName/> (Tag with no body)
=>Empty with optional/ mandatory attributes
<tl:tagLibraryPrefix:tagName attr1="..." attr2="..." />
eg: <tl:primenumbers num="10" /> <!-- prints all the prime number between 1 to 100 -->
<tl:primenumbers num="10" range="20" /> <!-- prints prime numbers between 1 to 20 -->
attr1 = "n" default value is 25)

=> Tag with body
<tl:tagLibraryPrefix:tagName attr1="..." attr2="...">
Any Tag Library or HTML or Text
</tl:tagLibraryPrefix:tagName>

<tl:displayMessage> Hello </tl:displayMessage>

=> Tag with body with optional or mandatory attributes
<tl:displayMessage count="value" color="red"> Hello </tl:displayMessage>

<tl:displayMessage count="5" color="red">
hello
</tl:displayMessage>

<tl:displayMessage count="5" color="red">
<%
for(int i=1;i<=5;i++){
%>

hello

<%
}
%>

<%@ taglib uri="/WEB-INF/mytags.tld" prefix="my" %>
<my:tag1/>

<my:tag2/>

PrimeTag.java
=============
package com.nt.tags;

import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;

public class PrimeTag extends TagSupport{
private int n=25;

public void setN(int n) {
System.out.println("PrimeTag.setN()");
this.n = n;
}

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

private boolean isPrime(int x){
for(int i=2;i<=x/2;i++) {
if(x%i==0)
return false;
}
return true;
}

@Override
public int doStartTag() throws JspException {
System.out.println("PrimeTag.doStartTag()");
JspWriter pw=pageContext.getOut();
try {
pw.println("Prime numbers between 1 to "+n+" are:");
for(int i=1;i<=n;i++) {
if(isPrime(i))
pw.println(i+" ");
}
}
catch(Exception e) {
e.printStackTrace();
}
return SKIP_BODY;
}

@Override
public int doEndTag() throws JspException {
System.out.println("PrimeTag.doEndTag()");
return EVAL_PAGE;
}
}

DisplayTag.java
===============
package com.nt.tags;

import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;

public class DisplayTag extends TagSupport {
private int count;
private String color;

public void setCount(String count) {
this.count = count;
}

public void setColor(String color) {
this.color = color;
}

public int doStartTag() throws JspException {
JspWriter pw=pageContext.getOut();
try {
pw.println("");
}
catch(Exception e) {
e.printStackTrace();
}
return EVAL_BODY_INCLUDE;
}

}//doStartTag()

public int doEndTag() throws JspException {
JspWriter pw=pageContext.getOut();
try {
pw.println("");
}
catch(Exception e) {
e.printStackTrace();
}
return EVAL_PAGE;
}

}//doEndTag()

}//class

PrimeTagTest.java
=================
package com.nt.tags;

import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;

public class PrimeTagTest extends TagSupport{

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

@Override
public int doStartTag() throws JspException {
System.out.println("PrimeTagTest.doStartTag()");
return EVAL_PAGE;
}
}

<%@ taglib uri="/WEB-INF/mytags.tld" prefix="my" %>
<my:tag1/>

<my:tag2/>

JSTL Core Tags

Dec 15 JSTL Tags

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

In a way these tags looks like if --- else if --- else and other way the same tags looks like switch --case ... default

core3.jsp

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

<c:choose>
  <c:when test="${param.p lt 0 }">
    <h1 style="color:red;text-align:center"> ${param.p} is Negetive </h1>
  </c:when>
  <c:when test="${param.p gt 0 }">
    <h1 style="color:red;text-align:center"> ${param.p} is Positive </h1>
  </c:when>
  <c:otherwise>
    <h1 style="color:red;text-align:center"> ${param.p} is Zero </h1>
  </c:otherwise>

</c:choose>
```

http://localhost:3031/JSTLCoreTagsApp/core3.jsp?p=100

<c:forEach>

=>It can be used as traditional for loop, enhaced for loop

core4.jsp

