CLASS-1
JSP
=====
JSP stands for Java Server Pages.
JSP is a web resource program which is used to develop dynamic web pages.

Limitations with servlets

To work with servlet strong java knowledge is required.
It is not suitable for non-java programmers.
Handling exceptions are mandatory.
Configuration of each servlet program in web.xml file is mandatory.
It does not give any implicit object.
(Object which can be used without any configuration is called implicit object)
We can't maintain html code and java code separately.

Advantages of JSP

To work with jsp strong java knowledge is not required.
It is suitable for java and non-java programmers.
Handling exceptions are optional.
Configuration of each jsp program in web.xml is optional.
It gives 9 implicit objects.
We can maintain HTML code and Java code seperately.
It support tag based programming.
It allows us to work with custom tags.
It contains all the features of servlet.

First web application development having jsp program as web resource program
===
Deployment Directory structure

```
JspApp1
|
|-----Java Resources
|

|-----Web Content
        |
        |---ABC.jsp
        |
        |---WEB-INF
                |
                |---web.xml
```
Note:

In above application we need to add "servlet-api.jar" file in project build path.
ABC.jsp

```
<center>
        <h1>
```

```
		Current Date and Time: <br>
		<%
			java.util.Date d=new java.util.Date();
			out.println(d);
		%>
	</h1>
</center>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
	<welcome-file>ABC.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Request url

```
	http://localhost:2525/JspApp1/
```

Configuration of JSP program in web.xml file

====================================

Deployment Directory structure

--

```
JspApp1
|
|-----Java Resources
|

|-----Web Content
	|
	|---ABC.jsp
	|
	|---WEB-INF
		|
		|---web.xml
```

Note:

In above application we need to add "servlet-api.jar" file in project build path.

ABC.jsp

```
<center>
	<h1>
		Current Date and Time: <br>
		<%
			java.util.Date d=new java.util.Date();
			out.println(d);
		%>
	</h1>
```

```
</center>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <servlet>
        <servlet-name>ABC</servlet-name>
        <jsp-file>/ABC.jsp</jsp-file>
  </servlet>
  <servlet-mapping>
        <servlet-name>ABC</servlet-name>
        <url-pattern>/test</url-pattern>
  </servlet-mapping>
</web-app>
```
Request url

```
        http://localhost:2525/JspApp1/ABC.jsp
        http://localhost:2525/JspApp1/test
```

How can we access web application through only url pattern.It means how can we hide web application accessible through file name

==

We can access web application though only url pattern if we placed ABC.jsp file inside "WEB-INF" folder.

Deployment Directory structure

--
```
JspApp1
|
|-----Java Resources
|

|-----Web Content
        |
        |
        |---WEB-INF
                |---ABC.jsp
                |
                |---web.xml
```
Note:

In above application we need to add "servlet-api.jar" file in project build path.

ABC.jsp

```
<center>
        <h1>
                Current Date and Time: <br>
                <%
                        java.util.Date d=new java.util.Date();
                        out.println(d);
```

```
				%>
			</h1>
</center>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <servlet>
		<servlet-name>ABC</servlet-name>
		<jsp-file>/WEB-INF/ABC.jsp</jsp-file>
  </servlet>
  <servlet-mapping>
		<servlet-name>ABC</servlet-name>
		<url-pattern>/test</url-pattern>
  </servlet-mapping>
</web-app>
```
Request url

