Content Menu ▼

BUILD ENTERPRISE APPLICATIONS FASTER.
Visual design. Advanced scaffolding. Open source.
CUBA.platform

JSTL (JSP Standard Tag Library)

The JSP Standard Tag Library (JSTL) represents a set of tags to simplif the JSP development.

Advantage of JSTL

1. **Fast Developement** JSTL provides many tags that simplifies th JSP.
2. **Code Reusability** We can use the JSTL tags in various pages.
3. **No need to use scriptlet tag** It avoids the use of scriptlet tag.

There JSTL mainly provides 5 types of tags:

Tag Name	Description
core tags	The JSTL core tag provide variable support, UR management, flow control etc. The url for the core tag i **http://java.sun.com/jsp/jstl/core** . The prefix of core tag is **c**.
sql tags	The JSTL sql tags provide SQL support. The ur for the sql tags i **http://java.sun.com/jsp/jstl/sql** and prefi is **sql**.
xml tags	The xml sql tags provide flow control transformation etc. The url for the xml tags i **http://java.sun.com/jsp/jstl/xml** and prefix is **x**.
internationalization tags	The internationalization tags provide support for message formatting, number and date formatting etc. The url for the internationalization tags is **http://java.sun.com/jsp/jstl/fmt** and prefix is **fmt.**
functions tags	The functions tags provide support for string

	manipulation and string length. The url for the functions tags is **http://java.sun.com/jsp/jstl/functions** and prefix is **fn**.

For creating JSTL application, you need to load jstl.jar file.

download jstl1.2.jar file

JSTL Core Tags

The JSTL core tags mainly provides 4 types of tags:

- **miscellaneous tags**: catch and out.
- **url management tags**: import, redirect and url.
- **variable support tags**: remove and set.
- **flow control tags**: forEach, forTokens, if and choose.

Syntax for defining core tags

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

c:catch

It is an alternative apporach of global exception handling of JSP. handles the exception and doesn't propagate the exception to error page. The exception object thrown at runtime is stored in a variable named **var**.

Let's see the simple example of c:catch.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:catch>
int a=10/0;
</c:catch>
```

c:out

It is just like JSP expression tag but it is used for exression. It renders data to the page.

Let's see the simple example of c:out.

index.jsp

```
<form action="process.jsp" method="post">
FirstName:<input type="text" name="fname"/><br/>
LastName:<input type="text" name="lname"/><br/>
<input type="submit" value="submit"/>
</form>
```

process.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
First Name:<c:out value="${param.fname}"></c:out><br/>
Last Name:<c:out value="${param.lname}"></c:out>
```

download this example

c:import

It is just like jsp include but it can include the content of any resource either within server or outside the server.

Let's see the simple example of c:import to display the content of other site.

index.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<h1>ABC.com</h1>
<hr/>
<c:import url="http://www.javatpoint.com"></c:import>
```

Example of c:import to display the source code

Let's see the simple example of c:import to display the source code of other site.

index.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<h1>ABC.com</h1>
<hr/>
<c:import var="data" url="http://www.javatpoint.com"></c:import>
```

```
<h2>Data is:</h2>
<c:out value="${data}"></c:out>
```

download this example

c:forEach

It repeats the nested body content for fixed number of times or over collection.

Let's see the simple example of c:forEach.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:forEach var="number" begin="5" end="10">
<c:out value="${number}"></c:out>
</c:forEach>
```

download this example

c:if

It tests the condition. Let's see the simple example of c:if.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:set var="number" value="${200}">
<c:if test="${number<500}">
<c:out value="number is less than 500"></c:out>
</c:if>
```

c:redirect

It redirects the request to the given url.

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

c:choose, c:when and c:otherwise

The c:when and c:otherwise works like if-else statement. But it must be placed inside c:choose tag.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<h1>c:when, c:otherwise, c:choose</h1>
<c:set value="21" var="i"></c:set>
<c:choose>
<c:when test="${i%2==0}">
<c:out value="Even Number"></c:out>
</c:when>
<c:otherwise>
<c:out value="Odd Number"></c:out>
</c:otherwise>
</c:choose>
```

JSTL Function Tags

There are various functions available in JSTL. A list of JSTL functions are given below:

- toLowerCase()
- toUpperCase()
- trim()
- length()
- substring()
- startsWith()
- endsWith()
- replace()
- split()
- contains()
- indexOf()
- join()
- substringBefore()
- substringAfter()

fn:toUpperCase

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<h1>fn:toUpperCase</h1>
<c:set value="rahul kumar" var="name"></c:set>
<c:out value="${fn:toUpperCase(name)}"></c:out>
```

fn:length

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<h1>fn:length</h1>
<c:set value="rahul kumar" var="name"></c:set>
<c:out value="${fn:length(name)}"></c:out>
```

fn:substring

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<h1>fn:substring</h1>
<c:set value="abcdefgh" var="name"></c:set>
<c:out value="${fn:substring(name,2,5)}"></c:out>
```

← prev next →

Share 39