JSP Standart Tag Library-JSTL

Objectives

- This session is learning about the jstl:
  - What jstl is
  - Why do we use it
  - Tag libraries included in jstl
  - A detailed glance to the jstl-core library
  - The other libraries and their functions
  - How to use the jstl in a project

What JSTL is?

- **JSTL (JSP Standard Tag Libraries)** is a collection of JSP custom tags developed by Java Community Process, www.jcp.org.
- The goal of JSTL is to help simplify JavaServer Pages page authors' lives. To achieve this goal, JSTL has provided custom tags for many common JSP page authoring tasks that require scripting statements to manipulate server side dynamic data.

What JSTL offers?

- **General-purpose actions:** Displaying, scope, setting and removing jsp scoped attributes, catching exceptions.
- **Control flow actions:** Conditional, iterators…
- **Tag library validators:** TLVs allow projects to only allow specific tag libraries, as well as enforce JSP coding styles that are free of scripting elements

What JSTL offers? (2)

- **The other key aspects of JSTL are**:
  - Accessing URL-based resources
  - Internationalization (i18n) and text formatting
  - Relational database access (SQL)
  - XML processing
  - String manipulation

JSTL Tag Libraries

Functional Area	URI	Prefix
Core	http://java.sun.com/jsp/jstl/core	c
Xml Processing	http://java.sun.com/jsp/jstl/xml	x
I18N capable formatting	http://java.sun.com/jsp/jstl/fmt	fmt
Relational database accesing (SQL)	http://java.sun.com/jsp/jstl/sql	sql
Functions	http://java.sun.com/jsp/jstl/functions	fn

General Purpose Actions

- **Uri**=http://java.sun.com/jsp/jstl/core , **prefix**=c
  - **<c:set>**
    - **Set the value of a scoped variable using attribute value**

      ```
      <c:set value=”value” var=”varName”
      [scope=”{page|request|session|application}”]/>
      ```

    - **Set the value of a scoped variable using body content**

      ```
      <c:set var=”varName” [scope=”{page|request|…}”]>
      body content  </c:set>
      ```

    - **Set a property of a target object using attribute value**

      ```
      <c:set value=”value” target=”target”
      property=”propertyName”/>
      ```

    - **Set a property of a target object using body content**

      ```
      <c:set target=”target” property=”propertyName”>
      body content  </c:set>
      ```

    - **Set a deferred value**

      ```
      <c:set var=”varName” value="deferred-value"/>
      ```

General Purpose Actions (2)

- <c:remove>
  - The natural companion to <c:set>, allowing the explicit removal of scoped variables

```
<c:remove var="cachedResult" scope="application"/>
```

- <c:catch>
  - provides a complement to the JSP error page mechanism

```
<c:catch var=”exception”>
<!-- Execution we can recover from if exception
occurs -->
...
</c:catch>
<c:if test=”${exception != null}”>
Sorry. Processing could not be performed
because...
</c:if>
```

General Purpose Actions (3)

- <c:out>
  - Without a body

```
<c:out value=”value” escapeXml=”{true|false}”]
[default=”defaultValue”] />
```

  - With a body (jsp body)

```
<c:out value=”value” [escapeXml=”{true|false}”]>
default value </c:out>
```

Conditional Actions

- <c:if>
  - Without body content

```
<c:if test="testCondition" var="varName"
  [scope="{page|request|…}"]/>
```

  - With body content (jsp body)

```
<c:if test="testCondition" [var="varName"]
  [scope="{page|request|…}"]>
body content  </c:if>
```

Conditional Actions(2)

- <c:choose>

```
<c:choose>
body content (<when> and <otherwise> subtags)
</c:choose>
```

- The body of the <c:choose> action can only contain:

■ White spaces

May appear anywhere around the <c:when> and <c:otherwise> subtags.

■ 1 or more <c:when> actions

Must all appear before <c:otherwise>

■ 0 or 1 <c:otherwise> action

Must be the last action nested within <c:choose>

Conditional Actions(3)

- <c:when>
  - Represents an alternative within a <c:choose> action.

```
<c:when test="testCondition">
body content
</c:when>
```

■ Must have <c:choose> as an immediate parent.

