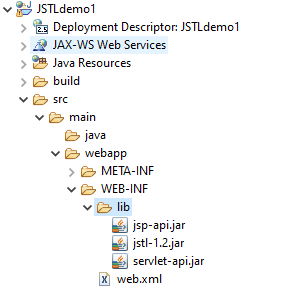
JSTL

Folder Structure



Step 1 :

Right click on Application 🡪 New 🡪 JSP file 🡪 Welcome.jsp

Step 2: add the core tag library to JSP file

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

Step 3: display the welcome message using c:out tag

<body>

<c:out value=*"Welcome to JSTL"*></c:out>

</body>

Step 4: Run the JSP file

Right click on application 🡪Run As 🡪 Run on Server

In the browser, type <http://localhost:8080/JSTLDemo1/welcome.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>

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"* rel=*"stylesheet"* integrity=*"sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"* crossorigin=*"anonymous"*>

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"* integrity=*"sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"* crossorigin=*"anonymous"*></script>

</head>

<body>

<div class=*"container"*>

<span class=*"badge bg-secondary"*><c:out value=*"Welcome to JSTL"*></c:out></span>

</div>

</body>

</html>

**JSTL Core tags**

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

There are following Core JSTL Tags:

|  |  |
| --- | --- |
| <c:out > | Like <%= ... >, but for expressions. |

<c:out value=*"Welcome to JSTL"*></c:out>

|  |  |
| --- | --- |
| <c:set > | Sets the result of an expression evaluation in a 'scope' |

<!-- used to set the attribute to the different scopes like

request, session or application

var -- Attribute Name

value -- Attribute Value

scope -- session or request or application

-->

<c:set var=*"sessA"* value=*"sessV"* scope=*"session"*></c:set>

<!-- How to retreive the value from session scope -->

<c:out value=*"*${sessionScope.sessA }*"*/>

|  |  |
| --- | --- |
| <c:remove > | Removes a scoped variable (from a particular scope, if specified). |

<c:set var=*"sessA"* value=*"sessV"* scope=*"session"*></c:set>

<!-- How to retreive the value from session scope -->

<c:out value=*"*${sessionScope.sessA }*"*/>

<!-- Removing the value from session scope -->

<c:remove var=*"sessA"* scope=*"session"*/>

<!-- How to retreive the value from session scope -->

<c:out value=*"*${sessionScope.sessA }*"*/>

|  |  |
| --- | --- |
| <c:if> | Simple conditional tag which evalutes its body if the supplied condition is true. |

<c:set var=*"number"* value=*"*${120}*"*/>

<c:if test=*"*${number<500}*"*>

<c:out value=*"number is less than 500"*></c:out>

</c:if>

|  |  |
| --- | --- |
| <c:choose> | Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise> |
| <c:when> | Subtag of <choose> that includes its body if its condition evalutes to 'true'. |
| <c:otherwise > | Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'. |

<!-- Switch case in JSTL -->

<c:set var=*"num"* value=*"*${2}*"*/>

<c:choose >

<c:when test=*"*${num==1 }*"*>

<c:out value=*"One"*/> <br>

</c:when>

<c:when test=*"*${num==2 }*"*>

<c:out value=*"Two"*/> <br>

</c:when>

<c:when test=*"*${num==3 }*"*>

<c:out value=*"Three"*/> <br>

</c:when>

<c:otherwise>

<c:out value=*"Invalid Number"*/>

</c:otherwise>

</c:choose>

|  |  |
| --- | --- |
| <c:import> | Retrieves an absolute or relative URL and exposes its contents to either the page, a String in 'var', or a Reader in 'varReader'. |

<button type=*"button"* class=*"btn btn-primary"*>

Notifications <span class=*"badge text-bg-secondary"*>4</span>

</button>

<div class=*"container text-bg-secondary"* bgcolor=*"yellow"*>

<c:import url=*"welcome.jsp"*/>

</div>

|  |  |
| --- | --- |
| <c:forEach > | The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality . |

<!-- it is used to execute the task for certain number of times -->

<c:forEach var=*"number"* begin=*"5"* end=*"10"*>

<span class=*"badge text-bg-secondary"*><c:out value=*"*${number}*"*/></span>

</c:forEach>

|  |  |
| --- | --- |
| <c:forTokens> | Iterates over tokens, separated by the supplied delimeters. |

<c:set var=*"strlist"* value=*"str1,str2,str3"*/>

<c:forTokens var=*"str"* items=*"*${strlist}*"* delims=*","*>

<span class=*"badge text-bg-secondary"*><c:out value=*"*${str }*"*/></span>

</c:forTokens>

|  |  |
| --- | --- |
| <c:redirect > | Redirects to a new URL. |

<c:redirect url=*"http://www.google.com"*>Click Here</c:redirect>

|  |  |
| --- | --- |
| <c:param> | Adds a parameter to a containing 'import' tag's URL. |
| <c:url> | Creates a URL with optional query parameters |

**SQL tags:**

<%@ taglib prefix="sql"

uri="http://java.sun.com/jsp/jstl/sql" %>

|  |  |
| --- | --- |
| Tag | **Description** |
| <sql:setDataSource> | Creates a simple DataSource suitable only for prototyping |
| <sql:query> | Executes the SQL query defined in its body or through the sql attribute. |
| <sql:update> | Executes the SQL update defined in its body or through the sql attribute. |
| <sql:param> | Sets a parameter in an SQL statement to the specified value. |
| <sql:dateParam> | Sets a parameter in an SQL statement to the specified java.util.Date value. |
| <sql:transaction > | Provides nested database action elements with a shared Connection, set up to execute all statements as one transaction. |

How to retrieve the data from Course table

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

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

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

<%@ taglib prefix=*"sql"*

uri=*"http://java.sun.com/jsp/jstl/sql"* %>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<sql:setDataSource var=*"snapshot"* driver=*"oracle.jdbc.driver.OracleDriver"*

url=*"jdbc:oracle:thin:@localhost:1521:orcl"*

user=*"advjava04"* password=*"advjava04"*/>

<sql:query dataSource=*"*${snapshot}*"* var=*"result"*>

SELECT \* from branch

</sql:query>

<table border=*"1"* width=*"100%"*>

<tr>

<th>Branch Id</th>

<th>Branch Name</th>

</tr>

<c:forEach var=*"row"* items=*"*${result.rows}*"*>

<tr>

<td><c:out value=*"*${row.bid}*"*/></td>

<td><c:out value=*"*${row.bname}*"*/></td>

</tr>

</c:forEach>

</table>

</body>

</html>

How to update the table

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

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

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

<%@ taglib prefix=*"sql"*

uri=*"http://java.sun.com/jsp/jstl/sql"* %>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<sql:setDataSource var=*"snapshot"* driver=*"oracle.jdbc.driver.OracleDriver"*

url=*"jdbc:oracle:thin:@localhost:1521:orcl"*

user=*"advjava04"* password=*"advjava04"*/>

<c:set var=*"bid"* value=*"4"*/>

<sql:update dataSource=*"*${snapshot}*"* var=*"count"*>

UPDATE branch SET bname = 'Civil' where bid=?

<sql:param value=*"*${bid}*"* />

</sql:update>

<sql:query dataSource=*"*${snapshot}*"* var=*"result"*>

SELECT \* from branch

</sql:query>

<h1> Record updated successfully </h1>

<table border=*"1"* width=*"100%"*>

<tr>

<th>Branch Id</th>

<th>Branch Name</th>

</tr>

<c:forEach var=*"row"* items=*"*${result.rows}*"*>

<tr>

<td><c:out value=*"*${row.bid}*"*/></td>

<td><c:out value=*"*${row.bname}*"*/></td>

</tr>

</c:forEach>

</table>

</body>

</html>

How to insert a new record to Course Table

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

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

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

<%@ taglib prefix=*"sql"*

uri=*"http://java.sun.com/jsp/jstl/sql"* %>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<sql:setDataSource var=*"snapshot"* driver=*"oracle.jdbc.driver.OracleDriver"*

url=*"jdbc:oracle:thin:@localhost:1521:orcl"*

user=*"advjava04"* password=*"advjava04"*/>

<sql:update dataSource=*"*${snapshot}*"* var=*"result"*>

INSERT INTO branch VALUES (7, 'AERO')