```
		http://localhost:2525/JspApp1/ABC.jsp //404 error
		http://localhost:2525/JspApp1/test    //valid
```

Life cycle methods of JSP

===========================

We have three life cycle methods of JSP.

1)_jspInit()

It is used for instantation event.
This method will execute just before JES class creation.
JES stands for Java Equivalent Servlet class.

2)_jspService()

It used for request arrival event.
This method will execute when request goes to jsp program.

3)_jspDestroy()

It is used for destruction event.
This method will execute just before JES class destruction.

Phases in JSP

==============

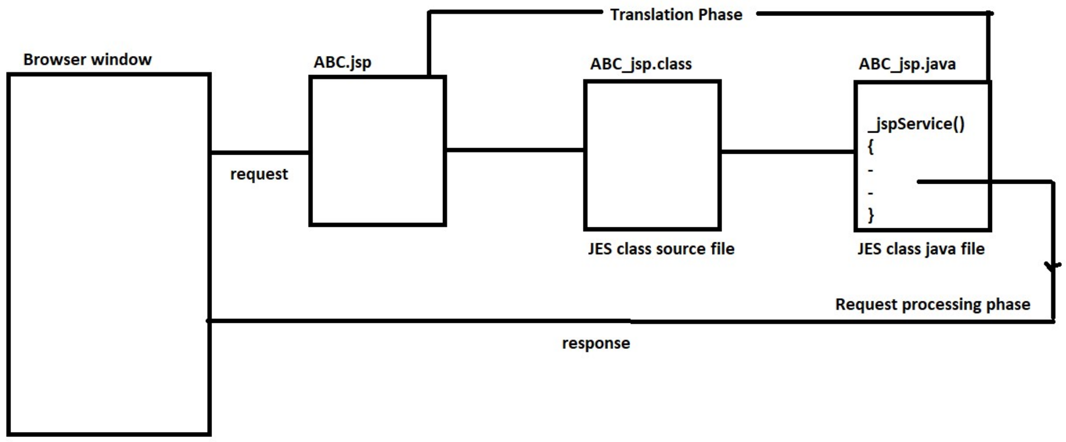
We have two types of phases in JSP.

1)Translation phase

In translation phase our JSP program converts to JES class i.e ABC_jsp.class, ABC_jsp.java.

2)Request processing phase

In request processing phase our request to JES class object and result will send to browser window as dynamic response.



How to enable <load-on-startup> and what happens if we enable <load-on-startup>

==

If we enable load-on-startup then translation phase will be performed during the server startup or during the deployment of web application.

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
    <servlet>
            <servlet-name>ABC</servlet-name>
            <jsp-file>/WEB-INF/ABC.jsp</jsp-file>
            <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
            <servlet-name>ABC</servlet-name>
            <url-pattern>/test</url-pattern>
  </servlet-mapping>
</web-app>
```

JSP tags/elements

====================

JSP contains mainly three tags.

1)Scripting tags

It is divided into three tags.

i) scriptlet tag

ex:

```
<%   code here   %>
```

ii) expression tag

ex:

```
                <%=   code here  %>
```

iii) declaration tag

ex:

```
                <%!  code here   %>
```

2)Directive tags

It is divided into two types.

i) page directive tag

ex:

```
                <%@page   attribute=value %>
```

ii)include directive tag

ex:

```
                <%@include  attribute=value %>
```

3)Standard tags

We have following list of standard tags.

ex:

```
        <jsp:include>
        <jsp:forward>
        <jsp:useBean>
        <jsp:setProperty>
        <jsp:getProperty>
        and etc.
```

JSP comment

```
        <%--   comment here   --%>
```

i) scriptlet tag

===========

It is used to declare java code.

syntax:

```
        <%   code here   %>
```

Deployment Directory structure

--

```
JspApp2
|
|-----Java Resources
|

|-----Web Content
        |
        |---form.html
        |
        |---process.jsp
        |
        |---WEB-INF
                    |
                    |---web.xml
```

Note:

In above application we need to add "servlet-api.jar" file in project build path.

form.html

```
<form action="process.jsp">
            Name: <input type="text" name="t1"/> <br>
            <input type="submit" value="submit"/>
</form>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
            <welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```

process.jsp

```
<center>
<h1>
<%
            String name=request.getParameter("t1");
            out.println("Welcome :"+name);
%>
</h1>
</center>
```

request url

```
            http://localhost:2525/JspApp2/
```

ii) expression tag
==============

The code which is written in expression tag will return to the output stream of a response.It means we don't need to write out.println() to print the data.

syntax:

```
            <%=   code here  %>
```

Expression tag does not support semicolon.

Deployment Directory structure

```
JspApp2
|
|-----Java Resources
```

```
|

|-----Web Content
            |
            |---form.html
            |
            |---process.jsp
            |
            |---WEB-INF
                        |
                        |---web.xml
```
Note:

In above application we need to add "servlet-api.jar" file in project build path.

form.html

```
<form action="process.jsp">

	Name: <input type="text" name="t1"/> <br>
	<input type="submit" value="submit"/>
</form>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
	<welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```
process.jsp

```
<center>
<h1>
<%
	String name=request.getParameter("t1");
%>
<%= "Welcome to Ihub : "+name %>
</h1>
</center>
```
request url

```
	http://localhost:2525/JspApp2/
```

iii) declaration tag
======================
It is used to declare fields and methods.
syntax:
```
	<%!  code here   %>
```
Deployment Directory structure

--

```
JspApp3
|
|-----Java Resources
|

|-----Web Content
            |
            |---index1.jsp
            |
            |---index2.jsp
            |
            |---WEB-INF
                        |
                        |---web.xml
```

Note:

In above application we need to add "servlet-api.jar" file in project build path.

index1.jsp

```
<%!
        int i=100;
%>
<%= "The value of i is ="+i %>
```

index2.jsp

```
<%!
        int cube(int n)
        {
                return n*n*n;
        }
%>
<%= "cube of a given number is ="+cube(5) %>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
</web-app>
```

request url

```
        http://localhost:2525/JspApp3/index1.jsp
        http://localhost:2525/JspApp3/index2.jsp
```

CLASS-2

Exception Handling in JSP

==========================

There may chance of raising exception anywhere in our application so handling the exception is safer side for the programmer.

Runtime errors are called exceptions.

Exceptions always raised at runtime so they are also known as runtime events.
In jsp , exception handling can be performed in two ways.

1)By using errorPage and isErrorPage attribute of page directive tag.
2)By using <error-page> element in web.xml file.

1)By using errorPage and isErrorPage attribute of page directive tag
---
Deployment Directory structure
--

```
JspApp4
|
|-----Java Resources
|
|-----Web Content
	|
	|---form.html
	|---process.jsp
	|---error.jsp
	|
	|---WEB-INF
		|
		|---web.xml
```

Note:

In above application we need to add "servlet-api.jar" file in project build path.
form.html

```
<form action="process.jsp">
	No1: <input type="text" name="t1"/> <br>
	No2: <input typ="text" name="t2"/> <br>
	<input type="submit" value="divide"/>
</form>
```

process.jsp

```
<%@page errorPage="error.jsp" %>
<%
	String sno1=request.getParameter("t1");
	String sno2=request.getParameter("t2");
	int a=Integer.parseInt(sno1);
	int b=Integer.parseInt(sno2);
	int c=a/b;
%>
<%= "Division of two numbers is ="+c %>
```

error.jsp

```
<%@page isErrorPage="true" %>
<center>
	<b>
		Sorry! exception occured <br>
		<%= exception %>
		</b>
</center>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
		<welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```
Request url

```
		http://localhost:2525/JspApp4/
```

2)By using <error-page> element in web.xml file

This approach is recommanded to use because we don't need to add "errorPage" attribute in each jsp page.Defining single entry of <error-page> element in web.xml file will handle all types of exceptions.

Deployment Directory structure

```
JspApp4
|
|-----Java Resources
|
|-----Web Content
		|
		|---form.html
		|---process.jsp
		|---error.jsp
		|
		|---WEB-INF
				|
				|---web.xml
```
Note:

In above application we need to add "servlet-api.jar" file in project build path.

form.html

```
<form action="process.jsp">
		No1: <input type="text" name="t1"/> <br>
		No2: <input typ="text" name="t2"/> <br>
		<input type="submit" value="divide"/>
</form>
```
process.jsp

```
<%
		String sno1=request.getParameter("t1");
		String sno2=request.getParameter("t2");
```