■ Must appear before an <c:otherwise> action that has the same parent.

Conditional Actions(4)

- <c:otherwise>
  - Represents the last alternative within a <c:choose> action.

```
<c:otherwise>
conditional block
</c:otherwise>
```

■ Must have <c:choose> as an immediate parent.

■ Must be the last nested action within <c:choose>.

Iterator Actions

- <forEach>
  - Iterate over a collection of objects

```
<c:forEach[var=″varName″] items=″collection″
  [varStatus=″varStatusName″][begin=″begin″] [end=″end″]
  [step=″step″]>
body content </c:forEach>
```

  - Iterate a fixed number of times

```
<c:forEach [var=″varName″] [varStatus=″varStatusName″] begin=″begin″
  end=″end″ [step=″step″]>
body content </c:forEach>
```

Iterator Actions (1)

- Example 1 – iteration of a collection (Arraylist /vector/…)

```
<table>
 <c:forEach var=″product″ items=″${products}″
      varStatus=″status″>
  <tr><td>${status.count}″</td>
  <td>${product.name}″</td></tr>
 </c:forEach>
</table>
```

- Example 2 – iteration of a has map

```
<c:forEach var="entry"
 items="${myHashtable}">
Next element is ${entry.value}
</c:forEach>
```

Iterator Actions(2)

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

```
<c:forTokens items="stringOfTokens“
 delims="delimiters“ [var="varName"]
 [varStatus="varStatusName"] [begin="begin"]
 [end="end"] [step="step"]>
body content </c:forTokens>
```

URL Related Actions

- **URL**
  - **import a resource with an absolute URL**

```
<c:import
 url=”http://acme.com/exec/customers?country=Japan”/>
```

  - **import a resource with a relative URL - same context**

```
<c:import url=”/copyright.html”/>
```

  - **import a resource with a relative URL - foreign context**

```
<c:import url=”/logo.html” context=”/master”/>
```

URL Related Actions(2)

Exporting the content of the url

- Export the content of the URL resource as a String

```
<c:import var="customers"
 url=”http://acme.com/exec/customers?country=USA"/>
```

- Export the content of the URL resource as a Reader

```
<c:import varReader="customers"
 url=”http://acme.com/exec/customers?country=USA">
    Body content </c:import>
```

URL Related Actions(3)

- <import>
  - **Imports the content of a URL- based resource:**
  - **Resource content inlined or exported as a String object**

    ```
    <c:import url=″url″ [context=″context″]
     [var=″varName″] [scope=″{page|request|…}″]
     [charEncoding=″charEncoding″]>
    optional body content for <c:param> subtags
    </c:import>
    ```

  - **Resource content exported as a Reader object**

    ```
    <c:import url=″url″ [context=″context″]
    varReader=″varReaderName″
    [charEncoding=″charEncoding″]>
    body content where varReader is consumed by
    another action
    </c:import>
    ```

URL Related Actions(4)

- <c:url>
  - Builds a URL with the proper rewriting rules applied.
  - **Without body content**

    *<c:url value=”**value**” [context=”**context**”] [var=”**varName**”] [scope=”**{page|request|…}**”]/>*
  - **With body content to specify query string parameters**

    *<c:url value=”**value**” [context=”**context**”] [var=”**varName**”] [scope=”**{page|request|…}**”]>*

    *<c:param> **subtags***

    </c:url>

URL Related Actions(5)

- **<redirect>**
  - **Sends an HTTP redirect to the client.**
  - **Without body content**

```
<c:redirect url="value" [context="context"]/>
```

  - **With body content to specify query string parameters**

```
<c:redirect url="value" [context="context"]>
<c:param> subtags
</c:redirect>
```

URL Related Actions(6)

- <c:param>
  - Adds request parameters to a URL. Nested action of <c:import>, <c:url>,<c:redirect>.
  - Parameter value specified in attribute “value”

```
<c:param name=”name” value=”value”/>
```

