

# **J2EE: Custom Tags**

Antonio Espín Herranz

# Problemas con JSPs

- JavaBeans nos permiten separar la parte de presentación de una página JSP de la implementación de una regla de negocio:
  - Sin embargo, sólo 3 elementos acción en JSP se pueden usar para acceder a un bean:
    - `jsp:useBean`
    - `jsp:getProperty`
    - `jsp:setProperty`
  - Por tanto a menudo tenemos que incluir código en un JSP
  - JSP (1.1 en adelante) define custom tags que pueden ser usadas para definir acciones propietarias, de manera que en nuestro JSP únicamente tenemos código de marcado.

# Características de las etiquetas personalizadas

- Custom tags tienen acceso a todos los objetos disponibles a JSPs
  - Pueden ser personalizadas usando atributos.
- Como JavaBeans se centran en reusabilidad de código.
- **Custom tags son extensiones de JSP definidas por el usuario.**
- Las librerías de etiquetas de JSP son creadas por programadores, expertos en acceder a datos y servicios, y son usadas por diseñadores de aplicaciones web especialistas en presentación.

# Crear una etiqueta personalizada

- **Necesitamos:**

1. Una **clase java** que herede de TagSupport o que implemente la interfaz Tag.
2. La definición del **fichero tld**.
3. Modificar el **web.xml** haciendo referencia a la etiqueta.
4. En la páginas JSP hacer una referencia a la etiqueta y utilizarla → Directiva **taglib**.
5. ***Ejemplo: Etiqueta personalizada que imprime un mensaje.***

# La clase Java

```
package etiquetas;

import javax.servlet.jsp.JspException;
import javax.servlet.jsp.JspWriter;
import javax.servlet.jsp.tagext.TagSupport;

public class MyCustomTag extends TagSupport {

    private static final long serialVersionUID = 1L;

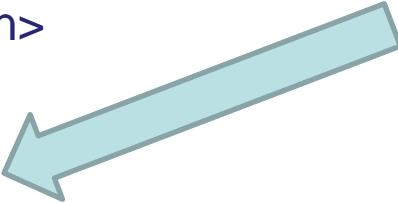
    public int doEndTag() throws JspException {
        JspWriter out = pageContext.getOut();

        try {
            out.println("Hello from the custom tag.");

        } catch (Exception e) {}
        return super.doEndTag();
    }
}
```

# Fichero tld

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!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/>
<tag>
  <name>myTag</name>
  <tag-class>etiquetas.MyCustomTag</tag-class>
  <body-content>empty</body-content>
</tag>
</taglib>
```



Un elemento **Tag** y una **clase Java** para cada etiqueta. Pero dentro del fichero TLD podemos tener declaradas varias etiquetas.

Este fichero se guarda en **WEB-INF/taglib.tld**

# Modificar el web.xml

**<jsp-config>**

**<taglib>**

**<taglib-uri>/myTLD</taglib-uri>**

**<taglib-location>/WEB-INF/taglib.tld</taglib-location>**

**</taglib>**

**</jsp-config>**

- Dentro del web.xml se indica la URI y la ubicación del fichero TLD.
- La URI es la que vamos a utilizar desde la página JSP para poder utilizar la etiqueta.

# Crear un JSP de prueba

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="/myTLD" prefix="easy"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Prueba de etiquetas personalizadas</title>
</head>
<body>
<p>llamamos a la etiqueta personalizada ...</p>
<easy:myTag/>
</body>
</html>
```



# Descriptor de la Librería de Etiquetas (Tag Library Descriptor)

- Fichero XML que define una librería de tags y sus etiquetas. Consta de:
  - Prólogo XML como todo documento XML
  - El elemento raíz `<taglib>`, que tiene como sub-elementos:
    - `tlib-version` → versión de la librería de etiquetas
    - `jsp-version` → la versión de JSP, actualmente 2.0
    - `shortname` → nombre corto para la librería
    - `description` → información para documentación
    - `uri` → enlace a más información sobre tag library
    - **tag** → elemento más importante, puede haber varios y tiene sub-elementos

