

Java Standard Tag Library(JSTL)

SUN people encapsulated the core functionality, which is common to many web application in the form of JSTL. Programmer can use this predefined library without writing on his own.

The main objective of EL is to remove java code from JSP, but it fails to replace java code which process some functionality. This problem we can overcome through "JSTL".

The main theme of JSTL is to remove java code from JSP
JSP = JSTL + EL + HTML.

JSTL library is divided into 5 parts

1. Core library
2. SQL library
3. Functional library
4. Formatting library
5. XML library (deprecated becoz of xml syntax)

Core Library

=====

It is divided into 4 parts based on functionality

1. General Purpose Tags
2. Conditional Tags
3. Iteration Tags
4. URL Related Tags

Installing JSTL

By default JSTL functionality is not available in jsp pages, we can provide JSTL functionality by placing the following jar files in webapplication folder.

Tomcat server will give the implementation in 2 jars for JSTL API, these jars are available in the following location.

location :: C:\Tomcat 9.0\webapps\examples\WEB-INF\lib

Note:

Copy past the jars from that location to our project (./WEB-INF/lib) to make it available to container

1. General Purpose Tags

a. <c:out>

It is used to write the template text data and expression to jsp.
syntax: <c:out var='' value='' default=''/>

b. <c:set>

It is used to set the attribute in any scope.(default scope is 'page')
syntax: <c:set var='' value='' scope=''/>

c. <c:remove>

It is used to remove the attributes from the particular scope.
syntax: <c:remove var='' scope=''/>

If we don't specify the scope, by default the searching for an attribute would be done in the following order

a. page b. request c. session d. application

d. <c:catch>

This tag can be used to catch an exception within jsp instead of forwarding to error page.

The risky code we have to place inside the body of <c:catch>.

syntax::

```
<c:catch>
    Risky code
</c:catch>
```

If the exception raised in the risky code, this tag suppress that exception object and rest of jsp is executed normally.

If an exception is raised we can hold that exception object using attribute called 'var' which is page scope.

refer: core1.jsp,core2.jsp,core3.jsp,core4.jsp

EL operators

=====

1. Arithmetic operators

EL does not support operator overloading, so there is no String concatenation operator.

'+' always acts as addition operator.

eg::

```
${2+3} =====> 5
${null + 3} ==> 3
${""+3} ==> 3
${iNeuron+3} ==> 3
${"iNeuron"+3} ==> NFE
```

Note: In Arithmetic operators null is treated as "0".

EmptyString are also treated as "0".

2. - operator

3. * operator

The rules are exactly same as "+" operator.

4. / operator

The rules are exactly same as '+' operator only, but it follows Floating point arithmetic operation.

```
${10/2} ==> 5.0
${10 div 2} ==> 5.0
${10/0} ==> Infinity
${0/0} ==> NaN(Not a Number)
```

Empty Operator

=====

\${empty Object}

```
true => if object is an empty String
        if object is an empty array
        if object is an empty Collection
        if object doesnot exist
otherwise returns false.
```

EL reserve words

=====

```
true, false,null,empty,instanceof
gt, lt, ge, le, ne, eq, and, or, mod, div, not
```

Note:

In EL null behaves nicely

1. In arithmetic operators null is treated as zero.
2. In String evaluation null is treated as empty String.
3. In Logical operator null is treated as false.

eg:

```
${10 + null } ==> 10
```

```
${empty null } ==> true
```

```
${!null} -----> true
```

Conditional tags

=====

1. <c:if>

```
<c:if test ="test_condition" var="" scope=""/>
```

```
<c:if test = "test_condition" var="" scope="">
```

body of the if statement(will be evaluated only is

test_condition is true)

```
</c:if>
```

JSTL tags for if-else, switch statement can be executed using the below tags.

2. <c:choose>
3. <c:when>
4. <c:otherwise>

Note:

1. <c:choose> should compulsorily have atleast one <c:when>,but <c:otherwise> is optional.

2. <c:when> tag implicitly contains break statement, so fall through will not happen.

3. <c:otherwise> should be at last only.

4. <c:choose> <c:otherwise> won't take any attribute where as <c:when> takes only one attribute called "test".

Iteration tags

=====

1. <c:forEach> ---> It can be made to work like normal for loop and also foreach loop.

2. <c:forTokens> ---> It is a specialized foreach loop which works with StringTokenizer object.

URL Related tags

=====

1. <c:import>
2. <c:url>
3. <c:redirect>
4. <c:param>