Dt : 27/2/2025

(d)<jsp:useBean>

(e)<jsp:setProperty>

(f)<jsp:getProperty>

(d)<jsp:useBean>:

=><jsp:useBean> is used to instantiate bean object or access the existing bean object.

syntax:

<jsp:useBean id="ub" class="test.UserBean" scope="session"/>

<jsp:useBean id="ub" type="test.UserBean"></jsp:useBean>

(e)<jsp:setProperty>:

=><jsp:setProperty> is used to load the data to bean Object.

syntax:

<jsp:useBean property="pro-name" param="para-name" name="bean-name">

(f)<jsp:getProperty>:

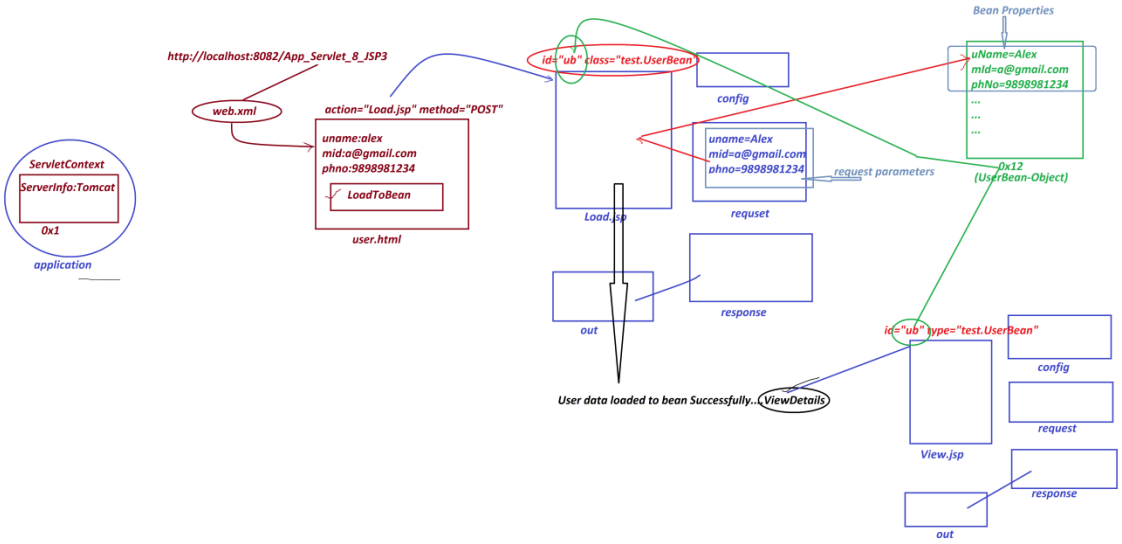
=><jsp:getProperty> is used to get the data from bean Object.

syntax:

=><jsp:getProperty property="pro-name" name="bean-name">

Ex:(Demonstrating <jsp:useBean>,<jsp:setProperty> and <jsp:getProperty>)

Layout:



user.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="Load.jsp" method="post">
UserName:<input type="text" name="uname"><br>
MailId:<input type="text" name="mid"><br>
PhoneNO:<input type="text" name="phno"><br>
<input type="submit" value="LoadToBean">
</form>
</body>
</html>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app>
   <welcome-file-list>
      <welcome-file>user.html</welcome-file>
```

```
   </welcome-file-list>
</web-app>
```

UserBean.java

```
package test;
import java.io.*;
@SuppressWarnings("serial")
public class UserBean implements Serializable
{
    private String uName,mId;
    private long phNo;
    public UserBean() {}
      public String getuName() {
            return uName;
      }
      public void setuName(String uName) {
            this.uName = uName;
      }
      public String getmId() {
            return mId;
      }
      public void setmId(String mId) {
            this.mId = mId;
      }
      public long getPhNo() {
            return phNo;
      }
      public void setPhNo(long phNo) {
            this.phNo = phNo;
      }

}
```

Load.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="test.UserBean"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<jsp:useBean id="ub" class="test.UserBean" scope="session"/>
<jsp:setProperty property="uName" param="uname" name="ub"/>
```

```
<jsp:setProperty property="mId" param="mid" name="ub"/>
<jsp:setProperty property="phNo" param="phno" name="ub"/>
<%
out.println("User data loaded to bean Successfully....");
%>
<a href="View.jsp">ViewDetails</a>
</body>
</html>
```

