

---

# **LENGUAJE DE EXPRESIONES JSTL**

**Eduard Lara**

# INDICE

---

1. Lenguaje de expresiones EL
2. Acceso a Beans mediante EL
3. Objetos implícitos disponibles con EL
4. Operadores EL
5. Etiquetas JSTL

# 1. LENGUAJE DE EXPRESIONES EL

---

- El lenguaje de expresiones se originó en JSTL 1.0 y está incorporado en la especificación JSP 2.0.
- Su finalidad es ayudar a crear páginas de JSP sin secuencias de comandos.
- La sintaxis del lenguaje de expresiones en una página de JSP es:

`$\{expr\}$`

- En esta sintaxis, `expr` indica una expresión válida del lenguaje de expresiones. Esta expresión se puede mezclar con texto estático y combinar con otras expresiones para formar expresiones más amplias.

# 1. LENGUAJE DE EXPRESIONES EL

---

- Para aplicar escape a los caracteres EL en una página de JSP, utilice la notación de barra invertida (\).
- Con escape no se evalúa la expresión EL.

`\${ expr }`

# 1. LENGUAJE DE EXPRESIONES EL

---

Las expresiones EL pueden utilizarse en las páginas de JSP de dos maneras:

- Como valores de atributo en acciones estándar y personalizadas. Cuando una expresión EL se utiliza como valor de un atributo, se evalúa y su valor sirve como valor del atributo. Por ejemplo:

`<jsp:include page="${location}">`

- Dentro de texto de plantilla, como HTML. Cuando una expresión EL se utiliza dentro de texto de plantilla, se evalúa y su valor se incorpora al flujo de salida. Por ejemplo:

`<h1>Welcome ${name}</h1>`

## 2. ACCESO A BEANS MEDIANTE EL

- Con el lenguaje de expresiones es fácil acceder a los beans dentro del espacio de nombres disponible para la página de JSP.
- Para acceder a un atributo del bean, basta con usar como sigue la notación de punto:

```
${bean.attribute}
```

- Cuando se especifica un nombre de bean sin ningún ámbito asociado, es preciso ubicar el bean para que el motor de JSP lo busque en los ámbitos.
- El orden de búsqueda en los ámbitos es: page, request, session y application (página, solicitud, sesión y aplicación).

## 2. ACCESO A BEANS MEDIANTE EL

---

- Las expresiones EL pueden manejar objetos nulos. En un scriptlet, es preciso comprobar un objeto antes de ejecutar un método para evitar una excepción `NullPointerException`.
- El mecanismo del lenguaje de expresiones maneja el valor nulo mostrando una cadena vacía.
- Para indicar el ámbito que contiene el bean, el nombre de atributo se precede con un objeto de ámbito implícito.
- En el ejemplo siguiente se recupera la propiedad `firstName` del bean `cust` ubicado en el ámbito de sesión:

```
${sessionScope.cust.firstName}
```

### 3. OBJETOS IMPLÍCITOS DISPONIBLES CON EL

---

Objeto implícito	Descripción
pageContext	El objeto PageContext
pageScope	Una asignación que contiene los atributos de ámbito de página y sus valores
requestScope	Una asignación que contiene los atributos de ámbito de solicitud y sus valores
sessionScope	Una asignación que contiene los atributos de ámbito de sesión y sus valores
applicationScope	Una asignación que contiene los atributos de ámbito de aplicación y sus valores
param	Una asignación que contiene los parámetros de solicitud y valores de cadenas individuales
paramValues	Una asignación que contiene los parámetros de solicitud y sus correspondientes matrices de cadenas



### 3. OBJETOS IMPLÍCITOS DISPONIBLES CON EL

Objeto implícito	Descripción
header	Una asignación que contiene los nombres de encabezado y valores de cadenas individuales
headerValues	Una asignación que contiene los nombres de encabezado y sus correspondientes matrices de cadenas
cookie	Una asignación que contiene los nombres de cookies y sus valores

- Por ejemplo, para acceder a un parámetro de solicitud denominado username puede emplearse esta expresión EL:
- Si un bean devuelve una matriz, se puede acceder a un elemento de la matriz mediante su índice:

``${param.username}``

``${paramValues.fruit[2]}``

## 4. OPERADORES EL

---

Operación aritmética	Operador
Suma	+
Resta	-
Multiplicación	*
División	/ y div
Resto	% y mod

