

# The Standard Tag Library

---



**Kevin Jones**

@kevinrjones [www.rocksolidknowledge.com](http://www.rocksolidknowledge.com)



# Overview



Understand why there is a Java Standard Tag Library

Understand the basic concepts behind the JSTL

Understand how the core tags work



## Why Tags?

**Provide encapsulation of (UI) logic**

**Replace script on a page**

**Better for page designers than having to write Java code**



# What is a 'TagLibrary'?



# What is a 'TagLibrary'?

**Tag libraries are collections of tags**



# What is a 'TagLibrary'?

**Tag libraries are collections of tags**

**The Java Standard Tag Library (JSTL) is a set of standard tags used in JavaServer Pages**



# Using Tags

## **Tags are managed as libraries**

- Individual tags are Java classes
- Tag metadata held in Tag Library Descriptor file
- Collected together in a JAR file



# Using Tags

## Tags are managed as libraries

- Individual tags are Java classes
- Tag metadata held in Tag Library Descriptor file
- Collected together in a JAR file

```
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>3.0</jsp-version>
  <short-name>u</short-name>
  <uri>http://kevinj.mantiso.com/tags</uri>

  <tag>
    <name>user</name>
    <tag-class>ejsp.User</tag-class>
  </tag>
</taglib>
```





# Using Tags

## Tag libraries are introduced with a directive

- Prefix is used on the page as a shorthand name
- URI identifies the tag library

## Container has to associate URI with tag library

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
  
<c:out value="${param['name']}" />
```

'c' is the standard  
prefix for the core  
library

<http://java.sun.com/jsp/jstl/core>  
is the JSTL core URIs



# The Different Libraries

## **Core ('c')**

- Setting and displaying data
- Flow control
- Text/URL management

## **I18N ('fmt')**

- Locale and resource management

## **XML**

- Basic XML
- Flow control
- Transforms

## **SQL**

- Database management

**Core covered here, others are covered in the appropriate chapter**



# General

Many tag attribute accept expressions

Many tags can process their bodies

- typically body replaces a tag attribute

Many tags can save result into a variable

Can configure JSTL tags

- Global configuration
- use **<context-param>**
- Access and set through static methods of **Context** class

```
<context-param>  
  <param-name>javax.servlet.jsp.jstl.fmt.locale</param-name>  
  <param-value>en_GB</param-value>  
</context-param>
```



# Core Tags - Conditional

## Conditional

- **if**

```
<c:if test="${param['name']} == 'fred'">
    <b>Hi <c:out value="${param['name']}" /></b>
</c:if>

<!-- var and scope are optional -->
<c:if test="${param['name']} == 'fred'" var="isFred" scope='request'>
    <b>Hi <c:out value="${param['name']}" /></b>
</c:if>
<br/>

<c:if test="${isFred}">
    Hello Again
</c:if>
```



# Core Tags - Conditional

## Conditional

- **choose/when/otherwise**

```
<c:choose>  
  <c:when test="${param.name == 'fred'}" >Hello Fred</c:when>  
  <c:when test="${param.name == 'alice'}" >Hello Alice</c:when>  
  <c:otherwise>You're not Alice or Fred</c:otherwise>  
</c:choose>
```



# Core Tags - Iteration

**Two tags, for-each and for-tokens that support**

- exporting an object that holds the current value
- ranges to iterator
- exposure of a status variable



# Core Tags - Iteration

## forEach

- Iterate over a collection
  - java.util.Collection
  - java.util.Map
  - array
  - java.util.Iterator
  - java.util.Enumeration
- comma separated **java.lang.String**

Use Iterator and  
Enumeration with care  
Not resettable.  
Must only be used  
by one tag at a time

```
<c:forEach[var="varName" items="collection"  
[varStatus="varStatusName"  
[begin="begin" [end="end" [step="step"]>
```

body content