# Elemento <tag> del descriptor de una librería de etiquetas

- Puede tener los siguientes sub-elementos (**name and tag-class son los únicos obligatorios**):
  - **name** → identificador de la etiqueta
  - **tag-class** → nombre de la clase completo que realiza el procesamiento de esta etiqueta
  - **tei-class** → clase de ayuda de esta etiqueta
  - **body-content** → el tipo de cuerpo de una etiqueta:
    - empty → no hay cuerpo.
    - JSP → cuerpo es un conjunto de elementos.
    - tagdependent → cuerpo debe ser interpretado por la etiqueta.
  - **description**
  - **attribute**: cero o más atributos que puede tener tres subelementos:
    - name (obligatorio)
    - required: true o false (valor por defecto false)
    - rtxprvalue: determina si el valor de este atributo se puede determinar en tiempo de ejecución
    - Los atributos se utilizan para declarar parámetros en las etiquetas.
  - ...

# Uso de etiqueta personalizada en JSP

- La directiva taglib en un JSP tiene el siguiente formato:
  - `<%@taglib uri="tagLibraryURI" prefix="tagPrefix" %>`
  - **uri** especifica una dirección absoluta o relativa que identifica el descriptor de la librería de etiquetas asociado con el prefijo
  - **prefix**: atributo que se precederá a las acciones personalizadas.
- Después de usar la directiva taglib es posible usar una etiqueta personalizada con los siguientes formatos:
  - `<prefix:tagName/>`
  - `<prefix:tagName>body</prefix:tagName>`
  - Para pasar atributos al manejador de la etiqueta hay que usar: `attributeName="attributeNameValue"`.
- Ejemplo:
  - `<math:power number="12" power="13"/>`

# Interface Tag

```
public interface Tag {  
    public final static int SKIP_BODY = 0;  
    public final static int EVAL_BODY_INCLUDE = 1;  
    public final static int SKIP_PAGE = 5;  
    public final static int EVAL_PAGE = 6;  
    public int doEndTag() throws JspException;  
    public int doStartTag() throws JspException;  
    public Tag getParent();  
    public void release();  
    public void setPageContext(PageContext pc);  
    public Tag setParent(Tag t);  
}
```

**setPageContext:** Este método lo invoca el motor de JSP para definir el contexto.

**setParent:** Lo invoca el motor de JSP para pasarle un manejador de etiqueta a la etiqueta padre.

**getParent:** Devuelve una instancia de la etiqueta padre.

**doStartTag:** Se procesa la etiqueta de apertura.

**doEndTag:** Se ejecuta después de doStartTag.

**release:** Lo invoca el motor de JSP para hacer limpieza

# Etiquetas con parámetros

- Ejemplo de otra etiqueta: // OJO un método set por cada atributo.

- `<ejemplos:suma num1=2 num2=3/>`

- ```
public class SumaTag extends TagSupport {  
    private int num1,num2;
```

```
    public void setNum1(int num1){ this.num1 = num1; }
```

```
    public void setNum2(int num2){ this.num2 = num2; }
```

```
    public int doStartTag() throws JspException {  
        try{  
            pageContext.getOut().print("Suma: " + (num1+num2));  
        } catch (IOException e) {  
            throw new JspException ("Error: IOException" + e.getMessage());  
        }  
        return SKIP_BODY;  
    }  
    public int doEndTag() throws JspException { return SKIP_PAGE; }  
}
```

# El fichero tld

```
<taglib>
  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name/>
  <tag>
    <name>sumaTag</name>
    <tag-class>etiquetas.SumaTag</tag-class>
    <body-content>empty</body-content>
    <attribute>
      <name>num1</name>
      <required>true</required>
      <type>int</type>
    </attribute>
    <attribute>
      <name>num2</name>
      <required>true</required>
      <type>int</type>
    </attribute>
  </tag>
</taglib>
```

## En el web.xml

```
<jsp-config>
  <taglib>
    <taglib-uri>/SumaTLD</taglib-uri>
    <taglib-location>/WEB-INF/taglib.tld</taglib-location>
  </taglib>
</jsp-config>
```

# En el JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="s" uri="/SumaTLD" %>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <h3>Prueba de etiquetas con Parámetros</h3>
    <s:sumaTag num1="4" num2="3" />
  </body>
</html>
```