Expresión EL	Resultado
$\{3 \text{ div } 4\}$	0.75
$\{1 + 2 * 4\}$	9
$\{(1 + 2) * 4\}$	12
$\{32 \text{ mod } 10\}$	2

## 4. OPERADORES EL

Comparación	Operador
Igual que	==y eq
Distinto de	!=y ne
Menor que	<y lt
Mayor que	>y gt
Menor o igual que	<=y le
Mayor o igual que	>=y ge

Operación lógica	Operador
and	&& y and
or	y or
not	!y not

## 5. ETIQUETAS JSTL

En esta sección se resumen las etiquetas disponibles en JSTL por categoría funcional. La siguiente tabla contiene las cinco categorías funcionales de etiquetas en JSTL, sus valores URI (que se usan en las directivas taglib de JSP) y el prefijo típico que se antepone en cada caso.

Área funcional	URI	Prefijo
Acciones básicas	<a href="http://java.sun.com/jsp/jstl/core">http://java.sun.com/jsp/jstl/core</a>	c
Acciones de procesamiento	<a href="http://java.sun.com/jsp/jstl/xml">http://java.sun.com/jsp/jstl/xml</a>	x
Acciones de formato	<a href="http://java.sun.com/jsp/jstl/fmt">http://java.sun.com/jsp/jstl/fmt</a>	fmt
Acciones de acceso a base de datos relacional	<a href="http://java.sun.com/jsp/jstl/sql">http://java.sun.com/jsp/jstl/sql</a>	sql
Acciones de función	<a href="http://java.sun.com/jsp/jstl/functions">http://java.sun.com/jsp/jstl/functions</a>	fn

## 5. ETIQUETAS JSTL

### Etiquetas de la biblioteca core

Las etiquetas del área funcional básica incluyen acciones para enviar al flujo de respuesta, realizar operaciones condicionales y efectuar iteraciones.

Tag	Finalidad
c:out	Evalúa una expresión y envía el resultado al objeto JspWriter actual
c:set	Define el valor de una variables o propiedad de ámbito
c:remove	Elimina una variable de ámbito
c:catch	Captura un objeto java.lang.Throwable que se produce en el cuerpo de la etiqueta
c:if	Evalúa una expresión y envía el resultado al objeto JspWriter actual
c:choose	Proporciona una condición autoexcluyente

# 11. ETIQUETAS JSTL

## Etiquetas de la biblioteca core

Tag	Finalidad
c:otherwise	Proporciona la última alternativa dentro de un elemento c:choose
c:forEach	Itera una colección de objetos o un número fijo de ciclos
c:forEachTokens	Divide una cadena en tokens y los itera
c:import	Importa el contenido de un recurso URL
c:url	Reescribe URL relativas
c:redirect	Envía un redireccionamiento HTTP al cliente
c:param	Agrega parámetros a la solicitud (se utiliza dentro de c:import, c:url y c:redirect)
c:when	Proporciona una alternativa dentro de un elemento c:choose

## 5. ETIQUETAS JSTL

---

### Etiquetas de la biblioteca xml

JSTL incluye etiquetas para procesar documentos XML. Las etiquetas disponibles en esta biblioteca son:

Tag	Finalidad
x:parse	Analiza un documento XML
x:out	Evalúa la expresión XPath y envía el resultado al objeto JspWriter actual
c:set	Evalúa la expresión XPath y almacena el resultado en una variable de ámbito
x:if	Evalúa el cuerpo de la etiqueta si la expresión XPath es true
x:choose	Proporciona una condición autoexcluyente
x:when	Proporciona una alternativa dentro de un elemento x:choose
x:otherwise	Proporciona la alternativa final dentro de un elemento x:choose

## 5. ETIQUETAS JSTL

---

### Etiquetas de la biblioteca xml

Tag	Finalidad
x:forEach	Analiza un documento XML
x:transform	Aplica una hoja de estilo XSLT a un documento XML
x:param	Proporciona parámetros de transformación (se utiliza dentro de un elemento x:transform)



# 11. ETIQUETAS JSTL

## Etiquetas de la biblioteca format

JSTL incluye etiquetas para internacionalización y asignación de formato.