  - Parameter value specified in the body content

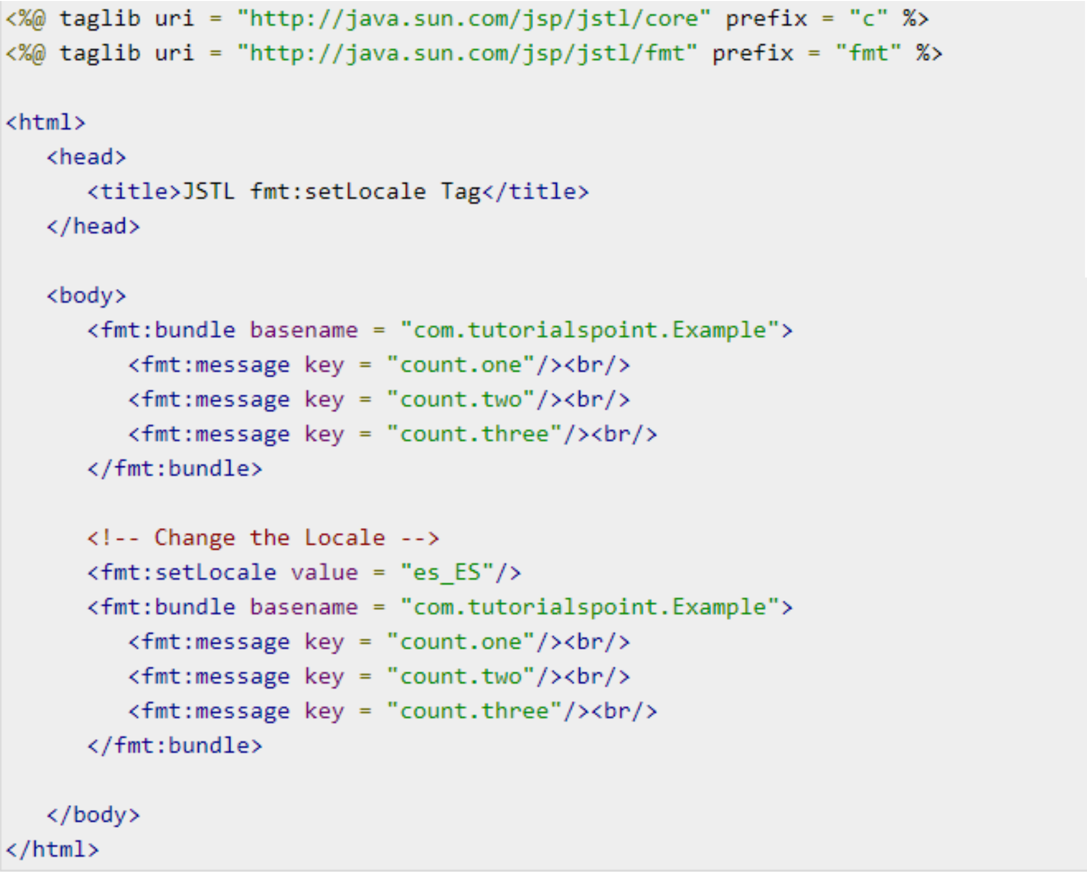
```
<c:param name=”name”>
parameter value
</c:param>
```

Internationalization(i18n) Actions

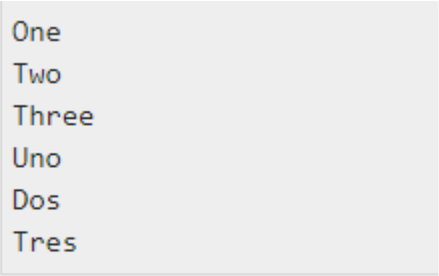
- URI: http://java.sun.com/jsp/jstl/fmt ,prefix="fmt"
  - **<fmt:setLocale>**: used to store the given locale in the locale configuration variable.
  - **<fmt:bundle>**: tag will make the specified bundle available to all **<fmt:message>** tags that occur between the bou**nding** **<fmt:bundle>** and **</fmt:bundle>** tags. With this, you need not specify the resource bundle for each of your **<fmt:message>** tags.
  - **<fmt:setBundle>**: used to load a resource bundle and stores it in the named scoped variable or the bundle configuration variable.
  - **<fmt:message>**: tag maps key to localized message and performs parametric replacement.
  - **<fmt:param>**: set parameter values for messages with parameters.
  - **<fmt:requestEncoding>**: used to specify the encoding type used by forms that post data back to the Web application.

