Contenidos

Artículos

Referencias

Licencias de artículos

OpenCms

Introducción

OpenCms es un sistema de gestión de contenido de fuentes abiertas basado en Java y en tecnología XML. Es distribuido por la empresa Alkacon Software bajo licencia **GNU Lesser General Public License**.

Tags OpenCms

Cómo se usan

Para poder utilizar la *OpenCms JSP taglib*, debe colocarse la directiva **taglib** en la página JSP:

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/
cms" %>
```

Esta directiva debe colocarse antes de cualquier tag cms.

Los tags

El tag /cms:contentcheck/

: Condicional. Comprueba la disponibilidad de un item de contenido XML.

El tag /cms:conteninfo/

: Provee acceso al resultado ('result set') de un *contentload*.

El tag /cms:contentload/

: Provee acceso a la lectura de items de contenido XML.

El tag /cms:contentloop/

: Permite iterar sobre los valores de los elementos de un item de contenido XML.

El tag /cms:contentshow/

: Provee acceso y muestra la información del contenido XML.

El tag /cms:decorate/

: Provee decoración del contenido HTML.

El tag /cms:editable/

: Habilita el modo *direct edit* en una página.

El tag /cms:img/

: Soporta escalado usando los métodos nativos de OpenCms.

El tag /cms:include/

: Incluye dinámicamente otros elementos JSP en tiempo de ejecución. Estos sub-elementos pueden ser cacheados (ver la documentación de FlexCache).

El tag /cms:info/

: Imprime información del sistema en tiempo de ejecución.

El tag /cms:label/

: Provee acceso a la lectura de texto con idioma de los archivos de propiedades del *OpenCms workplace*.

El tag /cms:link/

Actúa como un *envoltorio* para armar una URL de OpenCms válida para determinado recurso.

El tag /cms:property/

Provee acceso de lectura de las propiedades del archivo actual.

El tag /cms:template/

Divide una página JSP en elementos a incluír via el uso del tag **/cms:include/**.

El tag /cms:user/

Provee acceso a la lectura de las propiedades del usuario actual.

OpenCms/cms:conteninfo

Se utiliza para obtener más información acerca del *result set* obtenido por un recolector de contenido.

Atributos

var (requerido)

Define el nombre para acceder el bean de contentinfo.

scope

Define el scope utilizado para almacenar el bean de contentinfo. El valor por omisión es *page*.

value

Es un valor macro para acceder a un valor simple del bean. Los valores provistos por dicho bean son:

```
* name.resultSize: la cantidad de contenidos XML obtenidos.
   * name.resultIndex: el índice al contenido actual, comenzando en 1.
```

Cuerpo

Este tag no tiene cuerpo.

Ejemplo de uso

```
...
<cms:contentinfo var="info" />

<c:if test="${info.resultSize > 0}">
```

...

OpenCms/cms:contentcheck

Se comprueba si el item existe, y se actúa en consecuencia.

Atributos

ifexists

Comprueba si el item existe.

ifexistsone

Comprueba si uno de los contenidos separados por comas existe.

ifexistsall

Comprueba si todos los contenidos separados por comas existen.

ifexistsnone

Comprueba si ninguno de los contenidos separados por comas existe.

Cuerpo

Cualquier código HTML, JSP or JSTL del template será procesado si la condición del tag es verdadera.

Ejemplos de uso