```
            int a=Integer.parseInt(sno1);
            int b=Integer.parseInt(sno2);
            int c=a/b;
%>
<%= "Division of two numbers is ="+c %>
```

error.jsp

```
<%@page isErrorPage="true" %>
<center>
            <b>
                        Sorry! exception occured <br>
                        <%= exception %>
            </b>
</center>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <error-page>
            <exception-type>java.lang.Exception</exception-type>
            <location>/error.jsp</location>
  </error-page>
  <welcome-file-list>
            <welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Request url

```
            http://localhost:2525/JspApp4/
```

JSP to Database communication

==============================

Deployment Directory structure

```
JspApp5
|
|-----Java Resources
|
|-----Web Content
            |
            |---form.html
            |---process.jsp
            |
            |---WEB-INF
                        |
                        |---web.xml
                        |
                        |-------lib
                                    |
                              |---ojdbc14.jar
```

Note:

In above application we need to add "servlet-api.jar" and "ojdbc14.jar" file in project build path.
copy and paste "ojdbc14.jar" file inside "WEB-INF/lib" folder seperately.

form.html

```
<form action="process.jsp">
        No: <input type="text" name="t1"/> <br>
        Name: <input type="text" name="t2"/> <br>
        Address : <input type="text" name="t3"/> <br>
        <input type="submit" value="submit"/>
</form>
```

process.jsp

```
<%@page import="java.sql.*" buffer="8kb" contentType="text/html" language="java" %>
<%
        String sno=request.getParameter("t1");
        int no=Integer.parseInt(sno);
        String name=request.getParameter("t2");
        String address=request.getParameter("t3");

        Connection con=null;
        PreparedStatement ps=null;
        String qry=null;
        int result=0;
        try
        {
                Class.forName("oracle.jdbc.driver.OracleDriver");
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","admin");
                qry="insert into student values(?,?,?)";
                ps=con.prepareStatement(qry);
                //set the values
                ps.setInt(1,no);
                ps.setString(2,name);
                ps.setString(3,address);

                //execute
                result=ps.executeUpdate();
                if(result==0)
                        out.println("No Record inserted");
                else
                        out.println("Record inserted");

                ps.close();
                con.close();
        }
        catch(Exception e)
        {
                out.println(e);
        }
```

%>

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
        <welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Request url

----

        http://localhost:2525/JspApp5/

Action tags

==============

Action tags provides functionality along with servlet API features.
It contains xml tags.
It does not have any standard tags.
All action tags are executed dynamically at runtime.
Action tags are divided into two types.

1)Standard action tags
2)Custom action tags

1)Standard action tags

Built-In action tags are called standard action tags.
ex:

```
        <jsp:include>
        <jsp:forward>
        <jsp:useBean>
        <jsp:setProperty>
        <jsp:getProperty>
        and etc.
```

Action include

===================

In this case , output of source jsp program and output of destination jsp program goes to browser window as dynamic response.
It internal uses servlet api functionality called rd.include(req,res).
syntax:

```
        <jsp:include   page="page_name" />
```

Deployment Directory structure

--

```
JspApp6
|
|-----Java Resources
|
|-----Web Content
```

```
            |
            |---A.jsp
            |---B.jsp
            |
            |---WEB-INF
                        |
                        |---web.xml
```
Note:

In above application we need to add "servlet-api.jar" file in project build path.