View.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"
    import="test.UserBean"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<jsp:useBean id="ub" type="test.UserBean" scope="session"/>
UserName:<jsp:getProperty property="uName" name="ub"/><br>
MailId:<jsp:getProperty property="mId" name="ub"/><br>
PhoneNo:<jsp:getProperty property="phNo" name="ub"/>
</body>
</html>
```

==========

**imp*

Expression Language(EL):

=>Expression Language(EL) introduced into JSP to make the code more simple and can access all the implicit objects of JSP.

syntax:

$(Expression)

=>The folowing are the implicit objects of EL:

1.applicationScope

2.sessionScope

3.requestScope

4.pageScope

5.param

6.paramValues

7.header

8.headerValues

9.cookie

10.initParam

11.pageContext

1.applicationScope:

=>applicationScope is used to access the attribute from ServletContext.

2.sessionScope:

=>sessionScope is used to access the attribute from HttpSession.

3.requestScope:

=>requestScope is used to access the attribute from request-object.

4.pageScope:

=>pageScope is used to access the attribute from PageContext Object.

5.param:

=>param is used to access single parameter-value from the request-Object

6.paramValues:

=>paramValues is used to access multiple parameter-values from the request object

7.header:

=>header is used retrive HTTP protocol header name.

8.headerValues:

=>headerValues is used retrieve HTTP protocol multiple header names.

9.cookie:

=>cookie is used to get the cookies from request-object.

10.initParam:

=>initParam is used to access the initilization parameter-values from config object.

11.pageContext:

=>pageContext is implicit name used by JSP-EL to access PageContext-object

---

Summary of Objects Generated from CoreJava:

1.User defined Class Objects

2.String-Objects

3.WrapperClass-Objects

4.Array-Objects

5.Collection<E>-Objects

6.Map<K,V>-Objects

7.Enum<E>-Objects

Summary of Objects generated from JDBC

1.Connection Object

2.Statement Object

3.PreparedStatement Object

4.CallableStatement Object

5.ResultSet Object

(i)Scrollable ResultSet Object

(ii)NonScrollable ResultSet Object

6.RowSet Object

(i)JdbcRowSet Object

(ii)CachedRowSet Object

7.DatabaseMetaData Object

8.ParameterMetaData Object

9.ResultSetMetaData Object

10.RowSetMetaData Object

Summary of Objects generated from Servlet Programming:

1.ServletContext Object

2.ServletConfig Object

3.ServletRequest/HttpServletRequest Object

4.ServletResponse/HttpServletResponse Object

5.PrintWriter Object

6.HttpSession Object

7.Cookie Object

8.Bean Object

9.JCF Object

10.DAO Object

Summary of implicit Objects generated from JSP programming:

1.application

2.config

3.request

4.response

5.out

6.session

7.exception

8.page

9.pageContext

JSP-EL:(Implicit Objects)

1.applicationScope

2.sessionScope

3.requestScope

4.pageScope

5.param

6.paramValues

7.header

8.headerValues

9.cookie

10.initParam

11.pageContext

==========

define JSTL?

=>JSTL stands for 'JSP Standard Tag Library' and,which provide some tags to make code more simple.

=>The following are some important JSTL tags:

1.Core Tags

2.Function Tags

3.Formatting Tags

4.XML Tags

5.SQL Tags

1.Core Tags:

=>The fundamental tags used for writing basic codes are known as CoreTags.

2.Function Tags:

=>The tags which support String functions are known as Function Tags.

3.Formatting Tags:

=>The tags which are used to format the data for persentation are known as Formatting tags.

4.XML Tags:

=>The tags which are used to write XML code in JSP are known as XML tags.

5.SQL Tags:

=>The tags which are used to write DB related Code in JSP are known as SQL Tags.

Note:

=>According to realtime application development using MVC,'SQL-Tags' and 'XML-Tags' are deprecated.

=>"@taglib" directive tag is used to include JSTL libraries into Current Running Program.

==========