



3w Soluciones

Crear una nube de etiquetas en OpenCms



1. Objetivo

El objetivo del presente documento es explicar el uso de lucene para la creación de nube de etiquetas usando librerías externas para la visualización de las mismas. En este ejemplo en concreto usaremos OpenCloud y cumulus para la vista del frontend.

2. Requisitos previos

OpenCms V. 7.5.0 o superior

Cumulus: <http://www.roytanck.com/wp-content/uploads/wp-cumulus-example.zip>

OpenCloud: <http://sourceforge.net/projects/opencloud/files/>

3. Pasos previos

Crear un índice de lucene que se llame blog y que sólo indexe los contenidos estructurados del blog

4. Código

Realizar la nube de etiquetas con OpenCms apoyándonos en alguna librería externa que nos facilite la vista de esta funcionalidad es muy sencilla, lo único que necesitamos es usar la api de lucene, el motor de búsquedas de lucene que nos permita obtener los keywords o palabras clave de los contenidos estructurados y establecer un peso, que no servirá para establecer el tamaño de las palabras en el tag cloud.

Vamos a mostrar dos librerías para este módulo, uno que será para que las palabras claves sean simples enlaces usando el OpenCloud y otro que usará un flash mediante cumulus.

4.2 Código de lucene

```
<%@page buffer="none" session="false"  
import="org.opencms.jsp.*,org.opencms.main.*,org.opencms.file.*,  
java.util.*,org.mcavallo.opencloud.*,  
org.apache.lucene.index.*,org.opencms.search.CmsSearch"%>
```

Importamos las librerías que necesitamos

```
CmsJspActionElement cms = new  
CmsJspActionElement(pageContext,request,response);
```



Inicializamos los objetos CmsJspActionElement

```
TermEnum terms =  
OpenCms.getSearchManager().getIndex("blog").getSearcher().getIndex  
Reader().terms();
```

Obtenemos los terminus indexados en lucene

```
while(terms.next())  
{  
    if(terms.term().field().equals("keywords"))  
    {  
        terms.docFreq();  
        terms.term().text();  
    }  
}
```

Con este código obtenemos todos los keywords indexados y la frecuencia con la que aparecen.

4.3 Código para OpenCloud

```
Cloud cloud = new Cloud();
```

Inicializamos el objeto Cloud

```
cloud.setMaxTagsToDisplay(50);  
cloud.setMaxWeight(20.0);  
cloud.setMinWeight(5);  
cloud.setDefaultLink(cms.link("/nav/blog/tags/index.html"));
```

Indicamos el número máximo de tags a mostrar, queremos que haya 20 niveles de tamaño, así que inicializamos el peso de las palabras en 20 e inicializamos la url del enlace de los tags.



```
while(terms.next())
{
    if(terms.term().field().equals("keywords"))
    {
        cloud.addTag(new
        Tag(terms.term().text(),terms.docFreq()));
    }
}

%>
```

Iteramos sobre los términos de lucene y vamos insertando los tags en el objeto cloud

```
<div class="tagcloud" style="margin: auto; width: 100%;">
<%
for (Tag tag : cloud.tags()) {
%>
    <a href="<%= tag.getLink()+ "?query="+tag.getName() %>"
style="font-size:<%= tag.getWeightInt()%>px"><%= tag.getName()
%></a>
<%
}
%>
</div>
```

Iteramos sobre los elementos de cloud y mostramos las etiquetas

4.4 Código para cumulus

```
String tags="";
while(terms.next())
{
    if(terms.term().field().equals("keywords"))
    {
        tags+="<a
href="+cms.link("/nav/blog/tags/index.html")+ "?query="+terms.term().text
```



```
()+" style='"+(10+terms.docFreq())+"  
hicolor='0x000000'>"+terms.term().text()+"</a>";  
    }  
}  
tags+="  
%>
```

Iteramos sobre los terminus de lucen y creamos una cadena con los tags con el siguiente formato:

```
<tags><a href='http://www.alejandroalves.com' style='22' color='0xff0000'  
hicolor='0x00cc00'>3w Soluciones</a>
```

```
<script type="text/javascript"  
src="<cms:link>../resources/js/swfobject.js</cms:link>"></script>
```

Importamos el swfobject

```
<div id="flashcontent">
```

```
<a href="http://www.adobe.com/go/getflashplayer">
```

```
</a>
```

```
</div>
```

```
<script type="text/javascript">
```

```
var flashvars = {  
    tagcloud: "<%=tags%>",  
    tcolor: "0x8d8a8a",  
    tcolor1:"0xF29800",  
    hicolor:"0x000000",  
    tspeed: "100",  
    distr: "true"  
};  
var params = {  
    wmode: "transparent"  
};  
var attributes = false;
```

```
swfobject.embedSWF("http://www.alejandroalves.com/system/mo
```



```
dules/com.tresw.module.frontend/resources/swf/tagcloud.swf?r="+new  
Date().getTime()+"" , "flashcontent", "300", "120", "9.0.0",false, flashvars,  
params, attributes);
```

```
</script>
```

Inicializamos las variables y cargamos el flash.
Por problemas con la cache de Internet Explorer, para que se visualice el flash, hay que poner la ruta absoluta al swf y añadir la fecha al final, para forzar a que haga la llamada. Por lo tanto, en la inicialización del swfobject hay que cambiar la siguiente línea, poniendo nuestra ruta absoluta:
"http://www.alejandroalves.com/system/modules/com.tresw.module.frontend/resources/swf/tagcloud.swf?r="+new Date().getTime()+""

5. Usando el módulo

Para usar el módulo sólo tenemos que crear un índice de lucene que indexe sólo los recursos del blog que se llame blog o con el nombre que queramos pero entonces habría que cambiar la siguiente línea:

```
OpenCms.getSearchManager().getIndex("blog").getSearcher().getIndex  
Reader().terms();
```

Cambiando "blog" por el nombre de nuestro índice.

Después dónde queramos que aparezca la nube, tendremos que incluir la siguiente línea:

Para incluir OpenCloud:

```
<cms:include  
file="%{(link.weak:/system/modules/com.tresw.opencms.tagcloud/element  
s/e_tagcloud.jsp)" >
```

Para incluir cumulus:

```
<cms:include  
file="%{(link.weak:/system/modules/com.tresw.opencms.tagcloud/element  
s/e_tagcloud_flash.jsp)" >
```