</sql:update>

<sql:query dataSource=*"*${snapshot}*"* var=*"result"*>

SELECT \* from branch

</sql:query>

<table border=*"1"* width=*"100%"*>

<tr>

<th>Branch Id</th>

<th>Branch Name</th>

</tr>

<c:forEach var=*"row"* items=*"*${result.rows}*"*>

<tr>

<td><c:out value=*"*${row.bid}*"*/></td>

<td><c:out value=*"*${row.bname}*"*/></td>

</tr>

</c:forEach>

</table>

</body>

</html>

How to delete the record from course table

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

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

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

<%@ taglib prefix=*"sql"*

uri=*"http://java.sun.com/jsp/jstl/sql"* %>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<sql:setDataSource var=*"snapshot"* driver=*"oracle.jdbc.driver.OracleDriver"*

url=*"jdbc:oracle:thin:@localhost:1521:orcl"*

user=*"advjava04"* password=*"advjava04"*/>

<c:set var=*"bid"* value=*"6"*/>

<sql:update dataSource=*"*${snapshot}*"* var=*"count"*>

DELETE FROM branch WHERE bid = ?

<sql:param value=*"*${bid}*"* />

</sql:update>

<sql:query dataSource=*"*${snapshot}*"* var=*"result"*>

SELECT \* from branch

</sql:query>

<table border=*"1"* width=*"100%"*>

<tr>

<th>Branch Id</th>

<th>Branch Name</th>

</tr>

<c:forEach var=*"row"* items=*"*${result.rows}*"*>

<tr>

<td><c:out value=*"*${row.bid}*"*/></td>

<td><c:out value=*"*${row.bname}*"*/></td>

</tr>

</c:forEach>

</table>

</body>

</html>

**JSTL Functions:**

JSTL includes a number of standard functions, most of which are common string manipulation functions. Following is the syntax to include JSTL Functions library in your JSP:

<%@ taglib prefix="fn"

uri="http://java.sun.com/jsp/jstl/functions" %>

Following is the list of JSTL Functions:

|  |  |
| --- | --- |
| Function | **Description** |
| fn:contains() | Tests if an input string contains the specified substring. |
| fn:containsIgnoreCase() | Tests if an input string contains the specified substring in a case insensitive way.  boolean containsIgnoreCase(java.lang.String, java.lang.String) |
| fn:endsWith() | Tests if an input string ends with the specified suffix.  boolean endsWith(java.lang.String, java.lang.String) |
| fn:indexOf() | Returns the index withing a string of the first occurrence of a specified substring.  int indexOf(java.lang.String, java.lang.String) |
| fn:join() | Joins all elements of an array into a string.  String join (java.lang.String[], java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:split(string1, ' ')}" />  <c:set var="string3" value="${fn:join(string2, '-')}" />  <p>Final String : ${string3}</p>  </body>  </html> |
| fn:length() | Returns the number of items in a collection, or the number of characters in a string.  int length(java.lang.Object)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="This is second String." />  <p>Length of String (1) : ${fn:length(string1)}</p>  <p>Length of String (2) : ${fn:length(string2)}</p>  </body>  </html> |
| fn:replace() | Returns a string resulting from replacing in an input string all occurrences with a given string.  boolean replace(java.lang.String, java.lang.String, java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:replace(string1,  'first', 'second')}" />  <p>Final String : ${string2}</p>  </body>  </html> |
| fn:split() | Splits a string into an array of substrings.  java.lang.String[] split(java.lang.String, java.lang.String) |
| fn:startsWith() | Tests if an input string starts with the specified prefix.  boolean startsWith(java.lang.String, java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string" value="Second: This is first String."/>  <c:if test="${fn:startsWith(string, 'First')}">  <p>String starts with First</p>  </c:if>  <br />  <c:if test="${fn:startsWith(string, 'Second')}">  <p>String starts with Second</p>  </c:if>  </body>  </html> |
| fn:substring() | Returns a subset of a string.  java.lang.String substring(java.lang.String, int, int)  **Example:**  Following is the example to explain the functionality of this function:  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:substring(string1, 5, 15)}" />  <p>Final sub string : ${string2}</p>  </body>  </html>  Output: is first S |
| fn:substringAfter() | Returns a subset of a string following a specific substring.  java.lang.String substringAfter(java.lang.String, java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:substringAfter(string1, 'is')}" />  <p>Final sub string : ${string2}</p>  </body>  </html>  OUTPUT: is first String. |
| fn:substringBefore() | Returns a subset of a string before a specific substring.  java.lang.String substringBefore(java.lang.String,  java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:substringBefore(string1,  'first')}" />  <p>Final sub string : ${string2}</p>  </body>  </html>  Final sub string : This is |
| fn:toLowerCase() | Converts all of the characters of a string to lower case.  java.lang.String toLowerCase(java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:toLowerCase(string1)}" />  <p>Final string : ${string2}</p>  </body>  </html>  Final string : this is first string. |
| fn:toUpperCase() | Converts all of the characters of a string to upper case.  java.lang.String tolowercase(java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String."/>  <c:set var="string2" value="${fn:toUpperCase(string1)}" />  <p>Final string : ${string2}</p>  </body>  </html>  Final string : THIS IS FIRST STRING. |
| fn:trim() | Removes white spaces from both ends of a string.  java.lang.String trim(java.lang.String)  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>  <html>  <head>  <title>Using JSTL Functions</title>  </head>  <body>  <c:set var="string1" value="This is first String "/>  <p>String (1) Length : ${fn:length(string1)}</p>  <c:set var="string2" value="${fn:trim(string1)}" />  <p>String (2) Length : ${fn:length(string2)}</p>  <p>Final string : ${string2}</p>  </body>  </html>  String (1) Length : 29  String (2) Length : 20  Final string : This is first String |