```
<!-- As Trandtional for loop -->
<b> Using Tradtional for loop</b>
<table border="1" bgcolor="cyan">
<c:forEach var="i" begin="1" end="10" step="1">
  <tr>
    <td>2 * ${i } = ${2*i} </td>
  </tr>
</c:forEach>
</table>
<hr>
<!-- As Enhanced for loop -->
<b> Using Enhanced for loop</b>
<jsp:scriptlet>
   String names[]=new String[]{"raja","ravi","charl","karan","anil"};
   request.setAttribute("namesList",names);
</jsp:scriptlet>
<c:forEach var="name" items="${namesList}" >
    <h1>${name}</h1>
</c:forEach>
<hr>
<b> Using Enhanced for loop</b>
<jsp:scriptlet>
 <![CDATA[
  List<String> courses=new ArrayList();
  courses.add("java"); courses.add(".net");
  courses.add("php"); courses.add("python");
  request.setAttribute("coursesList",courses);
  ]]>

</jsp:scriptlet>

<c:forEach var="course" items="${coursesList}" >
    <h1>${course}</h1>
</c:forEach>
```

Core5.jsp

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

<h1> request param names and values</h1>

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

</c:forEach>

<h1> request header names and values</h1>

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

</c:forEach>
```

request url :: http://localhost:3031/JSTLCoreTagsApp/core5.jsp?sno=101&sname=raja&sadd=hyd&sadd=vizag

---

<c:forTokens>

=>Useful to split the given text content into multiple tokens based on the given delimeter.

=> It is similar to StringTokenizer concept

core6.jsp

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

  <c:set var="msg" value=" hello, how are u ?" scope="request"/>

  <c:forTokens var="token" items="${msg}" delims=" ">
     ${token} <br>
  </c:forTokens>
```

<c:url> :: Defines the url having scope

<c:import> :: imports the content of one jsp page in another jsp page

core7.jsp

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

<b> from core7.jsp (start)</b>

<c:url var="file" value="core4.jsp" scope="request"/>

<c:import url="${file}"/>

<b> from core7.jsp (end)</b>
```

<c:redirect> performs Send Redirection activity

This tag internally uses response.sendRedirect(-) method to perform SendRedirection activity

core8.jsp

```
%@taglib url="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<b> from core8.jsp (start)</b>
<c:url var="googleUrl" value="https://google.com/search?q=java" scope="request"/>
  <c:redirect url="${googleUrl}"/>

<b> from core8.jsp (end)</b>
```

<c:catch>

This tag is given for Exception handling

core9.jsp

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

<c:catch var="e">
  <jsp:scriptlet>
     java.util.Date d=null;
     d.getYear();
  </jsp:scriptlet>
</c:catch>
  <b>The Raised exception is :: ${e}</b>
```

Procedure to write html_print.jsp content of MiniProject using JSTL core Taglibrary tags

step1) add JSTL jar files to the Build path of the Project and also to the WEB-INF/lib folder

step2) Re write the code of html_print.jsp as shown below by avoding the java code

html_print.jsp

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

<c:choose>
  <c:when test="${!empty empDetails}">
    <h1> Employees Details Belonging to ${param.job}</h1>
    <table border="1" align="center" bgcolor="cyan">
    <tr><th> empno </th> <th> emp name </th> <th> desg </th> <th> salary </th><th> grossSalary </th> <th> netSalary </th> </tr>
     <c:forEach var="emp" items="${empDetails}">
      <tr>
       <td>${emp.eno} </td>
       <td>${emp.ename} </td>
       <td>${emp.desg} </td>
       <td>${emp.salary} </td>
       <td>${emp.grossSalary} </td>
       <td>${emp.netSalary} </td>
      </tr>
     </c:forEach>
    </table>
  </c:when>
  <c:otherwise>
      <h1 style="color:red;text-align:center"> Employees Not found </h1>
  </c:otherwise>
</c:choose>


<br><br>
<center> <a href="JavaScript:showPrint()">print</a></center>

<script language="JavaScript">
  function showPrint(){
  frames.focus();
  frames.print();
  }
</script>
```