A.jsp

```
<b><i>Begining of A.jsp program</i></b>
<br>
<jsp:include page="B.jsp"/>
<br>
<b><i>Ending of A.jsp program</i></b>
```
B.jsp

```
<br>
<b><i>This is  B.jsp program</i></b>
<br>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
            <welcome-file>A.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```
Request url
----

http://localhost:2525/JspApp6/

Action forward
===================
In this case , output of source jsp program will be discarded and output of destination jsp program goes to browser window as dynamic response.
It internal uses servlet api functionality called rd.forward(req,res).
syntax:
```
            <jsp:forward  page="page_name" />
```
Deployment Directory structure

```
JspApp6
|
|-----Java Resources
|
|-----Web Content
            |
            |---A.jsp
            |---B.jsp
            |
            |---WEB-INF
                        |
                        |---web.xml
```
Note:

In above application we need to add "servlet-api.jar" file in project build path.

A.jsp

```
<b><i>Begining of A.jsp program</i></b>
<br>
<jsp:forward page="B.jsp"/>
<br>
<b><i>Ending of A.jsp program</i></b>
```
B.jsp

```
<br>
<b><i>This is  B.jsp program</i></b>
<br>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
            <welcome-file>A.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```
Request url

```
            http://localhost:2525/JspApp6/
```

JSP to java bean communication
================================
JSP to java bean communication is possible by using three tags.

1)<useBean> tag

It is used to create and locate bean class object.

2)<setProperty> tag

It is used to set the value to bean object and calls setter method.

3)<getProperty> tag

It is used to get the value from bean object and calls getter method.

Note:

In all the above tags are independent tags.

Example:1

Deployment Directory structure

```
JspApp7
|
|-----Java Resources
            |
            |------src
                        |
                        |--com.ihub.www
                                    |
                                    |----CubeNumber.java
|
|-----Web Content
            |
            |---process.jsp
            |
            |---WEB-INF
                        |
                        |---web.xml
```

Note:

In above application we need to add "servlet-api.jar" file in project build path.

process.jsp

```
<jsp:useBean id="cn" class="com.ihub.www.CubeNumber"></jsp:useBean>
<%= "Cube of a given number is ="+cn.cube(5) %>
```

CubeNumber.java

```
package com.ihub.www;
public class CubeNumber
{
            public int cube(int n)
            {
                        return n*n*n;
            }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
            <welcome-file>process.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

request url

----

http://localhost:2525/JspApp7/

Example:2

Deployment Directory structure

--

```
JspApp8
|
|-----Java Resources
          |
          |------src
                    |
                    |--com.ihub.www
                              |
                              |----User.java
|
|-----Web Content
          |
          |---form.html
          |
          |---process.jsp
          |
          |---WEB-INF
                    |
                    |---web.xml
```

Note:

In above application we need to add "servlet-api.jar" file in project build path.

form.html

```
<form action="process.jsp">
	UserName: <input type="text" name="username"/> <br>
	Password: <input type="password" name="password"/> <br>
	Email : <input type="text" name="email"/> <br>
	<input type="submit" value="submit"/>
</form>
```

User.java

```
package com.ihub.www;
public class User
{
	private String username;
	private String password;
	private String email;

	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
	public String getEmail() {
		return email;
	}
	public void setEmail(String email) {
		this.email = email;
	}
}
```
process.jsp

```
<jsp:useBean id="u" class="com.ihub.www.User"></jsp:useBean>
<jsp:setProperty property="*" name="u"/>
Records are <br>
<jsp:getProperty property="username" name="u"/> <br>
<jsp:getProperty property="password" name="u"/> <br>
<jsp:getProperty property="email" name="u"/> <br>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
	<welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```
request url
----
	http://localhost:2525/JspApp8/

CLASS-3
Custom tags in JSP
===================
Tags which are created by the programmer or user based on the application required are called custom tags.
To create a custom tag in jsp we need to use taglib directive.
syntax:
```
	<%@taglib  uri="tldfilelocation"  prefix="prefix-name"  %>
```
To work with custom tags we need to add "jsp-api.jar" file in project build path.
Deployment Directory structure

```
JspApp9
|
|-------Java Resources
|			|
			|------src
```

```
                    |
                    |--com.ihub.www
                           |
                           |----CubeNumber.java
|
|-------Web Content
             |
             |------process.jsp
             |
             |------WEB-INF
             |      |
                    |--mytags.tld
                    |
                    |--web.xml
                    |
                    |-----lib
                           |
                           |---jsp-api.jar
```
Note:

In above application we need to add "servlet-api.jar" and "jsp-api.jar" in project build path.
Copy and paste "jsp-api.jar" file inside "Web Content/WEB-INF/lib" folder seperately.
Here tld stands for Tag library descriptor.

process.jsp

```
<%@taglib uri="/WEB-INF/mytags.tld" prefix="ihub" %>
Cube of a given number is =<ihub:cube  number="5" />
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
        <welcome-file>process.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```
mytags.tld

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE taglib
    PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
    "http://java.sun.com/j2ee/dtd/web-jsptaglibrary_1_2.dtd">
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>simple</short-name>
  <uri>mytags</uri>
  <description>A simple tab library for the examples</description>
  <tag>
        <name>cube</name>
        <tag-class>com.ihub.www.CubeNumber</tag-class>
        <attribute>
                <name>number</name>
                <required>true</required>
        </attribute>
  </tag>
</taglib>
```

CubeNumber.java

```
package com.ihub.www;

import java.io.IOException;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;

public class CubeNumber extends TagSupport
{
        private int number;

        //setter method
        public void setNumber(int number) {
                this.number = number;
        }
        public int doStartTag()throws JspException
        {
                JspWriter out=pageContext.getOut();
                try
                {
                        out.println(number*number*number);
                }
                catch(Exception e)
                {
                                try
                                {
                                        out.println(e);
                                }
                                catch (IOException ioe)
                                {
                                ioe.printStackTrace();
                                }
                }
                return SKIP_BODY;
        }
}
```

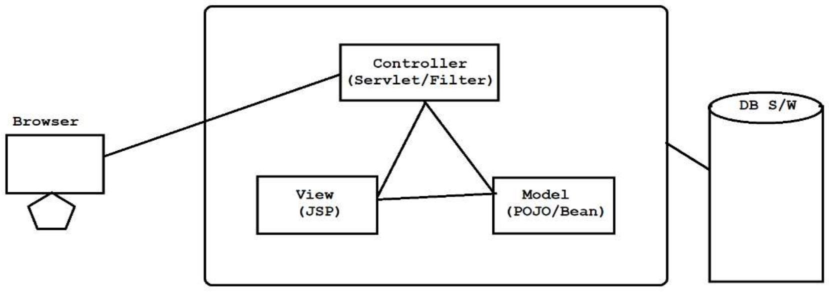
Request url

----

http://localhost:2525/JspApp9/

MVC in jsp

==============
MVC stands for Model View Controller.
It is one of the design pattern which seperates business logic,
persistence logic and data.
Controller is used to intercept all incoming request.
Model contains business logic or data.
View contains presentation logic i.e UI.



Deployment Directory structure

```
MVCApp
|
|-------Java Resources
|                  |
                   |------src
                            |
                            |--com.ihub.www
                                      |
                                      |----LoginBean.java
                                      |----LoginController.java
|
|-------Web Content
                   |
                   |------form.html
                   |------view.jsp
                   |------error.jsp
                   |
                   |------WEB-INF
                   |         |
                             |-------web.xml
                             |
```
Note:

In above application we need to add "servlet-api.jar" in project build path.

form.html

```
<form action="test">
        UserName: <input type="text" name="username"/> <br>
        Password: <input type="password" name="password"/> <br>
        <input type="submit" value="submit"/>
</form>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <servlet>
        <servlet-name>LoginController</servlet-name>
        <servlet-class>com.ihub.www.LoginController</servlet-class>
  </servlet>
  <servlet-mapping>
        <servlet-name>LoginController</servlet-name>
        <url-pattern>/test</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
        <welcome-file>form.html</welcome-file>
  </welcome-file-list>
</web-app>
```

LoginBean.java

```
package com.ihub.www;

public class LoginBean
{
        private String username;
        private String password;

        public String getUsername() {
                return username;
        }
        public void setUsername(String username) {
                this.username = username;
        }
        public String getPassword() {
                return password;
        }
        public void setPassword(String password) {
                this.password = password;
        }
}
```