The fn:contains() function determines whether an input string contains a specified substring.

**Syntax:**

The fn:contains() function has following syntax:

boolean contains(java.lang.String, java.lang.String)

**Example:**

Following is the example to explain the functionality of this function:

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

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

<html>

<head>

<title>Using JSTL Functions</title>

</head>

<body>

<c:set var="theString" value="I am a test String"/>

<c:if test="${fn:contains(theString, 'test')}">

<p>Found test string<p>

</c:if>

<c:if test="${fn:contains(theString, 'TEST')}">

<p>Found TEST string<p>

</c:if>

</body>

</html>

This would produce following result:

Found test string

Example

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

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

<html>

<head>

<title>Using JSTL Functions</title>

</head>

<body>

<c:set var="string1" value="This is first String."/>

<c:set var="string2" value="This <abc> is second String.</abc>"/>

<p>Index (1) : ${fn:indexOf(string1, "first")}</p>

<p>Index (2) : ${fn:indexOf(string2, "second")}</p>

</body>

</html>

Formatting Tags

<fmt:formatNumber> To render numerical value with specific precision or format.

<fmt:parseNumber> Parses the string representation of a number, currency, or percentage.

<fmt:formatDate> Formats a date and/or time using the supplied styles and pattern.

<fmt:setLocale> Stores the given locale in the locale configuration variable.

<fmt:formatNumber> demo

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

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

<html>

<head>

<title>JSTL fmt:formatNumber Tag</title>

</head>

<body>

<h3>Number Format:</h3>

<c:set var = "balance" value = "120000.2309" />

<p>Formatted Number (1): <fmt:formatNumber value = "${balance}"

type = "currency"/></p>

<p>Formatted Number (2): <fmt:formatNumber type = "number"

maxIntegerDigits = "3" value = "${balance}" /></p>

<p>Formatted Number (3): <fmt:formatNumber type = "number"

maxFractionDigits = "3" value = "${balance}" /></p>

<p>Formatted Number (4): <fmt:formatNumber type = "number"

groupingUsed = "false" value = "${balance}" /></p>

<p>Formatted Number (5): <fmt:formatNumber type = "percent"

maxIntegerDigits="3" value = "${balance}" /></p>

<p>Formatted Number (6): <fmt:formatNumber type = "percent"

minFractionDigits = "10" value = "${balance}" /></p>

<p>Formatted Number (7): <fmt:formatNumber type = "percent"

maxIntegerDigits = "3" value = "${balance}" /></p>

<p>Formatted Number (8): <fmt:formatNumber type = "number"

pattern = "###.###E0" value = "${balance}" /></p>

<p>Currency in USA :