Internationalization(i18n) Actions

- Example



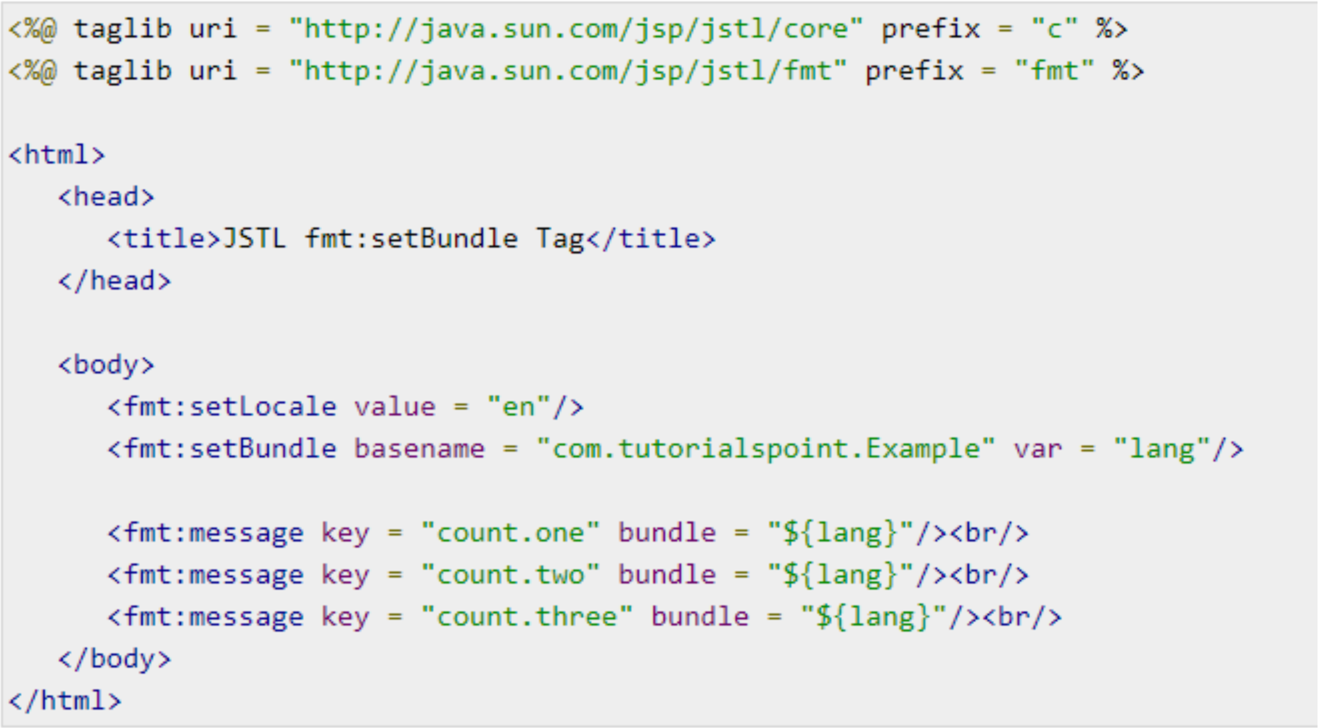
```
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<%@ taglib uri = "http://java.sun.com/jsp/jstl/fmt" prefix = "fmt" %>

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

   <body>
      <fmt:bundle basename = "com.tutorialspoint.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.tutorialspoint.Example">
         <fmt:message key = "count.one"/><br/>
         <fmt:message key = "count.two"/><br/>
         <fmt:message key = "count.three"/><br/>
      </fmt:bundle>

   </body>
</html>
```