```
<cms:contentcheck ifexists="Title" >
 ...
</cms:contentcheck >
```

```
<cms:contentcheck ifexistsone="Teaser[0],Teaser[2]" >
   ...
```

</cms:contentcheck >

OpenCms/cms:contentload

Se utiliza para cargar una colección de recursos de contenido xml. Dependiendo del recolector pasado en el atributo, puede ser un recurso simple o una lista de recursos los que se carguen, y que serán iterados en este tag.

Atributos

collector (requerido)

El recolector que leerá el recurso de contenido xml. Deberá estar definido en una clase Java, configurada en *opencms-vfs.xml*. OpenCms contiene la clase CmsDefaultResourceCollector, que es una agregación de los siguientes recolectores:

- **singleFile**: obtiene un recurso simple, cuyo nombre se pasará por parámetro.
- **allInFolder**: obtiene todos los recursos de una carpeta, cuyo nombre se pasará por parámetro.
- **allInFolderDateReleasedDesc**: obtiene todos los recursos (ordenados por fecha de release) de una carpeta, cuyo nombre se pasará por parámetro.
- **allInFolderNavPos**: obtiene todos los recursos (ordenados por el valor de la propiedad *NavPos*) de una carpeta, cuyo nombre se pasará por parámetro.
- **allInSubTree**: obtiene todos los recursos de un sub-árbol, cuyo nombre de raíz se pasará por parámetro.
- **allInSubTreeDateReleasedDesc**: obtiene todos los recursos (ordenados por fecha de release) de un sub-árbol, cuyo nombre de raíz se pasará por parámetro.
- **allInSubTreeNavPos**: obtiene todos los recursos (ordenados por el valor de la propiedad *NavPos*) de un sub-árbol, cuyo nombre de raíz se pasará por parámetro.

Hay más recolectores disponibles en la clase CmsPriorityResourceCollector, que ordena los recursos por los valores de las propiedades *collector.priority* y *collector.date*:

- **allInFolderPriorityDateDesc**: obtiene todos los recursos (ordenados por el valor de la propiedad *collector.priority*, desempatando con la propiedad *collector.date*) de una carpeta, cuyo nombre se pasará por parámetro.
- **allInFolderPriorityTitleDesc**: obtiene todos los recursos (ordenados por el valor de la propiedad *collector.priority*, desempatando con la propiedad *Title*) de una carpeta, cuyo nombre se pasará por parámetro.
- **allInSubTreePriorityDateDesc** : obtiene todos los recursos (ordenados por el valor de la propiedad *collector.priority*, desempatando con la propiedad *collector.date*) de un sub-árbol, cuyo nombre de raíz se pasará por parámetro.
- **allInSubTreePriorityTitleDesc**: obtiene todos los recursos (ordenados por el valor de la propiedad *collector.priority*, desempatando con la propiedad *Title*) de un sub-árbol, cuyo nombre de raíz se pasará por parámetro.

En lugar de pasar el nombre del recolector en el atributo, se puede especificar en una expresión macro, indicando que el nombre deberá leerse, por ejemplo, de la propiedad *collector* de la JSP:

```
${property.collector}
```

param

le da información adicional al recolector.

Cuando se utiliza CmsDefaultResourceCollector, el formato es: "[filename]|[resource type]|[count]", donde:

- **[filename]** : el nombre del recurso a cargar; puede incluir una macro "${number}", que se reemplaza con el número del siguiente recurso de contenido xml cuando itera sobre los elementos simples.
- **[resource type]**: uno de los tipos de contenido xml definidos en *opencms-vfs.xml*.

- **[count]**: usa el uri actual en el OpenCms VFS como el nombre de recurso. Es típico utilizarlo con el singleFile collector.

De nuevo, se pueden utilizar expresiones como:

```
${property.xml-content}
```

En este caso, sería reemplazada por el valor de la propiedad *xml-content*.

```
${opencms.uri}
```

En este caso, sería reemplazada por el URI actual del OpenCms VFS.

```
${opencms.filename}
```

En este caso, sería reemplazada por el nombre del recurso de contenido xml actual, cuando itera sobre los elementos simples.

```
${param.resourceType}
```

En este caso, sería reemplazada por el parámetro de request Http *resourceType*.

Ejemplo:

```
param="/linklistdir/linklist_%(number).html|linklist|5"
```

En este caso:

- *linklistdir* es el directorio dentro del *sites/default*, que contendrá los items.
- *linklist_%(number).html* es el formato en el cual se crearán los archivos cuando, mediante el direct edit, se realice una adición de un elemento.
- *linklist* es el tipo de elementos.
- *5* es la cantidad de elementos a mostrar.

editable [truelfalse (default)]

define si el contenido xml soporta direct edit.

preload [true]

define si el recolector carga en forma previa su contenido. Esto hace falta para comprobar si el recolector ha retornado algún resultado.

Cuerpo

El código HTML, JSP or JSTL dentro del tag, será iterado con cada elemento.

Ejemplo de uso

Una vista detallada de un recurso de contenido xml utiliza un tag contentload de la siguiente manera:

```
<cms:contentload collector="singleFile" param="${opencms.uri}"
editable="true">
...
</cms:contentload>
```

Un ejemplo de cómo leer todo el contenido xml del tipo 11 en un sub-árbol que comienza en */xml-content/* en el sitio actual. Debe setearse la propiedad *collector* con el valor *allInSubTree* sobre la JSP, junto con una propiedad *xm-content* con el valor */xmlcontent/*:

```
<cms:contentload collector="${property.collector}"
   param="${property.xml-content}article_${number}.html|11"
editable="true" preload="true">
...
</cms:contentload>
```

Un ejemplo para hacer la pre-carga y comprobar si hay resultados:

```
<cms:contentload collector="${property.collector}"
   param="${property.xml-content}article_${number}.html|11"
editable="true">

