ICON OF JAVA

ADV. JAVA

JSP

12. JSP Expression Language

Mr. Nagoor Babu M.Tech

[ICON of JAVA]
(Sun certified & Realtime Expert)

Ex. HCL Employee

Trained Lakhs of Students
for last 14 years across INDIA

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

JSP Expression Language

In JSP technology,scripting elements will allow java code inside the JSP pages,which is against to JSP principles.

To eliminate java code and scripting elements from JSP pages,we have to use standard actions,but which are in limited number with bounded functionality.

We will use custom actions,but it should require lot of java code internally inorder to implement simple programming constructs like if,switch,for and so on.

To overcome the above problems,we are able to use JSTL tag library,but still it is not sufficient to eliminate java code completely from JSP pages.

In the above context,to eliminate java code completely from JSP pages,we have to use Expression language syntax along with standard actions,custom actions and JSTL tag library.

Eg:

To get a particular request parameter from request object,we have to use java code before expression language.

<%

uname=request.getParameter("uname");

%>

In the above context,we are able to retrieve a particular request parameter without using java code, by using expression language syntax.

Ex:

${param.uname}

To eliminate java code completely from JSP pages,by using the following expression language elements.

1)EL operators

2)EL implicit objects.

3)EL functions.

1)EL operators:

To prepare expressions in expression language,we have to use the following operators.

-->**General purpose operators:**

.,(),{],[]

-->**Arithmetic Operators:**

+ - * / % and so on

-->**Comparison Operators:**

== or eq

!= or ne

< or lt

> or gt

<= or le

>= or ge

-->**Ternary Operators:**

expr1?expr2:expr3;

-->**Logical Operators:**

&& ||

2)EL implicit objects:

-->**Scope related:**

PageScope

requestScope

sessionScope

applicationScope

param

paramValues

initParam --->to get context parameters

cookie

header --->to get particular header value

headerValues

pageContext

In JSP technology,we will use JSP implicit objects in scripting elements to reduce java codeinside the scripting elements.

Similarly,Expression Language has provided the following list of implicit objects to

eliminate Java code from JSP pages.

pageScope,requestScope,sessionScope,applicationScope,param,paramValues,

initParam,cookie,header,headerValues,pageContext.

To access the attributes data from the page scope,request scope,session scope and application scope,we will use the above scope related implicit objects like pageScope,requestScope,sessionScope and applicationScope implicit objects respectively.

Ex:

<%

request.setAttribute("uname","Laddu");

%>

<html>

<body>

${requestScope.uname}

</body>

</html>

DURGASOFT, # 202,2ⁿᵈFloor,HUDAMaitrivanam,Ameerpet, Hyderabad - 500038, ☎ 040 – 64 51 27 86, 80 96 96 96 96, 9246212143 | www.durgasoft.com

param:

This can be used to access a particular request parameter.

paramValues:

This implicit object can be used to access more than one parameter values associated with a particular request parameter.

Ex:

form.html:

```
<html>

<body>

<center><b><font size="6">

<form method="get" action="first.jsp">

<br><br>

Name <input type="text" name="uname"/><br><br>

Food items <select size="3" multiple="true" name="food">

<option value="Dosa">Dosa</option>

<option value=="Vada">Vada</option>

<option value="Idly">Idly</option>

</select><input type="submit" value="Display"/>

</form></font></b></center>

</body>

</html>
```

first.jsp:

```
<html>

<body>
```

```
<center><b><font size="7"><br><br>

User Name.....${param.uname}

<br><br>

Food Items...<br>

${paramValues.food[0]}<br>

${paramValues.food[1]}<br>

${paramValues.food[2]}<br>

</font></b></center>

</body>

</html>
```


initParam

This implicit object can be used to access context parameters from ServletContex object.

Ex:

Web.xml:

```
<web-app>

<context-param>

<param-name>a</param-name>
```

```
            <param-name>AAA</param-name>

        </context-param>

              <context-param>

            <param-name>b/param-name>

            <param-name>BBB</param-name>

        </context-param>

        </web-app>


first.jsp:

<html>

<body>

<center><b><font size="7">

a--------->${initParam.a}<br>

b--------->${initParam.b}

</font></b></center>

</body>

</html>
```

cookie:

This implicit object can be used to get session id cookie name and session id cookie value.

```
Ex:

<html>

<body>

${cookie.JSESSIONID.name}<br>
```

```
${cookie.JSESSIONID.value}

</body>

</html>
```

header:

This implicit object can be used to access a particular request header value

Ex:

```
${header.cookie}
```

headerValues:

This implicit object can be used to access more than one value associated with a

single request header.

Ex:

```
${headerValues.accept[0]}
```

pageContext:

To get all the implicit objects and to get the values from either of the scopes like

pageScope,requestScope,sessionScope,applicationScope we will use pageContext implicit object.

Ex:

${pageContext.servletContext.servletInfo}

Expression Language Functions:

The main purpose of language functions is to perform a particular action in web applications.

In web applications we will utilise expression language functions as an alternative to JSP custom actions.

To prepare functions in expression language,we have to use the following steps:

1)Define a function by taking Java class under classes floder:

In expression language functions,we have to define a java method as a static method with the required implementation.

Ex:

public class Hello

{

public static String sayHello(String name)

{

return "Good Morning..."+name;

}

}

2)Configure expression language function in TLD file:

<taglib>

<function>

<name>function_name</name>

<function_class>Class_name</function_class>

<function_signature>method_signature</function_signature>

</function>

</taglib>

3)Access the function from JSP page:

To access the function in JSP page,we have to use the following syntax:

${fn:function_name([param_list])}

Ex:

${fn:sayHello("Nag")}

Ex:

firstapp

|

|---------first.jsp

|

|---------WEB-INF

| |

| |----hello.tld

| |----classes

```
|                    |                |---com.dss.Hello.class
|                    |                |
|
```

Hello.class

package com.dss;

public class Hello

{

public static String sayHello(String name)

{

return name;

}

}

hello.tld:

<taglib>

<jsp-version>2.1</jsp-version>

<tlib-version>1.0</tlib-version>

<function>

<name>sayHello</name>

<function-class>com.dss.Hello</function-class>

<function-signature>java.lang.String sayHello(java.lang.String)</function-signature>

</function>

</taglib>

first.jsp:

```
<%taglib uri="/WEB-INF/hello.tld" prefix="fn"%>

<html>

<body>

<b><fint size="7"><br><br>

${fn:sayHello("durga")}

</font></b>

</body>

</html>
```