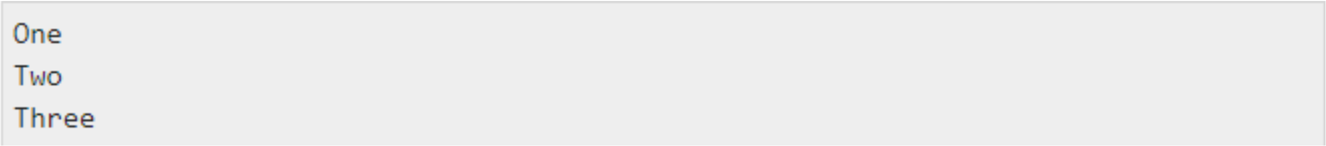
```
One
Two
Three
Uno
Dos
Tres
```

Internationalization(i18n) Actions

- Example



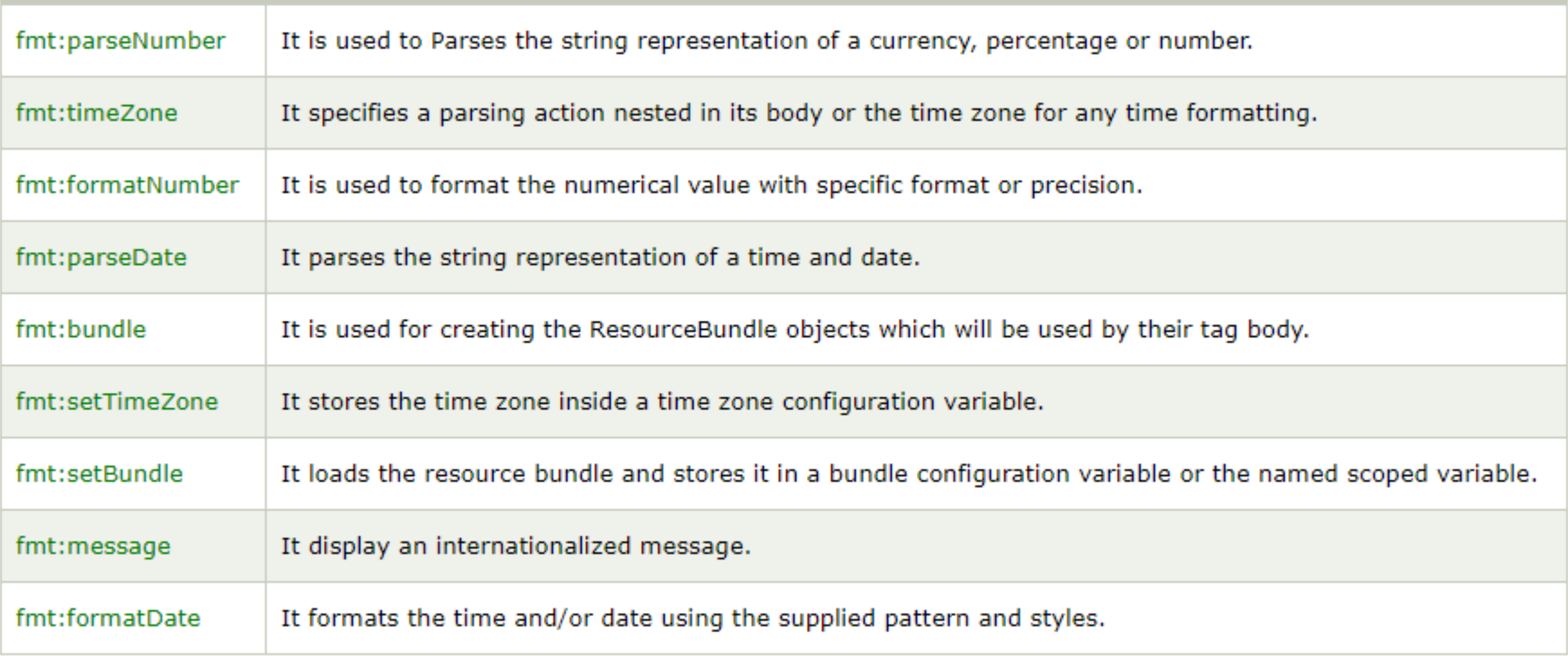
```
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<%@ taglib uri = "http://java.sun.com/jsp/jstl/fmt" prefix = "fmt" %>

<html>
   <head>
      <title>JSTL fmt:setBundle Tag</title>
   </head>

   <body>
      <fmt:setLocale value = "en"/>
      <fmt:setBundle basename = "com.tutorialspoint.Example" var = "lang"/>

      <fmt:message key = "count.one" bundle = "${lang}"/><br/>
      <fmt:message key = "count.two" bundle = "${lang}"/><br/>
      <fmt:message key = "count.three" bundle = "${lang}"/><br/>
   </body>
</html>
```



```
One
Two
Three
```