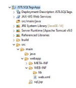
JSTL SQL Taglibrary **Dec 19 JSTL Tags**

==================

=>Gives tags for DB interactions..
=> placing persistence logic in view comp(jsp) is against of MVC2 rules, So this taglibrary is not so useful while work with MVC2 architecture.. This taglibrary is use ful while developing Model1 archicture (all logics in single jsp comp) based web application.

recomanded prefix :: sql
taglib uri :: http://java.sun.com/jsp/jstl/sql
tld file name : sql.tld

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



jars (jstl two jar files)
+
ojdbc6/7/8.jar file

sql.jsp
=====

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

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

 <h1>  The records  are </h1>
<!-- Prcess the ResultSet obj -->
<table border="1" bgcolor="cyan" align="center" >
   <tr> <th> empno </th> <th> ename </th> <th>  job  </th> <th> salary </th> </tr>

 <c:forEach var="e"  items="${rs.rows}">
  <tr>
    <td> ${e.empno} </td>
    <td> ${e.ename} </td>
    <td> ${e.job} </td>
    <td> ${e.sal} </td>
  </tr>
 </c:forEach>
</table>
```

sql1.jsp
========

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

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

<!-- Send and execute the SELECT SQL query -->
<sql:update dataSource="${ds}" var="count"  sql="UPDATE EMP SET SAL=SAL+? WHERE  JOB=?">
  <sql:param value="500"/>
  <sql:param>CLERK</sql:param>
</sql:update>

<b> ${count}  no.of records are updated</b>
```

note: <sql:setDataSource> for creating connection with Db s/w
<sql:query> for executing SQL Select Query
<sql:update> for executing SQL Non-Select Queries

Formatting TagLibrary
=================

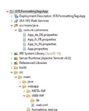
=> Tags are given supporting Internationalization (I18n) where we can format the numbers ,dates , currency symbols, messages/labels and etc.. based on the Locale that we have chosen

Locale means Language + country
eg: en-US (english as speaks in USA)
eg: en-AU (english as speaks in Austrilia)
eg: fr-FR (french as speaks in France)
eg: de-DE (German as it speaks in Gernay)
eg: hi-IN (hindi as it speaks in India)

and tc..

taglib uri:: http://java.sun.com/jsp/jstl/fmt
recomanded prefix :: fmt
tld file name :: fmt.tld

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



App.properties ***(Base file)***

#For english
welcome.msg=Welcome to JSTL tags

The base file name should reflect all locale specific properties file names

App_fr_FR.properties
----
welcome.msg=Bienvenue dans les balises JSTL

=>The keys must be same in all properties files
=>we can use google translator to get the content in different languages.

App_de_DE.properties
----
welcome.msg=Willkommen bei JSTL-Tags

App_hi_IN.properties
----
welcome.msg=JSTL \u091F\u0948\u0917\u094D\u0938 \u092E\u0947 \u0902 \u0906\u092A\u0915\u093E \u0938\u094D\u0935\u093E\u0917\u0924 \u0939\u0948

formatting_test.jsp

```
<%@ page  isELIgnored="false"  contentType="text/html; charset=UTF-8"   %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt"  prefix="fmt"%>

 <fmt:setLocale value="fr_CA"/>

<fmt:formatNumber var="salary" value="4500000"   type="currency"/>
 <b>Salary :: ${salary}</b>  <br>

 <fmt:formatNumber var="range" value="4500000.999"   type="number"
           maxFractionDigits="3"  maxIntegerDigits="10" minIntegerDigits="4"/>
<b>ranage :: ${range}</b> <br>

 <fmt:formatNumber var="avg" value="4.5"   type="percent"/>
<b>avg :: ${avg}</b>
<br>
<jsp:useBean id="dt" class="java.util.Date"/>
<fmt:formatDate value="${dt}"  var="fdate"  type="both"  dateStyle="full" />
<b> date and time :: ${fdate}</b>

<fmt:setBundle basename="com/nt/commons/App"/>
<fmt:message  var="message" key="welcome.msg"  />
<br> <b>message is :: <b>${message}</b>
```

