



**Tag Libraries  
e JSTL**

*Helder da Rocha (helder@acm.org)*  
*[www.argonavis.com.br](http://www.argonavis.com.br)*

# Custom tags

- *JSP com JavaBeans fornecem um meio de diminuir código Java da página, mas não totalmente*
  - *Designers de página ainda têm que usar elementos de script para loops e lógica condicional (getProperty e setProperty não bastam)*
  - *Nem sempre os JavaBeans são suficientes para encapsular toda a lógica da aplicação*
- *A especificação prevê a criação de elementos XML personalizados (custom tags) para resolver essas limitações*
  - *Organizados em bibliotecas (taglibs)*
  - *Cada biblioteca tem seu próprio namespace*
- *Taglibs são declaradas no início de cada página ...*
- *... e usadas em qualquer lugar*

`<%@taglib uri="http://abc.com/ex" prefix="exemplo"%>`

← namespace

`<exemplo:dataHoje />`

→ produz

**Tuesday, May 5, 2002 13:13:13 GMT-03**

# Como usar custom tags

- A URI usada para identificar o prefixo de um custom tag não precisa ser real (e apontar para um local)
  - Serve apenas como **identificador**
  - Ligação entre a especificação da biblioteca (arquivo TLD) e o identificador é feito no arquivo **web.xml**

```
<web-app>
  ...
  <taglib>
    <taglib-uri>http://abc.com/ex</taglib-uri>
    <taglib-location>
      /WEB-INF/mytaglib.tld
    </taglib-location>
  </taglib>
</web-app>
```

Este é o deployment descriptor do Taglib.

Localização real!

# Exemplo de arquivo TLD

```
<?xml version="1.0" ?>  
<!DOCTYPE taglib PUBLIC  
"-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"  
"http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">
```

```
<taglib>
```

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

```
  <jsp-version>1.2</jsp-version>
```

```
  <short-name>exemplo</short-name>
```

```
  <uri>http://abc.com/ex</uri>
```

Sugestão de prefixo  
(autor de página pode  
escolher outro na hora)

URI identifica o prefixo.  
(autor de página tem que  
usar exatamente esta URI)

```
  <tag>
```

```
    <name>dataHoje</name>
```

```
    <tag-class>exemplos.DateTag</tag-class>
```

```
    <description>Data de hoje</description>
```

```
  </tag>
```

```
</taglib>
```

# Exemplos de Custom Tags

- *Veja exemplos/cap11/taglibs/*
  - *Vários diferentes exemplos de custom tags (de [6])*
  - *Código fonte em taglib/src/taglibdemo/\*.java*
  - *Páginas exemplo em src/\*Test.jsp (6 exemplos)*
    1. *Configure build.properties, depois, monte o WAR com:*  
> **ant build**
    2. *Copie o WAR para o diretório webapps do Tomcat*  
> **ant deploy**
    3. *Execute os tags, acessando as páginas via browser:*  
**http://localhost:porta/mut/**
- *Veja também exemplos/cap12/mvc/hellojsp\_2/*
  - *Aplicação MVC que usa custom tags*

# JSP Standard Tag Library

- *Esforço de padronização do JCP: JSR-152*
  - *Baseado no Jakarta Taglibs (porém bem menor)*
- *Oferece dois recursos*
  - *Conjunto padrão de tags básicos (Core, XML, banco de dados e internacionalização)*
  - *Linguagem de expressões do JSP 1.3*
- *Oferece mais controle ao autor de páginas sem necessariamente aumentar a complexidade*
  - *Controle sobre dados sem precisar escrever scripts*
  - *Estimula a separação da apresentação e lógica*
  - *Estimula o investimento em soluções MVC*