LoginController.java

```
package com.ihub.www;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class LoginController extends HttpServlet
{
	protected void doGet(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
	{
		PrintWriter pw=res.getWriter();
		res.setContentType("text/html");

		//reading form data
		String uname=req.getParameter("username");
		String pass=req.getParameter("password");

		//set the values to bean object
		LoginBean lb=new LoginBean();
		lb.setUsername(uname);
		lb.setPassword(pass);
		//send the bean object to jsp
		req.setAttribute("bean",lb);
		if(pass.equals("admin"))
		{
			RequestDispatcher rd=req.getRequestDispatcher("view.jsp");
			rd.forward(req,res);
		}
		else
		{
			RequestDispatcher rd=req.getRequestDispatcher("error.jsp");
			rd.forward(req,res);
		}
		pw.close();
	}
}
```

view.jsp

```
<%@page  import="com.ihub.www.LoginBean" %>
<%
	LoginBean lb=(LoginBean)request.getAttribute("bean");
%>
<%= "UserName :"+lb.getUsername() %> <br>
<%= "Password :"+lb.getPassword() %> <br>
```

error.jsp

```
<b><i>
	<font color="red">
		Sorry! incorrect username and passord
		</font>
</i></b>
<%@include  file="form.html" %>
```

request url
----

	http://localhost:2525/MVCApp/

Jsp Implicit Objects

=====================
There are 9 implicit objects present in jsp.
Implicit objects are created by the web container that is available for every jsp program.
Object which can be used directly without any configuration is called implicit object.
The list of implicit objects are.

Object	Type
=======	======
out	JspWriter
request	HttpServletRequest
response	HttpServletResponse
config	ServletConfig
application	ServletContext
session	HttpSession
pageContext	pageContext
page	Object
exception	Throwable

response
=========
In jsp, response is a implicit object of type HttpServletResponse.
It can be used to add or manipulate response such as redirect response or another resources,send error and etc.

Deployment Directory structure
===========================
```
JspApp10
|
|------------Java Resources
|                   |
                    |----src
|
|------------Web Content
|                   |
                    |-----index.html
                    |-----welcome.jsp
                    |
                    |-----WEB-INF
                              |
                              |----web.xml
```
Note:

In above project, we need to add "servlet-api.jar" file in project build path.

index.html

```
<center>
        <a href="welcome.jsp">
                Facebook Login
        </a>
</center>
```

welcome.jsp

```
<%  response.sendRedirect("http://www.facebook.com/login");  %>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
  <welcome-file-list>
        <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

requesturl

----

	http://localhost:2525/JspApp9/

JSP config

===========

It is an implicit object of type ServletConfig.
The config object is created by web container for each jsp page.
This object is used to read initialized parameters for a perticular jsp page.

Deployment Directory structure

==========================

```
JspApp11
|
|---------Java Resources
|                   |
                    |-----src
|
|---------Web Content
                    |
                    |-----index.html
                    |-----process.jsp
                    |
                    |-----WEB-INF
                              |
                              |---web.xml
```

Note:

	In above project, we need add "servlet-api.jar" file in project build path.

index.html

```
<center>
        <a href="test">click</a>
</center>
```

process.jsp

```
<%
        out.println(config.getInitParameter("driver"));
```

```
%>
<br>
<%
	out.println(config.getInitParameter("url"));
%>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
	<servlet>
		<servlet-name>ABC</servlet-name>
		<jsp-file>/process.jsp</jsp-file>
		<init-param>
		<param-name>driver</param-name>
		<param-value>oracle.jdbc.driver.OracleDriver</param-value>
		</init-param>
	</servlet>
	<servlet-mapping>
		<servlet-name>ABC</servlet-name>
		<url-pattern>/test</url-pattern>
	</servlet-mapping>
	<Welcome-file-list>
		<welcome-file>index.html</welcome-file>
	</welcome-file-list>
</web-app>
```

Request url
----

	http://localhost:2525/JspApp10/test

jsp application
==================
In jsp, application is an implicit object of type ServletContext.
This instance of ServletContext is created only once by the web container.
This object is used to read initialized parameters from configuration file web.xml file.

Deployment Directory structure
==========================