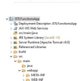
JSTL functions taglibrary
======================

=> Lots if EL Functions are given to to manipulate the String content.

recomanded to prefix: fn
tld file : fn.tld
taglib uri : http://java.sun.com/jsp/jstl/functions

Following table lists out the various JSTL Functions −

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.



functions_test.jsp
================

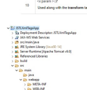
```
<%@ page  isELIgnored="false"  contentType="text/html; charset=UTF-8"
%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/functions"  prefix="fn" %>

<c:set var="msg"  value="welcome to  JSTL" scope="request"/>

 content ::  ${msg } <br>
 uppercase ::  ${fn:toUpperCase(msg)}<br>
 lowercase ::  ${fn:toLowerCase(msg)}<br>
 subString ::  ${fn:substring(msg,0,7) } <br>
 subString After ::  ${fn:substringAfter(msg, "to") } <br>
```

JSTL Xml taglibrary
===============

=>Given to parse and process xml file/document.

recomanded prefix : x
tld file : x.tld
tagliburi :: http://java.sun.com/jsp/jstl/xml

S.No.	Tag & Description
1	<x:out> Like <%= ... >, but for XPath expressions.
2	<x:parse> Used to parse the XML data specified either via an attribute or in the tag body.
3	<x:set > Sets a variable to the value of an XPath expression.
4	<x:if > Evaluates a test XPath expression and if it is true, it processes its body. If the test condition is false, the body is ignored.
5	<x:forEach> To loop over nodes in an XML document.
6	<x:choose> Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by **<when>** and **<otherwise>** tags.
7	<x:when > Subtag of **<choose>** that includes its body if its expression evalutes to 'true'.
8	<x:otherwise > Subtag of **<choose>** that follows the **<when>** tags and runs only if all of the prior conditions evaluates to 'false'.
9	<x:transform > Applies an XSL transformation on a XML document
10	<x:param > Used along with the **transform** tag to set a parameter in the XSLT stylesheet



(old jstl jars)
For xml processing
(download all these jars from mvnrepository.com)

orders.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<orders>
  <order>
    <name>table</name>
    <price>5000</price>
  </order>
  <order>
    <name>fan</name>
    <price>10000</price>
  </order>
  <order>
    <name>TV</name>
    <price>40000</price>
  </order>
  <order>
    <name>chair</name>
    <price>40000</price>
  </order>
</orders>
```

xml _test.jsp
=================

```
<%@ page  isELIgnored="false"  contentType="text/html; charset=UTF-8"   %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/functions"  prefix="fn" %>

<c:set var="msg"  value="welcome to  JSTL" scope="request"/>

 content ::  ${msg } <br>
 uppercase ::  ${fn:toUpperCase(msg)}<br>
 lowercase ::  ${fn:toLowerCase(msg)}<br>
 subString ::  ${fn:substring(msg,0,7) } <br>
 subString After ::  ${fn:substringAfter(msg, "to") } <br>
```

mostly use jstl tags are coretags, fomatting tags, JSTL functions

Scripting tags in jsp (Given to place java code in jsp page)

a) Scriptlet b)Expression c) Declaration

=>The code that is suronded by tags is called script code ..
=> The Java script code surounded by html tags is called script code
=> The Java code surounded by jsp tags is called script code

to

a) Scriptlet

=>Given to place java code in jsp age which goes _jspService(-,-) of JES class as it is..

standary syntax

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

xml syntax

```
<jsp:scriptlet>
   ....
   ...
</jsp:scriptlet>
```

=> The variables declared in scriptlet becomes local variables of _jspService(-,-) method in JES class
=> All the implicit objs of Jsp are orginally local objs of _jspService(-,-) method.. So they can used in scriplet also

first.jsp

```
<% int a=10;
   out.println("Sqaure value:"+(a*a));
 %>
<b> end of jsp page </b>
```

In JES class

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

 public void _jspService(req,res)throws SE,IOE{
   ...
   ... //implicit objs
   ..
   ...
   int a=10;
   out.print("square value :"+(a*a));

   out.write("<b> end of jsp page </b>");

 }
}
```