# Como usar JSTL

- 1. Fazer o download da última versão do site da Sun
- 2. Copiar os JARs das bibliotecas desejadas para o diretório **WEB-INF/lib/** da sua aplicação Web e os arquivos TLD para o diretório **WEB-INF/**
- 3. Declarar cada taglib e associá-la com seu TLD no deployment descriptor `web.xml`.
- 4. Incluir em cada página que usa os tags:

```
<%@ taglib uri="uri_da_taglib"
        prefix="prefixo" %>
```
- 5. Usar os tags da biblioteca com o prefixo definido no passo anterior

```
<prefixo:nomeTag atributo="..."> ...
</prefixo:nomeTag>
```

# Quatro bibliotecas de tags

- *Core library: tags para condicionais, iterações, urls, ...*  
`<%@ taglib uri="http://java.sun.com/jstl/ea/core" prefix="c" />`
  - *Exemplo: <c:if test="..." ... >...</c:if>*
- *XML library: tags para processamento XML*  
`<%@ taglib uri="http://java.sun.com/jstl/ea/xml" prefix="x" />`
  - *Exemplo: <x:parse>...</x:parse>*
- *Internationalization library*  
`<%@ taglib uri="http://java.sun.com/jstl/ea/fmt" prefix="fmt" />`
  - *Exemplo: <fmt:message key="..." />*
- *SQL library*  
`<%@ taglib uri="http://java.sun.com/jstl/ea/sql" prefix="sql" />`
  - *Exemplo: <sql:update>...</sql:update>*



# Linguagem de expressões

- *Permite embutir em atributos expressões dentro de delimitadores `${...}`*
  - *Em vez de `request.getAttribute("nome")`  
`${nome}`*
  - *Em vez de `bean.getPessoa().getNome()`  
`${bean.pessoa.nome}`*
- *Suporta operadores aritméticos, relacionais e binários*
- *Converte tipos automaticamente*  
`<tag item="${request.valorNumerico}" />`
- *Valores default*  
`<tag value="${abc.def}" default="todos" />`

# Principais ações

## ■ Suporte à impressão da linguagem expressões

- `<c:out value="{pessoa.nome}" />`

## ■ Expressões condicionais

- `<c:if test="{pessoa.idade >= 18}">`  
    `<a href="adultos.html">Entrar</a>`  
    `</c:if>`

- `<c:choose>`

- `<c:when test="{dia.hora == 13}">`  
        `<c:out value="{mensagemEspecial}" />`

- `</c:when>`

- `<c:otherwise>`

- `<c:out value="{mensagemPadrao}" />`

- `</c:otherwise>`

- `</c:choose>`

## ■ Iteração

- `<c:forEach items="{pessoas}" var="p" varStatus="s">`  
    `<c:out value="{s.count}" />. <c:out value="{p}" />`  
    `</c:forEach>`

# Internacionalização, XML e SQL

- *Ler propriedade de ResourceBundle*
  - `<fmt:message key="chave.do.bundle" />`
- *Operações diretas em banco de dados*
  - `<sql:query dataSource="{dsn}">`  
`SELECT...</sql:query>`
  - `<sql:transaction>`, `<sql:update>`, etc.
- *Operações com XML*
  - *Uso de expressões XPath em tags JSTL para XML*
  - *Ações XML: <x:out>, <x:set>, <x:if>, <x:choose>, <x:forEach> (atributo select contém expr. XPath)*
  - `<x:parse>` *Processa XML usando DOM ou filtro SAX*
  - `<x:transform>` *Realiza transformação XSLT.*

- *1. Inclua a taglib exemplo.tld em uma de suas aplicações e use o tag dataHoje*
  - *O tag está dentro de exemplo.jar que deve ser copiado para o diretório lib de sua aplicação*
- *2. Instalar tags do JSTL*
  - *Instale a biblioteca JSTL na sua aplicação, copiando os JARs e TLDs para os locais exigidos*
  - *Veja a documentação e os tags disponíveis*
- *3. Use os tags de lógica <if> e <forEach> para remover as expressões condicionais e iterações das páginas da aplicação de mensagens*

*helder@acm.org*

***argonavis.com.br***