```
JspApp11
|
|---------Java Resources
|                  |
                   |-----src
|
|---------Web Content
                   |
                   |-----index.html
                   |-----process.jsp
                   |
                   |-----WEB-INF
```

```
                                    |
                                    |---web.xml
```
Note:

In above project, we need add "servlet-api.jar" file in project build path.

index.html

```
<center>
	<a href="process.jsp">click</a>
</center>
```
process.jsp

```
<%
	out.println(application.getInitParameter("driver"));
%>
<br>
<%
	out.println(application.getInitParameter("url"));
%>
```
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
	<servlet>
		<servlet-name>ABC</servlet-name>
		<jsp-file>/process.jsp</jsp-file>
	</servlet>
	<servlet-mapping>
		<servlet-name>ABC</servlet-name>
		<url-pattern>/test</url-pattern>
	</servlet-mapping>

	<context-param>
		<param-name>driver</param-name>
		<param-value>com.mysql.driver.Driver</param-value>
	</context-param>
	<welcome-file-list>
		<welcome-file>index.html</welcome-file>
	</welcome-file-list>
</web-app>
```
Request url

```
	http://localhost:2525/JspApp10/test
```

JSP session
============

In JSP, session is an implicit object of type HttpSession.
It is used to get or set the session formation.

Deployment Directory structure

```
==========================
JspApp12
|
|----------Java Resources
                    |
                    |----src
                    |
|
|----------WEB Content
                    |
                    |---index.html
                    |
                    |---welcome.jsp
                    |
                    |---second.jsp
                    |
                    |----WEB-INF
                              |
                              |---web.xml
```

Note:

In above project we need to add "servlet-api.jar" file in project build path.

index.html

```
<form action="welcome.jsp">
	Name :<input type="text" name="t1"/>
	<br>
	<input type="submit" value="submit"/>
</form>
```

welcome.jsp

```
<%
	String name=request.getParameter("t1");
	out.println("Welcome ="+name);
	session.setAttribute("pname",name);
%>
<a href="second.jsp">goto second.jsp</a>
```

second.jsp

```
<%
	String name=(String)session.getAttribute("pname");
	out.println("Hello ="+name);
%>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
 <welcome-file-list>
	<welcome-file>index.html</welcome-file>
 </welcome-file-list>
</web-app>
```
request url
----

    http://localhost:2525/JspApp11/

JSP pageContext
==============

In jsp, pageContext is an implicit object of type pageContext class.
The pageContext object can be used to set ,get ,remove attributes from one the following scopes.

page
request
session
application
In JSP, page scope is a default scope.
Deployment Directory structure
==========================
```
JspApp13
|
|----------Java Resources
                    |
                    |----src
                    |
|
|----------WEB Content
                    |
                    |---index.html
                    |
                    |---welcome.jsp
                    |
                    |---second.jsp
                    |
                    |----WEB-INF
                            |
                            |---web.xml
```
Note:

    In above project we need to add "servlet-api.jar" file in project build path.

index.html

```
<form action="welcome.jsp">
	Name :<input type="text" name="t1"/>
	<br>
	<input type="submit" value="submit"/>
</form>
```
welcome.jsp

```
<%
	String name=request.getParameter("t1");
	out.println("Welcome ="+name);
	pageContext.setAttribute("pname",name,pageContext.SESSION_SCOPE);
%>
<a href="second.jsp">goto second.jsp</a>
```

second.jsp

```
<%
	String name=(String)pageContext.getAttribute("pname",pageContext.SESSION_SCOPE);
	out.println("Hello ="+name);
%>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">
 <welcome-file-list>
	<welcome-file>index.html</welcome-file>
 </welcome-file-list>
</web-app>
```

request url

```
	http://localhost:2525/JspApp12/
```