<fmt:setLocale value = "en\_US"/>

<fmt:formatNumber value = "${balance}" type = "currency"/>

</p>

</body>

</html>

Example – 2

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

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

<html>

<head>

<title>JSTL fmt:parseNumber Tag</title>

</head>

<body>

<h3>Number Parsing:</h3>

<c:set var = "balance" value = "1250003.350" />

<fmt:parseNumber var = "i" type = "number" value = "${balance}" />

<p>Parsed Number (1) : <c:out value = "${i}" /></p>

<fmt:parseNumber var = "i" integerOnly = "true"

type = "number" value = "${balance}" />

<p>Parsed Number (2) : <c:out value = "${i}" /></p>

</body>

</html>

fmt:dateNumber demo

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

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

<html>

<head>

<title>JSTL fmt:dateNumber Tag</title>

</head>

<body>

<h3>Number Format:</h3>

<c:set var = "now" value = "<% = new java.util.Date()%>" />

<p>Formatted Date (1): <fmt:formatDate type = "time"

value = "${now}" /></p>

<p>Formatted Date (2): <fmt:formatDate type = "date"

value = "${now}" /></p>

<p>Formatted Date (3): <fmt:formatDate type = "both"

value = "${now}" /></p>

<p>Formatted Date (4): <fmt:formatDate type = "both"

dateStyle = "short" timeStyle = "short" value = "${now}" /></p>

<p>Formatted Date (5): <fmt:formatDate type = "both"

dateStyle = "medium" timeStyle = "medium" value = "${now}" /></p>

<p>Formatted Date (6): <fmt:formatDate type = "both"

dateStyle = "long" timeStyle = "long" value = "${now}" /></p>

<p>Formatted Date (7): <fmt:formatDate pattern = "yyyy-MM-dd"

value = "${now}" /></p>

</body>

</html>

setLocale Demo

package com.sssit;

import java.util.ListResourceBundle;

public class Example\_es\_ES extends ListResourceBundle {

public Object[][] getContents() {

return contents;

}

static final Object[][] contents = {

{"count.one", " ఒకటి "},

{"count.two", " రెండు "},

{"count.three", " మూడు "},

};

}

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

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

<html>

<head>

<title>JSTL fmt:setLocale Tag</title>

</head>

<body>

<fmt:bundle basename = "com.sssit.Example">

<fmt:message key = "count.one"/><br/>

<fmt:message key = "count.two"/><br/>

<fmt:message key = "count.three"/><br/>

</fmt:bundle>

<!-- Change the Locale -->

<fmt:setLocale value = "es\_ES"/>

<fmt:bundle basename = "com.sssit.Example">

<fmt:message key = "count.one"/><br/>

<fmt:message key = "count.two"/><br/>

<fmt:message key = "count.three"/><br/>

</fmt:bundle>

</body>

</html>

Custom Tags:

Example 1

Step 1 ---- HelloTag.java

import javax.servlet.jsp.tagext.\*;

import javax.servlet.jsp.\*;

import java.io.\*;

public class HelloTag extends SimpleTagSupport {

public void doTag() throws JspException, IOException {

JspWriter out = getJspContext().getOut();

out.println("Hello Custom Tag!");

}

}

Step 2 ---- custom.tld(must be placed in web-inf folder)

<taglib>

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

<jsp-version>2.0</jsp-version>

<short-name>Example TLD</short-name>

<tag>

<name>Hello</name>

<tag-class>com.sssit.HelloTag</tag-class>

<body-content>empty</body-content>

</tag>

</taglib>

Step 3 ---- use in jsp page

<%@ taglib prefix="ex" uri="WEB-INF/custom.tld"%>

<html>

<head>

<title>A sample custom tag</title>

</head>

<body>

**<ex:Hello/>**

</body>

</html>

Output: Hello Custom Tag!