```
</c:forEach>
```



# Core Tags - Iteration Examples

```
<c:forEach items="${list}" begin="1" end="10" step="2"
    var="dte" varStatus="status">
  <c:if test="${status.first}" >
    Begin: ${status.begin}<br/>
    End: ${status.end}<br/>
    Step: ${status.step}<br/>
    <hr/>
  </c:if>
  Date: ${dte}<br/>
  Current: ${status.current}<br/>
  Last: ${status.last}<br/>
  First: ${status.first}<br/>
  Count: ${status.count}<br/>
  Index: ${status.index}<br/>
  <hr/>
</c:forEach>
```

**status variable holds  
information about  
the iteration**

Begin: 1  
End: 10  
Step: 2

Date: Thu Jan 01 06:33:20 GMT 1970  
Current: Thu Jan 01 06:33:20 GMT 1970  
Last: false  
First: true  
Count: 1  
Index: 1

Date: Thu Jan 01 12:06:40 GMT 1970  
Current: Thu Jan 01 12:06:40 GMT 1970  
Last: false  
First: false  
Count: 2  
Index: 3





# Core Tags - Iteration

## forTokens

- Iterate over a token 'delimited' list

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



# Core Tags - Iteration Examples

```
<c:forTokens items="This;is;a;test"
              delims=";"
              var="item">
    ${item}<br/>
</c:forTokens>
```

This  
is  
a  
test



# Core Tags - Import

## Managing text is an important part of JSP

- `<c:import> == <jsp:include>++`

## Can import from

- This context
- Another context on this server
- Another server

## Imported data can be

- written to page
- stored in a string
- 'stored' in a Reader

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



# Core Tags - Import

## Managing text is an important part of JSP

- `<c:import> == <jsp:include>++`

## Can import from

- This context
- Another context on this server
- Another server

## Imported data can be

- written to page
- stored in a string
- 'stored' in a Reader

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



# Core Tags - Import

## Examples

```
<c:import url="/header.jsp"/>
```

**This context**

```
<c:import url="footer.jsp" var='footer' />
```

**To a variable**

```
${footer}
```

```
<c:import context="/tomcat-docs"  
          url="/servletapi/help-doc.html" />
```

**Cross context**  
**This must be**  
**allowed by the**  
**container**

```
<c:import url="http://kevinj.mantiso.com/index.jsp"/>
```

**Another server**



# Core Tags - URL

**There are issues with relative URLs in a web application**



# Core Tags - URL

**There are issues with relative URLs in a web application**

**No concept of a 'web application'**



# Core Tags - URL

**There are issues with relative URLs in a web application**

**No concept of a 'web application'**

**Using '/[url]' give server relative urls**

- /somepage.jsp
- http://myserver.com/sompage.jsp





# Core Tags - URL

**There are issues with relative URLs in a web application**

**No concept of a 'web application'**

**Using '/[url]' give server relative urls**

- /somepage.jsp
- http://myserver.com/sompage.jsp

**Using '[url]' somepage.jsp**

- is relative to the page
- From foo/index.jsp would give
- http://myserver.com/myapp/foo/sompage.jsp



# Core Tags - URL

**<c:url ... />**

- Allows the creation of application relative URLs
- Also adds session encoding where necessary



# Core Tags - URL

## Can specify

- 'value' of URL
- 'params' to be added
- Can create **application** relative urls
- application relative URLs start with a '/'

## Set 'context' param if importing from a foreign context

```
<c:url value="value" [context="context"]  
[var="varName"] [scope="{page|request|session|application}"]/>  
  
<c:url value="value" [context="context"]  
[var="varName"] [scope="{page|request|session|application}"]>  
<c:param> subtags  
</c:url>
```



# URL - Examples

## Server relative

```
<a href='/ForEach.jsp'>ForEach</a><br/>
<!-- output as -->
<a href='/ForEach.jsp'>ForEach</a><br/>
```

## Application relative

```
<a href='<c:url value="/ForEach.jsp"/>'>ForEach</a>
<!-- output as -->
<a href='/TestJSTL/ForEach.jsp'>ForEach</a>
```

```
<a href='
  <c:url value="/RemoteImport.jsp">
    <c:param name="url" >http://localhost:8080/index.jsp</c:param>
  </c:url>
'>Remote Import</a>
<!-- output as -->
<a href='
  /TestJSTL/RemoteImport.jsp?url=http%3A%2F%2Flocalhost%3A8080%2Findex.jsp
'>Remote Import</a>
```



# Summary



**JSTL Provides a core set of tags**

**These allow page authors control over 'view' logic**

**Allow better control over URLs**

**Other functionality also provided**

- SQL
- XML



What's Next