 <cms:contentinfo var="info" />

 <c:if test="${info.resultSize > 0}">

 <cms:contentload>
 </cms:contentload>
```

</cms:contentload>

OpenCms/cms:contentloop

Un recurso de contenido xml consiste en varios items de contenido xml. Dependienda de la deficición de esquema, cada item puede tener múltiples valores.

Este tag itera sobre todos los valores de los elementos de un item de contenido xml.

Atributos

element (requerido)

El nombre del item de contenido sobre el cual iterar.

Cuerpo

El código HTML, JSP or JSTL dentro del tag, será iterado con cada valor.

Ejemplo de uso

Un recurso de contenido xml contiene múltiples valores en su item *Teaser*:

```
<cms:contentloop element="Teaser">
...
```

</cms:contentloop>

OpenCms/cms:contentshow

Se utiliza para mostrar el valor de los items de contenido xml.

Atributos

element

El nombre del item cuyo valor ha de mostrarse. Cuando se accede a un item con múltiples valores, cada uno puede accederse utilizando su índice:

```
elementname[index]
```

Si no hay elemento definido, el tag utilizará el nombre del elemento definido en un nodo padre.

index

Cuando se accede a un item con múltiples valores, cada uno puede accederse utilizando su índice.

Cuerpo

Este tag no tiene cuerpo.

Ejemplo de uso

Para mostrar el valor del item de contenido *Title*:

```
<cms:contentshow element="Title" />
```

Para mostrar el tercer valor del item de contenido *Teaser*:

```
<cms:contentshow element="Teaser[2]" />
```

ó

```
<cms:contentshow element="Teaser" index="2" />
```

Para mostrar todos los valores del item de contenido *Teaser*:

```
<cms:contentloop element="Teaser">
  <cms:contentshow />
```

</cms:contentloop>

OpenCms/cms:decorate

Se utiliza para decorar palabras o frases en las páginas HMTL ya renderizadas, con tags adicionales en un paso de procesamiento posterior.

Atributos

file (requerido)

El nombre del archivo de configuración utilizado.

locale

Parámetro de nacionalidad.

Cuerpo

El cuerpo de este tag debe contener el HTML a ser decorado.

Ejemplo de uso

En este ejemplo, la configuración de la decoración está definida en el archivo */alkacon-documentation/documentation_taglib/test_tag_decorate.xml*. Por cada utilización del tag <*cms:decorate*>, se puede utilizar un archivo de configuración diferente.

Si el tag <*cms:include*> se utiliza dentro del tag <cms:decorate>, el atributo *cachable=false* debe utilizarse en <*cms:include*>, de otra forma la decoración no funcionará.

```
<cms:decorate
file="/system/modules/mymodule/decoration/configuration.xml" >
 <cms:include element="body" editable="true" cacheable="false" />
</cms:decorate>
```

OpenCms/cms:include

Se utiliza para incluir archivos del OpenCms VFS dinámicamente en tiempo de ejecución. Este archivo es tratado como una request con parámetros opcionales.

Hay distintas opciones para determinar el nombre del archivo incluido utilizando los siguientes atributo:

- page
- property
- attribute

Si ninguno de estos atributos se han seteado, el cuerpo del tag <*cms:include*> es evaluado y el resultado se utiliza como el nombre del archivo. Si aun no se puede resolver el nombre, se utiliza el valor del método *getUri()* del CmsRequestContext actual.

Atributos

page|file

Nombre del archivo incluido.

property

Nombre de una propiedad del JSP, que a su vez especifica el nombre del archivo incluido.

attribute

Clave de la entrada en la hashtable de request, correspondiente al nombre del archivo incluido.

element

Si la JSP incluida está dividida en elementos mediante el tag *cms:template*, solo el elemento especificado de la JSP será incluido. El atributo *element* es tratado como un parámetro adicional de request.

suffix

Agrega un sufijo al nombre del archivo.

cachable

Si es *false*, el archivo incluido no será cacheado por Flexcache.

Cuerpo

Se puede utilizar código scriplet para determinar el nombre del archivo incluido. Se pueden agregar claves/valores adicionales a la hastable de parámetros de request para pasarlos como el archivo incluido, por ejemplo:

```
<cms:param name="myparam" value="myvalue" />
```

Ejemplo de uso

Incluir la JSP *some_page.html*:

```
<cms:include page="some_page.html" />
```

ó

```
<cms:include file="some_page.html" />
```

Evaluar el cuerpo para determinar el nombre del archivo:

```
<cms:include>
 <cms:property name="template" file="parent"/>
</cms:include>
```

Incluir un archivo seteando su nombre en la hashtable de atributos de request:

```
<%
...
request.setAttribute("body", "../elements/template-body.html" );
...
%>
```