Tag	Finalidad
fmt:setLocale	Almacene la configuración regional especificada en la variable de configuración regional
fmt:bundle	Crea un contexto de localización i18n que se utiliza en el cuerpo de la etiqueta
fmt:setBundle	Crea un contexto de localización i18n y lo almacena en la variable de ámbito o en la variable de configuración de contexto de localización
fmt:message	Busca el mensaje localizadoEvalúa el cuerpo de la etiqueta si la expresión XPath es true
fmt:param	Suministra un parámetro para sustitución dentro de un elemento fmt:message
Fmt:requestEncoding	Configura la codificación de caracteres de la solicitud

## 5. ETIQUETAS JSTL

---

### Etiquetas de la biblioteca format

Tag	Finalidad
<code>fmt:timeZone</code>	Especifica la zona horaria en la que se procesará la información horaria del cuerpo de la etiqueta
<code>fmt:setTimeZone</code>	Almacena la zona horaria especificada en una variable de ámbito o en la variable de configuración de la zona horaria
<code>fmt:formatNumber</code>	Asigna a un valor numérico un formato personalizado o de configuración regional, como una cifra, una divisa o un porcentaje
<code>fmt:parseNumber</code>	Analiza la representación en una cadena de cifras, divisas y porcentajes a los que se ha asignado un formato personalizado o de configuración regional
<code>fmt:parseDate</code>	Permite asignar formatos personalizados o de configuración regional de fecha y hora
<code>fmt:parseDate</code>	Analiza la representación en una cadena de fechas y horas a las que se ha asignado un formato personalizado o de configuración regional

## 5. ETIQUETAS JSTL

---

### Etiquetas de la biblioteca sql

JSTL incluye etiquetas para acceso a base de datos relacional.

Tag	Finalidad
fmt:query	Consulta la base de datos y almacena los resultados en una variable de ámbito
fmt:update	Ejecuta una sentencia INSERT, DELETE, UPDATE o SQL DDL y almacena en una variable de ámbito
fmt:transaction	Establece un contexto de transacción para los elementos sql:query y sql:update
fmt:setDataSource	Exporta una fuente de datos como una variable de ámbito o como variable de configuración de la fuente de datos
fmt:param	Configura los valores de los marcadores de parámetros (se utiliza con los elementos sql:query y sql:update)
fmt:dateParam	Configura los valores de los marcadores de parámetros de fecha (se utiliza con los elementos sql: query y sql:update)

# 11. ETIQUETAS JSTL

## Etiquetas de la biblioteca functions

JSTL incluye etiquetas para funciones, muchas de `java.lang.String`

Tag	Finalidad
<code>fn:contains</code>	Realiza una prueba de una subcadena especificada con distinción entre mayúsculas y minúsculas; devuelve true o false
<code>fn:containsIgnoreCase</code>	Realiza una prueba de una subcadena especificada sin distinción entre mayúsculas y minúsculas; devuelve true o false
<code>fn:endsWith</code>	Comprueba si una cadena termina en un sufijo especificado y devuelve true o false
<code>fn:escapeXml</code>	Aplica escape a caracteres que se interpretarían como marcado XML
<code>fn:indexOf</code>	Devuelve la posición de la primera vez que aparece una subcadena especificada
<code>fn:join</code>	Une los elementos de una matriz en una cadena

## 5. ETIQUETAS JSTL

---

### Etiquetas de la biblioteca functions

Tag	Finalidad
fn:length	Devuelve el número de elementos de una colección o el número de caracteres de una cadena
fn:replace	Devuelve una cadena después de substituir una subcadena por otra siempre que aparece
fn:split	Divide una cadena en una matriz de subcadenas basándose en un delimitador
Fn:startsWith	Comprueba si una cadena empieza con un prefijo especificado y devuelve true o false
fn:substring	Devuelve un subconjunto de una cadena, delimitado por puntos de inicio y fin
fn:substringAfter	Devuelve la subcadena que sigue a la subcadena especificada

## 5. ETIQUETAS JSTL

---

### Etiquetas de la biblioteca functions

Tag	Finalidad
fn:substringBefore	Devuelve la subcadena que procede a la subcadena especificada
fn:toLowerCase	Convierte los caracteres de una cadena a minúsculas
fn:toUpperCase	Convierte los caracteres de una cadena a mayúsculas
fn:trim	Suprime el espacio en blanco en ambos extremos de una cadena