Formatting Actions

- URI: http://java.sun.com/jsp/jstl/fmt ,prefix=“fmt”



fmt:parseNumber	It is used to Parses the string representation of a currency, percentage or number.
fmt:timeZone	It specifies a parsing action nested in its body or the time zone for any time formatting.
fmt:formatNumber	It is used to format the numerical value with specific format or precision.
fmt:parseDate	It parses the string representation of a time and date.
fmt:bundle	It is used for creating the ResourceBundle objects which will be used by their tag body.
fmt:setTimeZone	It stores the time zone inside a time zone configuration variable.
fmt:setBundle	It loads the resource bundle and stores it in a bundle configuration variable or the named scoped variable.
fmt:message	It display an internationalized message.
fmt:formatDate	It formats the time and/or date using the supplied pattern and styles.

SQL Actions

- URI: http://java.sun.com/jsp/jstl/sql, prefix="sql"
    - <sql:query>
    - <sql:update>
    - <sql:transaction>
    - <sql:setDataSource>
    - <sql:param>
    - <sql:dateParam>

XML Core Actions

- URI: http://java.sun.com/jsp/jstl/xml, prefix="x"
  - <x:parse>
  - <x:out>
  - <x:set>

XML Flow Control Actions

- URI: http://java.sun.com/jsp/jstl/xml, prefix="x"
  - <x:if>
  - <x:choose>
  - <x:when>
  - <x:otherwise>
  - <x:forEach>

XML Transform Actions

- URI: http://java.sun.com/jsp/jstl/xml, prefix="x"
  - <x:transform>
  - <x:param>

Functions Tag Library

- URI: http://java.sun.com/jsp/jstl/functions, prefix="x"
  - <fn:contains>
  - <fn:containsIgnoreCase>
  - <fn:endsWith>
  - <fn:escapeXml>
  - <fn:indexOf>
  - <fn:join>
  - <fn:length>
  - <fn:replace>

Functions Tag Library(2)

- <fn:split>
- <fn:startsWith>
- <fn:substring>
- <fn:substringAfter>
- <fn:substringBefore>
- <fn:toLoweCase>
- <fn:toUpperCase>
- <fn:trim>

How to use JSTL in a project

- Download the lates version of jst from
- http://www.apache.org/dist/jakarta/taglibs/standard/binaries/
- Open the archive and, copy the jstl.jar and standart.jar files under the lib folder to the /WEB-INF/lib folder of your project
- Define the taglib in jsp as:
- <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
- If the code above doesn’t works, use below
- <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core_rt" %>

Summary

JSTL Introduction

- What is JSTL?
- What JSTL offer?
- JSTL Tag Libraries

Actions

- General purpose
- Conditional
- Iterator
- URL relate
- Internationalization (i18n)
- Formatting
- SQL
- XML core
- XML flow control
- XML transform

Function tag library

How to use JSTL in a project