```
<cms:include attribute="body">
   <cms:param name="__locale"><%= locale
%></cms:param>
</cms:include>
```

Incluir el archivo especificado en el atributo *file*. Agregar parámetros adicionales a la request de los archivos incluidos:

```
<cms:include file="../elements/template-nav-top.jsp">
  <cms:param name="__locale"><%= locale %></cms:param>
  <cms:param name="__navpart" value="toprow" />
</cms:include>
```

Incluir el elemento *head* de la JSP especificada por la propiedad *template*:

<cms:include property="template" element="head" />

OpenCms/cms:label

Provee acceso a la lectura de texto con idioma de los archivos de propiedades del *OpenCms workplace*.

Sólo debe utilizarse este tag si se quiere extender el OpenCms workplace.

Atributos

Este tag no tiene atributos.

Cuerpo

El cuerpo contiene el nombre de la propiedad clave del valor de string a ser leído.

Ejemplo de uso

Leer el valor de la clave *flex.cache.label.title*:

<cms:label>flex.cache.label.title</cms:label>

OpenCms/cms:link

Actúa como un envoltorio para armar una URL de OpenCms válida para un recurso interno. Esto permite colocar URLs sin necesidad de agregar el nombre de contexto manualmente.

Atributos

Este tag no tiene atributos.

Cuerpo

El cuerpo contiene el nombre del recurso.

Ejemplo de uso

Armar una URL para un recurso de la carpeta actual:

```
<cms:link>index.html</cms:link>
```

Armar una URL para un recurso de otra carpeta:

```
<cms:link>/some/other/folder/index.html</cms:link>
```

Armar una URL para un recurso de la carpeta raíz:

<cms:link>/index.html</cms:link>

OpenCms/cms:property

Provee acceso de lectura de las propiedades del archivo actual o sus carpetas contenedoras, tal como esté configurado en el diálogo contextual de *Properties* de la vista de explorador de OpenCms.

Atributos

name (requerido)

Nombre de la propiedad a ser leída.

file

Especifica donde buscar la propiedad. Soporta los siguientes valores:

- **uri (default)**: busca la propiedad en el archivo correspondiente a la uri.
- **search.uri** ó **search**: busca la propiedad en todas las carpetas contenedoras, comenzando con el archivo, y yendo hacia arriba del árbol si no la encuentra.
- **element.uri**: busca la propiedad en el sub-elemento procesado actualmente. Esto es útil en templates u otras páginas constituidas por varios elementos.
- **search.element.uri**: busca la propiedad en todas las carpetas contenedoras, comenzando con el archivo del sub-elemento procesado actualmente, y yendo hacia arriba del árbol si no la encuentra.
- **{some-file-uri}**: busca la propiedad en el archivo exacto del OpenCms VFS.

Cuerpo

Este tag no tiene cuerpo.

Ejemplo de uso

Leer la propiedad *Title* del archivo actual:

```
<cms:property name="Title" />
```

ó

```
<cms:property name="Title" file="uri" />
```

Tratar de leer la propiedad *locale* del archivo actual, y si no está ahí, buscar en las carpetas contenedoras:

```
<cms:property name="Title" file="search" />
```

ó

```
<cms:property name="Title" file="search.uri" />
```

Leer la propiedad *Title* del archivo */index.html*: <cms:property name="Title" file="/index.html" />

OpenCms/cms:template

Divide una página JSP en elementos a incluír via el uso del tag **cms:include**.

Atributos

element

Especifica el nombre mediante el cual el template es incluido.

Cuerpo

Código HTML, JSP or JSTL del template.

Ejemplo de uso

Un template JSP con elementos *head* y *foot*.

```
<cms:template element="head">
 <html>
  <body>

   Este es el template head.

    <hr>


</cms:template>

<cms:template element="foot">

    <hr>

   Este es el template foot.

  </body>
 </html>
```

</cms:template>

Fuentes y contribuyentes del artículo

OpenCms *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102219 *Contribuyentes*: 5 ediciones anónimas

OpenCms/cms:conteninfo *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102087 *Contribuyentes*: 2 ediciones anónimas

OpenCms/cms:contentcheck *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102085 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:contentload *Fuente*: http://es.wikibooks.org/w/index.php?oldid=104973 *Contribuyentes*: Rgfernan, 1 ediciones anónimas

OpenCms/cms:contentloop *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102190 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:contentshow *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102456 *Contribuyentes*: 2 ediciones anónimas

OpenCms/cms:decorate *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102203 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:include *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102209 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:label *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102211 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:link *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102214 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:property *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102217 *Contribuyentes*: 1 ediciones anónimas

OpenCms/cms:template *Fuente*: http://es.wikibooks.org/w/index.php?oldid=102220 *Contribuyentes*: 1 ediciones anónimas

Licencia