=> we can call methods in scriplet ..but we can not define methods in scriptlet as java does not support nested methods.

first.jsp

```
<%
  out.println("browser name::"+request.getHeader("user-agent"));
  String s="hello";
     out.println("<br>length of "+s+" is "+s.length());
%>
<%
  public void m1(){

}
%>
```

Invalid becoz java does not support nested method definitation (method inside the method)

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

 public void _jspService(req,res)throws SE,IOE{
   ...
   ... //implicit objs
   ...
   ...
   out.write('\r');
    out.write('\n');

   out.println("browser name::"+request.getHeader("user-agent"));
   String s="hello";
      out.println("<br>length of "+s+" is "+s.length());

    out.write("\r\n");
    out.write("\r\n");
    out.write("\r\n");

   public void m1(){
  }
```

Javac raises error for compilation.

```
 }

}
```

=>we can place class defination inside the scriplet as java supports method level local class definitations

first.jsp

```
<%
  class Test{
  }
%>

<b> hello </b>
```

class defination

The Local classes must be take with out access modifiers. (private/public/protected/default)

JES class

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

 public void _jspService(req,res)throws SE,IOE{

   ...
   ... //implicit objs
   ...
   ...
   class Test{
     ..
     ..
   }

   out.write("<b> hello </b>");

 }
}
```

types of inner classes in java
a) inner classes
b)nested inner classes (static classes)
c) Local inner classes
d) Anonymous inner classes.

Since java does not support Local interfaces ,we can say that we can not place interfaces in scriptlet of jsp page..

first.jsp

```
<%
interface Demo{
 }
%>

<b> hello </b>
```

In JES class

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

 public void _jspService(req,res)throws SE,IOE{
   ...
   ...
   ... //implicit objs
   ...
   ...
   interface Demo{
   }
```

Java does not support inner interfaces..

```
   out.write("\r\n");
   out.write("\r\n");
   out.write("<b> hello </b>");
 }
}
```

---

note:: we generally uses scriptlet for placing request processing logics or b.logics.

Xml syntax of scriplet

first.jsp

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

In JES class

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

 public void _jspService(req,res)throws SE,IOE{
   ...
   ...
   ... //implicit objs
   ...
   ...
   out.write("\r\n");
    out.write("\r\n");

   int a=10;
      int b=20;
      out.println("result ::"+(a+b));

   out.write('\r');
   out.write('\n');
 }
}
```

"<" symbol problem with "xml syntax"

first.jsp

```
<jsp:scriptlet>
   int a=10;
   int b=20;
   out.println(" a less than b?"+(a<b));
</jsp:scriptlet>
```

This less than symbol ("<") is taken as tag begining symbol

error: org.apache.jasper.JasperException: /first.jsp (line: [3], column: [15]) Unterminated [<jsp:scriptlet>] tag

---

solution1:: (not good) **Using standard syntax::**

first.jsp

```
<%
   int a=10;
      int b=20;
      out.println(" a less than b?"+(a<b));
 %>
```

Solution2:: (Best)

first.jsp

```
<jsp:scriptlet>
 <![ CDATA[
   int a=10;
      int b=20;
      out.println(" a less than b?"+(a<b));
   ]]>
</jsp:scriptlet>
```

<![[CDATA[...]]> makes xml parsert of Jsp page complier to given data as the character data with out applying any xml meaning on it.

note1: In one jsp page we can place 0 or more scriptlet tag either having xml syntax or standard syntax or both

note2: Nested scritplets are not allows i.e we can not place scriptlet tag inside another scriptlet tag.

```
<%
   int a=10;
      int b=20;
      out.println(" a less than b?"+(a<b));
   <%
   int c=a+b;
   %>
%>
```

//invalid code

note:: For the code placed in scriplet , we need not to handle exceptions as the _jspSerivce(-,-) method takes care of excepton handing..