Dt : 19/11/2024(Day-44)

Ex-3:

(Construct JSP Application to demonstrate <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("data loaded to bean object 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"/><br>
</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

---

****imp***

Summary of Objects generated:

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

JDBC:

1.Connection Object

2.Statement Object

3.PreparedStatement Object

4.CallableStatement Object

5.ResultSet Object

6.DatabaseMetaData Object

7.ParameterMetaData Object

8.ResultSetMetaData Object

Servlet:

1.ServletContext Object

2.ServletConfig Object

3.ServletRequest/HttpServletRequest Object

4.ServletResponse/HttpServletResponse Object

5.PrintWriter Object(for Servlet response)

6.HttpSession Object

7.Cookie Object

8.Bean Object

9.DAO Object

10.JCF Object

JSP:(Implicit Objects)

1.application

2.session

3.config

4.request

5.response

6.out

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

Diagram:

==========

Dt : 20/11/2024(Day-45)

define JSTL?

=>JSTL stands for 'JSP Standard Tag Librarie' 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.

==========

Ex:(EL-JSTL Application)

step-1 : Download "jstl-1.2.jar" file from Internet Source.

step-2 : Copy the JSTL jar file into "lib" folder of JSP.

step-3 : Type and execute the program

Ex:

input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="post" action="CoreTags.jsp">
 Enter the String:<input type="text" name="str"><br>
 <input type="submit" value="Display">
 </form>
</body>
</html>
```

CoreTags.jsp

```
<%@ page language="java"  contentType="text/html; charset=ISO-8859-1"
```

```
pageEncoding="ISO-8859-1"%>

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

<!DOCTYPE html> <html> <head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

<c:set var="n" value="${param.str}" />

<c:forEach var="j" begin="1" end="5">

<c:out value="${n}"/><p>

</c:forEach>

<c:forTokens items="${param.str}" delims=" " var="name">

<c:out value="${name}"/><p>

</c:forTokens>

</body>

</html>
```

web.xml

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

<web-app>

  <welcome-file-list>

    <welcome-file>input.html</welcome-file>

  </welcome-file-list>

</web-app>
```

==========

faq:

define 'page'?

=>'page' is an implicit object of JSP and which is generated from

'java.lang.Object' class

=>'page' is used in JSP to perform Cloning-process and Thread-Communication

process.

==========

faq:

define 'pageContext'?

=>'pageContext' is an implicit object of JSP and which is generated from

'PageContext' abstractClass.

=>This 'pageContext' object can access all other implicit JSP objects directly.

=>The following are some important methods of 'pageContext':

```
public abstract jakarta.servlet.ServletContext getServletContext();
public abstract jakarta.servlet.ServletConfig getServletConfig();
public abstract jakarta.servlet.http.HttpSession getSession();
public abstract jakarta.servlet.ServletRequest getRequest();
public abstract jakarta.servlet.ServletResponse getResponse();
public abstract jakarta.servlet.jsp.JspWriter getOut();
public abstract java.lang.Exception getException();
public abstract java.lang.Object getPage();
```

==========

faq:

JSP Life-Cycle:

=>JSP Life-Cycle demonstrates different states(stages) of JSP program from starting to ending.

=>The following are the stages of JSP program:

1. *Translation process*
2. *Compilation process*
3. *Loading process*
4. *Instantiation process*
5. *Initialization process*
6. *Request Handling Process*
7. *Destroying process*

1.Translation process:

=>The process of separing java-code from JSP-Program is known as Translation process.

2.Compilation process:

=>The process of compiling the Java-code and generating CompiledCode is known as Compilation process.

(After Compilation Process the generated SourceCode will be destroyed automatically)

3.Loading process:

=>The process of loading JSP-Program for execution is known as Loading process.

4.Instantiation process:

=>when the JSP program loaded for execution it is automatically instantiated known as Instantiation process.

Note:

=>After Instantiation process,the execution controls will identify the following LifeCycle methods:

(i)_jspInit()

(ii)_jspService()

(iii)_jspDestroy()

5.Initialization process:

=>The process of making the programming components ready for execution is known as Initialization process.

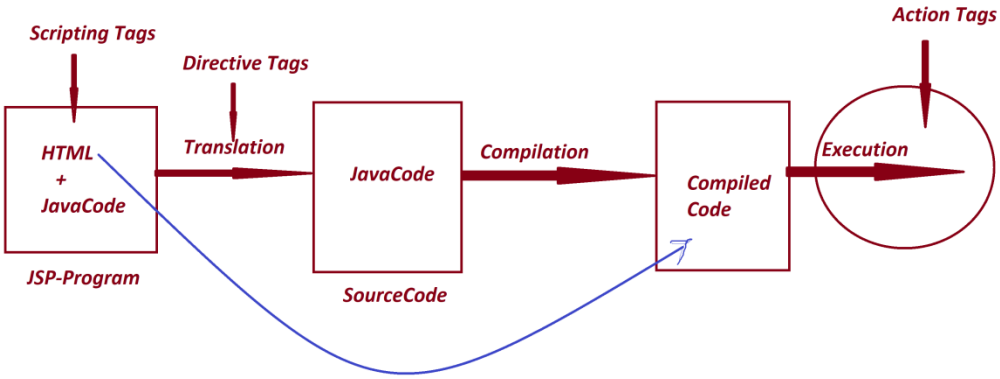
6.Request Handling Process:

=>The process of accepting the request and providing the response is known as Request Handling Process.

7.Destroying process:

=>The process of closing the opened resources and services is known as Destroying Process.

Diagram:



==========